ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανίχνευση ηλικίας των χρηστών του Twitter μέσω υβριδικών αλγορίθμων μηχανικής μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ευάγγελος Μ. Αγορογιάννης

Επιβλέπουσα: Ιωάννα Ρουσσάκη
Επίκουρη Καθηγήτρια

Αθήνα, Σεπτέμβριος 2020

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Ανίχνευση ηλικίας των χρηστών του Twitter μέσω υβριδικών αλγορίθμων μηχανικής μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ευάγγελος Μ. Αγορογιάννης

Επιβλέπουσα: Ιωάννα Ρουσσάκη
Επίκουρη Καθηγήτρια

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 20η Νοεμβρίου 2020.

.....	.....	.....
Ιωάννα Ρουσσάκη	Μιλτιάδης Αναγνώστου	Συμεών Παπαβασιλείου
Επίκουρη Καθηγήτρια Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβιος 2020

.....
Ευάγγελος Μ. Αγορογιάννης
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε. Μ. Π.

Περίληψη

Τα τελευταία χρόνια η χρήση των μέσων κοινωνικής δικτύωσης είναι ευρεία και διαρκώς αυξανόμενη. Το Twitter, ακολουθώντας αυτή τη γενικότερη τάση, αποτελεί ένα από τα κοινωνικά δίκτυα που σημειώνει σημαντική άνοδο και γίνεται ιδιαίτερα δημοφιλές. Η πλατφόρμα αυτή παρέχει μεγάλο όγκο διαθέσιμων, ελεύθερων και εύκολα προσβάσιμων δεδομένων, η ανάλυση των οποίων στις μέρες μας αποτελεί κίνητρο για πολλούς ερευνητές ανά τον κόσμο και αποτελεί χρήσιμη πηγή πληροφοριών για πολλούς τομείς όπως οι επιχειρήσεις, η διαφήμιση και η υγεία. Επίσης, κρίνεται σημαντική η μελέτη τους με σκοπό να εξαχθούν στοιχεία για τις προτιμήσεις και τα ενδιαφέροντα των χρηστών. Ωστόσο, το Twitter παρά την πληθώρα δεδομένων που διαθέτει, δεν περιλαμβάνει δεδομένα σχετικά με τα δημογραφικά στοιχεία των χρηστών του, γεγονός που έχει προσελκύσει το ενδιαφέρον πολλών μελετητών για την εξαγωγή τέτοιου είδους πληροφορίας. Ειδικότερα για την ανίχνευση της ηλικίας, έχουν πραγματοποιηθεί πολλές έρευνες αξιοποιώντας τα διαθέσιμα δεδομένα και εφαρμόζοντας τεχνικές μηχανικής μάθησης για την επίλυση του προβλήματος.

Στην παρούσα διπλωματική εργασία προτείνονται δύο προσεγγίσεις για την ανίχνευση της ηλικίας των χρηστών του Twitter. Η πρώτη υλοποιείται με την δοκιμή αλγορίθμων παλινδρόμησης ώστε να πραγματοποιηθεί πρόβλεψη για την ακριβή τιμή της ηλικίας, ενώ η δεύτερη επιδιώκει μέσω εφαρμογής μοντέλων ταξινόμησης να πραγματοποιήσει προβλέψεις ώστε να τους κατηγοριοποιήσει σε 8 ηλικιακές ομάδες. Για την διεξαγωγή των πειραμάτων λαμβάνονται και αξιοποιούνται δεδομένα που παρέχει το Twitter. Συγκεκριμένα, έπειτα από την επεξεργασία αυτών των πληροφοριών και την παραγωγή νέων μεταδεδομένων, δημιουργείται ένα μεγάλο σύνολο χαρακτηριστικών. Αυτά περιλαμβάνουν στατιστικά στοιχεία σχετικά με το προφίλ του χρήστη στην ιστοσελίδα, καθώς και λεξικογραφικά δεδομένα που εξήχθησαν από τα tweets τους μέσω εφαρμογής τεχνικών Επεξεργασίας Φυσικής Γλώσσας κειμένου. Ορισμένα από αυτά είναι το πλήθος των followers, των followings, των likes, των δημοσιεύσεων, των αναδημοσιεύσεων, το θέμα που αναφέρονται τα tweets αλλά και ο αριθμός των hashtags (#) ή των tags (@) που περιέχουν. Το σύνολο των χαρακτηριστικών αποτελεί την είσοδο για τους διάφορους αλγορίθμους παλινδρόμησης και ταξινόμησης που δοκιμάστηκαν. Για την ανάδειξη του βέλτιστου μοντέλου προβλέψεων για κάθε περίπτωση χρησιμοποιήθηκε η μέθοδος βελτίωσης υπερπαραμέτρων και cross-validation μέσω του RandomizedSearchCV αλγορίθμου. Αυτή η μελέτη και για τις δύο προσεγγίσεις οδήγησε στην επιλογή του XGBoost μοντέλου ως καταλληλότερου για την ανίχνευση της ηλικίας, το οποίο παρουσίασε μέσο απόλυτο σφάλμα MAE ίσο με 4,09 έτη στην παλινδρόμηση και ακρίβεια (accuracy) 70% στην ταξινόμηση.

Για τις ανάγκες της παρούσας διπλωματικής εργασίας, συλλέγονται δεδομένα για το προφίλ και τα tweets ενός συνόλου χρηστών του Twitter ταυτοποιημένων ηλικιακά. Τα δεδομένα αυτά υπόκεινται σε μία σειρά από τεχνικές επεξεργασίας και οδηγούν στην πραγματοποίηση προβλέψεων για την ανίχνευση της ακριβούς ηλικίας και της ηλικιακής ομάδας που ανήκουν οι χρήστες, καταδεικνύοντας με αυτόν τον τρόπο την σπουδαιότητα των πληροφοριών που παρέχει το Twitter για την επίλυση του προβλήματος.

Λέξεις κλειδιά

Μέσα κοινωνικής Δικτύωσης, Twitter, Μηχανική Μάθηση, Ανίχνευση ηλικίας, Επεξεργασία Φυσικής Γλώσσας, Μοντελοποίηση Θέματος Κειμένου, Εξόρυξη Δεδομένων, Ανάλυση δεδομένων, Παλινδρόμηση, Ταξινόμηση, Επιστήμη Δεδομένων

Abstract

The spread and the daily use of social media have been significantly increased in recent years. Twitter consists one of them that follows the current trend and records impressive progress and expediently popularity. It provides a large amount of data that are public and easily accessed. Many researchers worldwide are motivated and try to exploit and analyze those data in order to acquire enhanced knowledge useful for marketing purposes or companies. Additionally, work on Twitter data may produce important indications related to the interests of the users. However, despite the multiple data that Twitter includes, it doesn't contain information about user demographics. This fact impelled the scientific community to perform related research to extract that kind of data. The age detection of users is one of the domains that the researchers are the most interested in and tried to solve this problem by applying machine learning techniques on the available data.

This diploma thesis suggests two different approaches in order to detect the age of Twitter users. The first solution predicts the exact age by using regression algorithms. The second aims to separate the users in 8 groups and predict the class they belong by examining classification models. Data coming from Twitter are leveraged to fulfill the purposes of this study. Those data are processed and produce new metadata, where all together construct a set of features. Some of these refer to the user profile and other are related to sociolinguistics, that are extracted after a Natural Language Processing operation on users' tweets. The features contain fields such as the number of followers, followings, likes, tweets, retweets, the topic of the tweets and the included hashtags (#) or tags (@). Those information create the dataset that is used for the training phase of the examined regression and classification algorithms. Hyperparameter tuning and cross-validation, via the RandomizedSearchCV algorithm, are used in both approaches to find the best model. The research qualified the XGBoost model as the optimal solution both for regression and classification, resulting respectively a mean absolute error (MAE) of 4,09 years and accuracy of 70%.

For the purposes of this diploma thesis, data related to user profile and tweets are collected from Twitter platform regarding a set of age identified users. The gathered data pass through many steps of preprocessing and arrive to the final phase of the study where the predictions about the exact age and the age group of users are performed. This fact demonstrates and highlights the importance and the contribution of Twitter data on resolving the problem of age detection.

Keywords

Social Media, Twitter, Machine Learning, Age detection, Natural Language Processing, NLP, Data Mining, Data analysis, Topic Modelling, Regression, Classification, Data Science

Ευχαριστίες

Για την εκπόνηση της παρούσας διπλωματικής εργασίας θα ήθελα ιδιαιτέρως να ευχαριστήσω την επιβλέπουσα καθηγήτρια Ιωάννα Ρουσσάκη για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα, καθώς και για την πολύτιμη βοήθεια και καθοδήγηση που μου προσέφερε καθόλη την διάρκεια της διαδικασίας.

Ακόμα, θα ήθελα να ευχαριστήσω θερμά την οικογένεια μου για την αμέριστη στήριξη και υπομονή όλα αυτά τα χρόνια.

Τέλος, ευχαριστώ τους φίλους μου εκτός και εντός σχολής για τα όμορφα αυτά φοιτητικά χρόνια.

Πίνακας Περιεχομένων

Λίστα Εικόνων

Λίστα Πινάκων

Κεφάλαιο 1

1 Εισαγωγή

1.1 Κοινωνικά δίκτυα και μέσα κοινωνικής δικτύωσης

Τα μέσα κοινωνικής δικτύωσης (online social networks – OSNs) αποτελούν ένα σύνολο συνεχών διαπροσωπικών αλληλεπιδράσεων και σχέσεων. Κάνοντας αναφορά σε ένα κοινωνικό δίκτυο πρόκειται για μια κοινωνική δομή που αποτελείται από πολλούς διαφορετικούς και ξεχωριστούς παράγοντες, όπως άτομα ή οργανισμούς. Ο όρος «κοινωνικά δίκτυα» αναφέρθηκε για πρώτη φορά από τους Émile Durkheim και Ferdinand Tönnies, η έρευνα των οποίων αποσκοπούσε στην μελέτη των κοινωνικών ομάδων και της ιδέας των κοινωνικών δικτύων, όπως παρατηρείται στις θεωρίες τους κατά τη δεκαετία του 1890. Ο τομέας των κοινωνικών δικτύων αναπτύχθηκε τις επόμενες δεκαετίες και κίνησε το ενδιαφέρον πολλών ερευνητών.

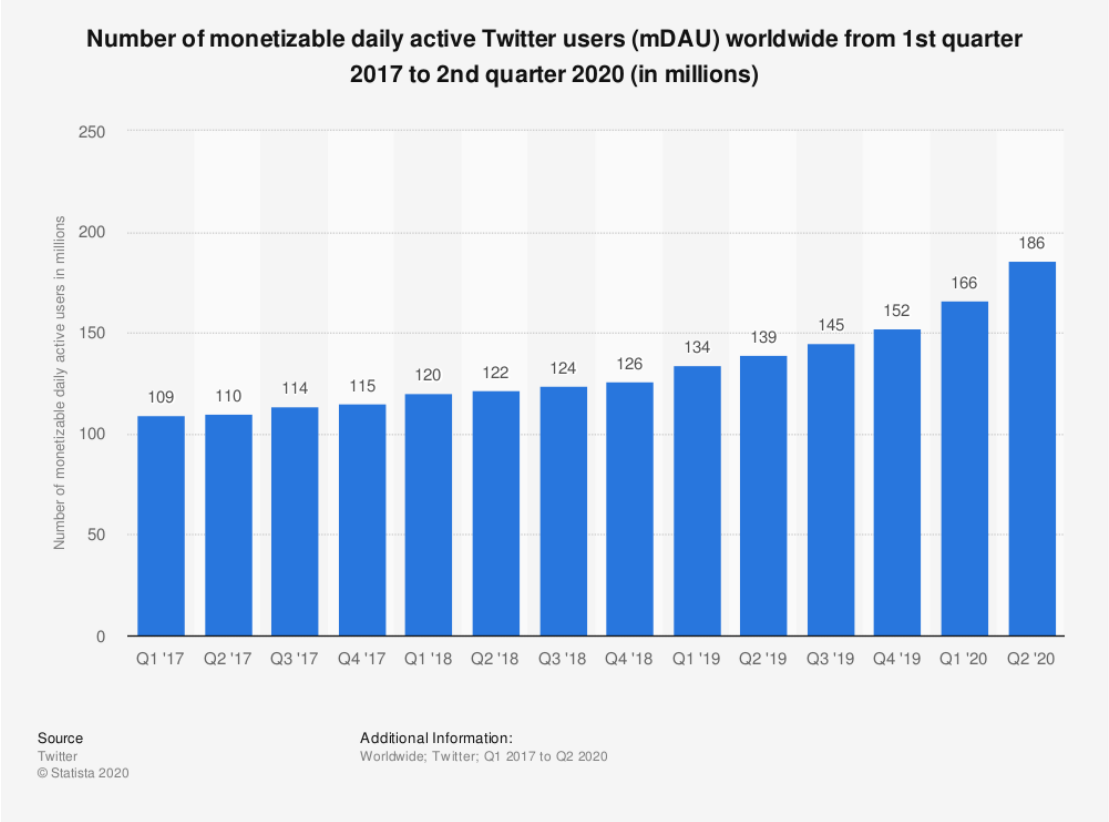
Τα μέσα κοινωνικής δικτύωσης μέσω του διαδικτύου δίνουν τη δυνατότητα δημιουργίας και ανάπτυξης κοινωνικών σχέσεων μεταξύ ανθρώπων με κοινές προτιμήσεις και δραστηριότητες, παρόμοιες κλίσεις και ενδιαφέροντα, που είναι ενεργά μέλη του κοινωνικού δικτύου. Ο όρος αυτός χρησιμοποιείται έντονα στις μέρες μας για να περιγράψει πλατφόρμες και ιστοσελίδες οι οποίες προσφέρουν την ανάπτυξη διεπαφών ανάμεσα στους χρήστες τους. Τα πιο διάσημα διαδικτυακά κοινωνικά δίκτυα σήμερα είναι το Facebook, το Instagram, το Twitter, το LinkedIn, το Pinterest, το Snapchat και το TikTok. Οι χρήστες των προαναφερθέντων ιστοσελίδων μπορούν να επικοινωνούν και να αναπτύσσουν επαφές μεταξύ τους μέσω αυτών των εικονικών κοινοτήτων.

Στην εποχή μας, η διάδοση των κοινωνικών δικτύων είναι ραγδαία και συνεχώς αυξανόμενη, με τα μέσα κοινωνικής δικτύωσης να αποτελούν πλέον, αναπόσπαστο κομμάτι της καθημερινότητας εκατομμυρίων ανθρώπων παγκοσμίως, καθώς και του σύγχρονου πολιτισμού. Η ευρεία χρήση των διάφορων OSNs τα τελευταία χρόνια έχουν οδηγήσει σε σημαντικές αλλαγές στις κοινωνικές αλληλεπιδράσεις των ανθρώπων και απασχολούν αρκετό από τον ελεύθερο χρόνο τους. Οι χρήστες των μέσων κοινωνικής δικτύωσης έχουν τη δυνατότητα, όχι μόνο να επικοινωνήσουν μεταξύ τους αλλά και να πληροφορηθούν για τις εξελίξεις και τα νέα της επικαιρότητας, καθώς και να μοιραστούν τις απόψεις, τις ιδέες και τα συναισθήματα τους για οποιοδήποτε θέμα επιθυμούν μέσω αναρτήσεων στα προφίλ τους.

1.2 Twitter

Το Twitter είναι ένα online μέσο κοινωνικής δικτύωσης που ιδρύθηκε το Μάρτιο του 2006 από τους Jack Dorsey, Evan Williams, Noah Glass και Biz Stone. Πρόκειται για ένα ιδιαίτερα διαδεδομένο μέσο κοινωνικής δικτύωσης που από την πρώτη του εμφάνιση κέντρισε το ενδιαφέρον πολλών ανθρώπων ώστε να το χρησιμοποιήσουν, κερδίζοντας έτσι δημοτικότητα φτάνοντας να αριθμεί πλέον 330 εκατομμύρια ενεργούς χρήστες παγκοσμίως σε μηνιαία βάση. Μάλιστα, από αυτούς περίπου 186 εκατομμύρια είναι ενεργοί καθημερινά και καταναλώνουν περίπου 3.53 λεπτά κάθε φορά που εισέρχονται στο Twitter. Τα συγκεκριμένα στοιχεία

αφορούν το δεύτερο τρίμηνο του 2020 και σύμφωνα με στοιχεία του Statista παρατηρείται συνεχής ανοδική τάση από το 2017 μέχρι σήμερα όπως φαίνεται στην Εικόνα 1.1[1]. Το Twitter είναι ένας ιστοχώρος, όπου οι χρήστες του ως επί το πλείστον αναρτούν και παρακολουθούν σύντομα μηνύματα, σε αντίθεση με άλλα εξίσου δημοφιλή κοινωνικά δίκτυα, όπως το Facebook και το Instagram, στα οποία οι χρήστες συνηθίζουν να δημοσιεύουν φωτογραφίες, τραγούδια καθώς και να επικοινωνούν με ιδιωτικές συνομιλίες. Τα σύντομα μηνύματα που δημοσιεύουν και διαβάζουν οι χρήστες του Twitter ονομάζονται tweets και έχουν μήκος έως 280 χαρακτήρες, γεγονός που κατατάσσει το Twitter στην κατηγορία του microblogging. Σύμφωνα με στατιστικές μελέτες του 2018 περίπου 500 εκατομμύρια tweets αναρτώνται ημερησίως το οποίο αντιστοιχεί με περίπου 5.700 το δευτερόλεπτο. Οι χρήστες μέσω των tweets μπορούν να εκφράσουν την άποψη τους για γεγονότα της επικαιρότητας σε ψυχαγωγικό, κοινωνικό ή πολιτικό επίπεδο καθώς και να αναπαράγουν κάποια είδηση.



Εικόνα 1.1: Αριθμός των ενεργών χρηστών του Twitter καθημερινά από το 2017 μέχρι σήμερα

Το Twitter διαθέτει ορισμένα στοιχεία και χαρακτηριστικά, τα οποία το διαφοροποιούν από άλλα OSNs και το διαμορφώνουν ως ιστότοπο. Τα hashtags χρησιμοποιούνται ευρέως από τους χρήστες στα tweets τους και πρόκειται για λέξεις ή φράσεις που ξεκινούν με το σύμβολο #, και είναι πολύ διαδεδομένα γιατί χρησιμοποιούνται για την ομαδοποίηση των tweets με βάση το θέμα τους. Επιπρόσθετα αρκετές φορές παρατηρείται και η χρήση των mentions στα tweets, δηλαδή λέξεων που ξεκινούν με το σύμβολο @ ακολουθούμενο από το όνομα ενός χρήστη, ώστε να στέλνεται άμεση ειδοποίηση για το συγκεκριμένο tweet στον αναφερόμενο χρήστη. Επίσης, ένα tweet μπορεί να περιλαμβάνει φωτογραφίες, βίντεο, τραγούδια και συνδέσμους. Ο κάθε χρήστης ακολουθεί (Following) τα άτομα για τα οποία θέλει να ενημερώνεται όταν

[1] Πηγή: https://www.statista.com/statistics/970920/monetizable-daily-active-twitter-users-worldwide/

δημοσιεύουν ένα tweet και αντιστοίχως, ακολουθείται και αυτός από άλλους χρήστες (Followers). Το Twitter δίνει τη δυνατότητα στους χρήστες να απαντήσουν με σύντομο μήνυμα στα tweets άλλων χρηστών (reply) καθώς και να τα αναδημοσιεύσουν (retweet) στο προσωπικό τους προφίλ, ενώ ταυτόχρονα υποστηρίζει και την απευθείας ανταλλαγή ιδιωτικών μηνυμάτων μεταξύ των χρηστών (direct messages). Η πλατφόρμα του Twitter προσφέρει αρκετές επιλογές στην μηχανή αναζήτησης που διαθέτει καθώς ο χρήστης μπορεί να αναζητήσει μέσω κειμένου, hashtag ή mention ότι επιθυμεί.

1.3 Δημογραφικά στοιχεία από το Twitter

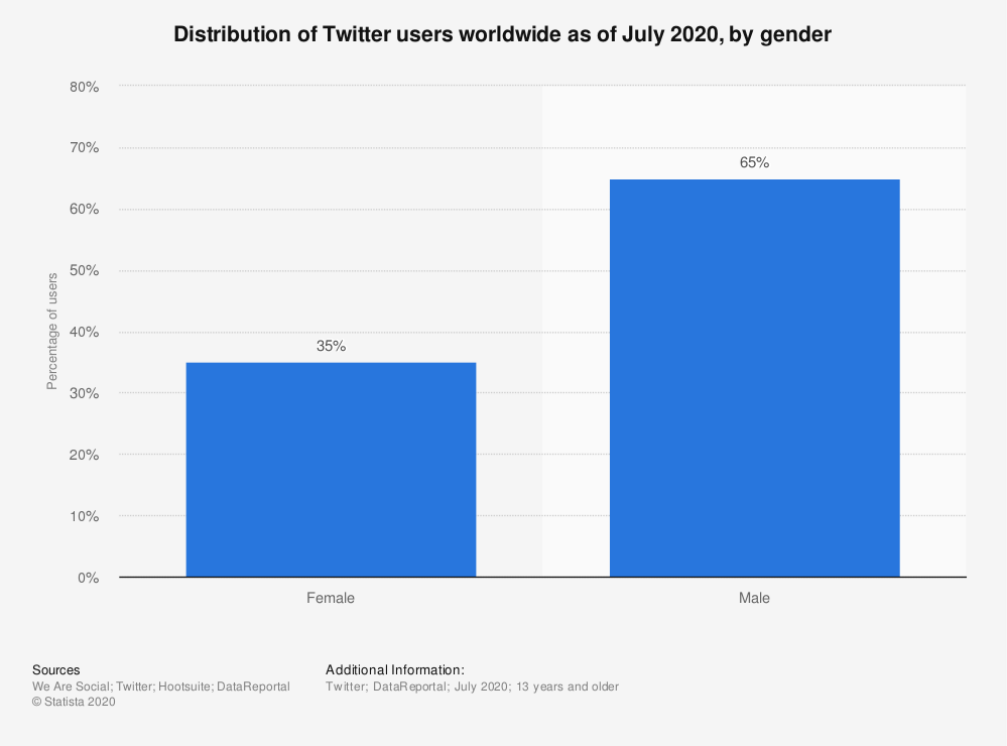
Η ραγδαία εξέλιξη των κοινωνικών δικτύων στην εποχή μας, έχει ωθήσει πολλούς ερευνητές διαφόρων επιστημονικών πεδίων να στρέψουν το ενδιαφέρον τους προς αυτά, ώστε να αναλύσουν να επεξεργαστούν και να αντλήσουν σημαντικές πληροφορίες για τα ποικίλα χαρακτηριστικά των ανθρώπων παγκοσμίως. Η ανάλυση των δεδομένων των OSNs έχει συμβάλλει στην παρακολούθηση των προτιμήσεων των χρηστών, των διάφορων κοινοτήτων καθώς και την πραγματοποίηση προβλέψεων.

Το Twitter, όντας ένα μέσο κοινωνικής δικτύωσης προσφέροντας ποικίλες δυνατότητες προσελκύει συνεχώς νέους χρήστες, οι οποίοι δεν είναι μόνο απλοί χρήστες αλλά και πρακτορεία ειδήσεων, εταιρίες, δημόσιοι φορείς, ομάδες, καλλιτέχνες, σχολεία και κυβερνήσεις με στόχο την ενημέρωση, την προώθηση και την επικοινωνία. Το Twitter αποτελεί μία πηγή πληροφόρησης, η οποία προβάλει ειδήσεις, εκδηλώσεις, καινοτομίες κι έτσι μπορεί να χρησιμοποιηθεί ως εργαλείο από εταιρίες και από ακαδημαϊκές κοινότητες για τη διεξαγωγή ερευνών επιστημονικού και εμπορικού ενδιαφέροντος. Χαρακτηριστικά στοιχεία που δείχνουν τη σημασία μελέτης των δεδομένων του Twitter είναι η δύναμη του και η επιρροή του στους χρήστες αφού αποτελεί τη νούμερο ένα πλατφόρμα αναζήτησης προϊόντων. Το 53% των χρηστών του είναι πιο πιθανό να αγοράσουν πρώτοι κάποιο νέο προϊόν και αναλώνουν το 26% του χρόνου επισκεψιμότητας τους στην πλατφόρμα παρακολουθώντας διαφημίσεις[2]. Συνεπώς μέσω ερευνών και αξιοποιώντας τα δεδομένα που παράγονται από τους χρήστες του Twitter, οι οργανισμοί αποσκοπούν στο να εξάγουν χρήσιμα συμπεράσματα για τα δημογραφικά στοιχεία του πληθυσμού όπως η ηλικία, το φύλο ή η εθνικότητά, καθώς και για τις προτιμήσεις τους, τις αντιδράσεις τους και τα ενδιαφέροντα τους.

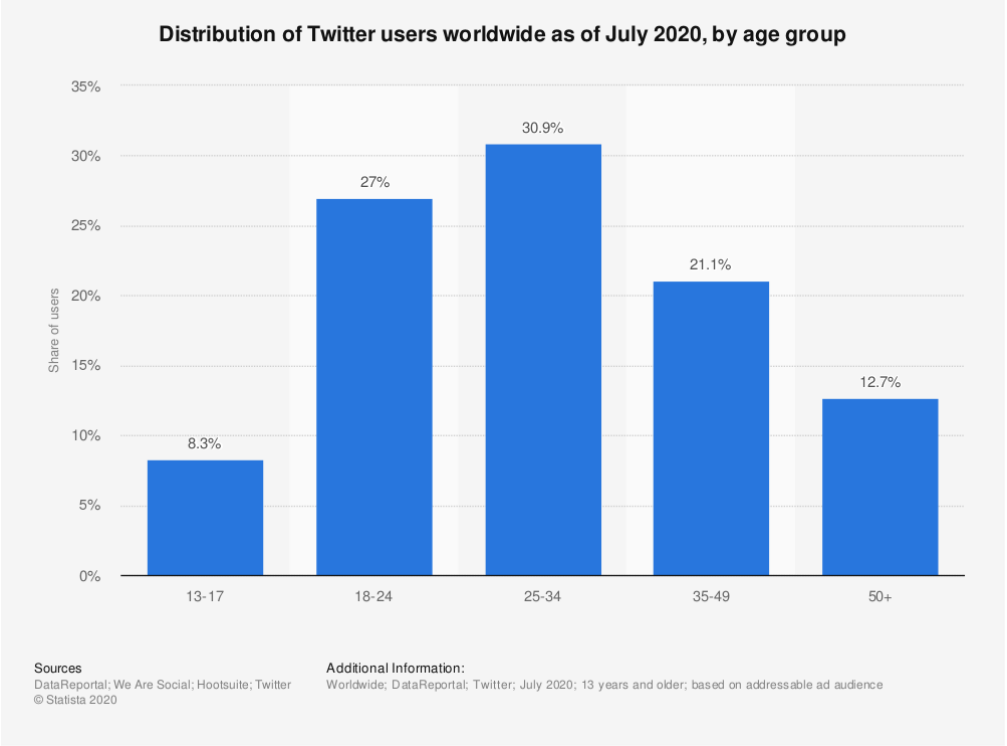
Αρκετά χρήσιμα στοιχεία παρέχονται για τα δημογραφικά χαρακτηριστικά των χρηστών του Twitter και έχουν μεγάλη χρησιμότητα σε πολλούς διαφορετικούς τομείς. Ενδεικτικά, για το φύλο των χρηστών του Twitter παγκοσμίως σύμφωνα με στοιχεία του Statista μέχρι των Ιούλιο του 2020 διαπιστώνεται ότι το 65% είναι άνδρες και το 35% γυναίκες, όπως φαίνεται και στην Εικόνα 1.2[3]. Επιπλέον, ιδιαίτερα σπουδαία είναι και τα στοιχεία που αφορούν την ηλικία των χρηστών και η ανάλυσή τους μπορεί να αποφέρει πολλά ενδιαφέροντα αποτελέσματα. Παρατηρώντας την Εικόνα 1.3[4] που δημοσιεύτηκε στο Statista και παρουσιάζει πληροφορίες μέχρι τον Ιούλιο του 2020 γίνεται σαφές ότι η πλειοψηφία των χρηστών με ποσοστό 30,9% έχουν ηλικία από 25 μέχρι 34 ετών και το νεανικό κοινό με ηλικίες 18 έως 24 αποτελεί το 27% του συνόλου. Σημαντικό είναι και το ποσοστό για τις ηλικίες 35-49 που φθάνει το 21,1%, ενώ για χρήστες άνω των 50 ετών το ποσοστό ανέρχεται στο 12,7%. Οι ανήλικοι χρήστες από 13

[2] Πηγή: https://business.twitter.com/
[3] Πηγή: https://www.statista.com/statistics/828092/distribution-of-users-on-twitter-worldwide-gender/
[4] Πηγή: https://www.statista.com/statistics/283119/age-distribution-of-global-twitter-users/

ετών, που είναι το κατώτατο ηλικιακό όριο ώστε να μπορεί κάποιος να δημιουργήσει λογαριασμό στο Twitter, μέχρι 17 ετών καταλαμβάνουν το μικρότερο ποσοστό με 8,3% του συνόλου.



Εικόνα 1.2: Κατανομή των χρηστών του Twitter παγκοσμίως με βάση το φύλο τους



Εικόνα 1.3: Κατανομή των χρηστών του Twitter παγκοσμίως με βάση την ηλικιακή ομάδα που ανήκουν

Όλα αυτά τα δεδομένα μπορούν να τα εκμεταλλευτούν:

- Εταιρίες ή πανεπιστημιακά ιδρύματα που επιθυμούν να εξάγουν συμπεράσματα για την κοινή γνώμη (ανάλυση συναισθήματος, πολιτική δραστηριότητα), για την εξάπλωση ασθενειών ή για να βελτιώσουν τον χρόνο αντίδρασης σε φυσικές καταστροφές [8], [22].
- Κοινωνικοί επιστήμονες που μελετούν τις αλλαγές της συμπεριφοράς των ανθρώπων εντός των διαδικτυακών κοινωνιών με βάση τα δημογραφικά τους στοιχεία.
- Εταιρίες που λαμβάνοντας υπόψη τα στοιχεία του κοινού που απευθύνονται μπορούν να βελτιώσουν την προώθηση των προϊόντων τους μέσω πιο στοχευμένων διαφημίσεων και μέσω επιλογής κάποιου καταλληλότερου διάσημου προσώπου για να το προβάλουν [12].
- Ψυχολογικές έρευνες που αφορούν άτομα μιας συγκεκριμένης κοινότητας [13].
- Οι έρευνες της αστυνομίας κάθε χώρας που μπορούν να γίνουν πιο αποτελεσματικές χρησιμοποιώντας δημογραφικά χαρακτηριστικά [12].

Ορισμένα από τα πλεονεκτήματα της χρήσης του Twitter για την εξαγωγή δημογραφικών στοιχείων είναι τα εξής:

- Ο μεγάλος αριθμός των χρηστών του και ο μεγάλος όγκος πληροφορίας που παράγουν μπορεί να οδηγήσει σε αρκετά αξιόπιστα συμπεράσματα.
- Με την μεγάλη καθημερινή χρήση της πλατφόρμας τα δεδομένα που παράγονται από τους χρήστες του ανανεώνονται συχνά και αυτό δίνει τη δυνατότητα να ανανεωθούν ταυτόχρονα οι προβλέψεις των δημογραφικών τους χαρακτηριστικών.
- Το περιεχόμενο που παράγεται από τους χρήστες του Twitter και τα δεδομένα τους είναι διαθέσιμα στην ερευνητική κοινότητα και μέσω του API που διαθέτει η ιστοσελίδα είναι εύκολη και δωρεάν η συλλογή μεγάλης ποσότητας δεδομένων.
- Ως μια δημοφιλής πλατφόρμα, έχει κινήσει το ενδιαφέρον αρκετών ερευνητών στο παρελθόν και υπάρχουν αρκετές έρευνες και εφαρμοσμένες μεθοδολογίες με διαθέσιμα τα αποτελέσματα τους, ώστε να μπορεί κάποιος να τα αξιοποιήσει και να τα επεκτείνει διεξάγοντας μια νέα έρευνα.

1.4 Κίνητρο και προκλήσεις

Η πλειονότητα των επιστημονικών ερευνών που μελετούν τα δημογραφικά στοιχεία των ανθρώπων για να εξάγουν πληροφορίες για τη συμπεριφορά τους και επικεντρώνονται στα μέσα κοινωνικής δικτύωσης, δεν λαμβάνονται ιδιαίτερα υπόψιν από την κοινότητα λόγω της απουσίας επαρκών στοιχείων. Πιο συγκεκριμένα, το Twitter παρά το γεγονός ότι παρέχει εύκολη και γρήγορη πρόσβαση στα δεδομένα των χρηστών του, έχει εξαιρέσει κάποια δημογραφικά στοιχεία, όπως η ηλικία και το φύλο, από τα υποχρεωτικά πεδία που πρέπει να συμπληρώσει κάποιος κατά τη δημιουργία ενός προφίλ στην ιστοσελίδα. Αυτό έχει σαν αποτέλεσμα αυτά τα στοιχεία να μην είναι διαθέσιμα και να πρέπει να εξαχθούν με έμμεσους τρόπους. Υπάρχουν αρκετοί τρόποι για την εξαγωγή αυτών των πληροφοριών και μπορεί να πραγματοποιούνται με την ανάλυση των δεδομένων από τα πεδία του προφίλ ενός χρήστη ή με την μελέτη και τη λεξικογραφική ανάλυση των tweets που δημοσιεύει. Διαπιστώνουμε λοιπόν, πως είναι αρκετά δύσκολο να αποκτήσουμε αυτή τη χρήσιμη πληροφορία για τα δημογραφικά στοιχεία των χρηστών του Twitter. Η ηλικιακή ομάδα και κατ' επέκταση η ηλικία είναι ένα από τα θεμελιώδη και σημαντικότερα δημογραφικά χαρακτηριστικά που απασχολούν πολλές μελέτες, οι οποίες επικεντρώνονται στις διαφορές γνωρισμάτων, προτιμήσεων και

ενδιαφερόντων μεταξύ των ανηλίκων, των νέων, των μεσηλίκων και των ηλικιωμένων αλλά και πώς αυτές επηρεάζουν την έρευνα. Στόχος της παρούσας διπλωματικής εργασίας είναι να βάλουμε ένα ακόμη ακρογωνιαίο λίθο σε αυτό το δημοφιλές και γεμάτο προκλήσεις πεδίο έρευνας για την ανίχνευση των χαρακτηριστικών των χρηστών του Twitter και την πρόβλεψη της ηλικίας τους ή της ηλικιακής ομάδας που ανήκουν.

1.5 Αντικείμενο της Διπλωματικής Εργασίας

Η παρούσα διπλωματική επικεντρώνεται στην την ανίχνευση των χαρακτηριστικών των χρηστών του Twitter και την πρόβλεψη της ηλικίας τους ή της ηλικιακής ομάδας που ανήκουν. Λαμβάνοντας υπόψιν τα δεδομένα που εξάγονται από τα tweets, γραμμένα στην αγγλική γλώσσα, που αναρτούν στα προφίλ τους οι χρήστες του Twitter, εφαρμόζονται τεχνικές Επεξεργασίας Φυσικής Γλώσσας (ΕΦΓ) ή Natural Language Processing (NLP) για την ανίχνευση των γλωσσικών χαρακτηριστικών τους [15], [17], των ενδιαφερόντων τους καθώς και των θεμάτων που προτιμούν να σχολιάζουν στα tweet που δημοσιεύουν [14]. Επιπλέον, αξιοποιώντας τα στοιχεία αυτά σε συνδυασμό με τα χαρακτηριστικά που εξάγονται από τα προφίλ τους και είναι ανεξάρτητα από τη γλώσσα [18], εφαρμόζονται και συγκρίνονται ποικίλοι και διαφορετικοί αλγόριθμοι μηχανικής μάθησης για την πρόβλεψη της ηλικίας τους ή της ηλικιακής ομάδας που ανήκουν. Το προφίλ του χρήστη του Twitter περιλαμβάνει πληθώρα πεδίων, όπου επιλέγοντας και χρησιμοποιώντας ορισμένα από αυτά όπως ο αριθμός των followers, των friends, των likes ή το πλήθος των posts, των retweets μπορούν να εξαχθούν χρήσιμα συμπεράσματα για τον χρήστη.

Οι στόχοι της παρούσας διπλωματικής εργασίας είναι οι παρακάτω:

- Ο συνδυασμός των γνωρισμάτων του προφίλ του χρήστη και της εξαγόμενης γνώσης από το περιεχόμενο των tweets που αναρτά [16], να οδηγήσει σε ανταγωνιστικά αποτελέσματα για την πρόβλεψη της ηλικίας του ή της ηλικιακής ομάδας που ανήκει.
- Η επίτευξη μικρής απόκλισης μεταξύ της πρόβλεψης της ακριβούς ηλικίας των χρηστών και της πραγματικής τους που δεν έχει πραγματοποιηθεί σε παρόμοιες μελέτες.
- Ο διαχωρισμός των χρηστών σε 8 (πολλές μικρές) ισομερείς ηλικιακές ομάδες που δεν έχει πραγματοποιηθεί σε παρόμοιες μελέτες και η επίτευξη μιας ανταγωνιστικής απόδοσης για την πρόβλεψη της ηλικιακής ομάδας που ανήκουν.
- Η εξαγωγή συμπερασμάτων για το θέμα του περιεχομένου των tweets που δημοσιεύουν οι χρήστες.

1.6 Οργάνωση Κειμένου

Το υπόλοιπο της παρούσας διπλωματικής είναι οργανωμένο ως εξής:

Στο κεφάλαιο 2 παρουσιάζεται η σχετική βιβλιογραφία πάνω στην οποία βασίστηκε η παρούσα διπλωματική για την εκπόνηση της.

Στο κεφάλαιο 3 παρουσιάζεται αναλυτικά το θεωρητικό υπόβαθρο της διπλωματικής. Αρχικά γίνεται μια εισαγωγή στις τεχνικές Επεξεργασίας Φυσικής Γλώσσας και σε μεθόδους εξαγωγής πληροφοριών για το περιεχόμενο ενός κειμένου. Επιπλέον, αναφέρονται ορισμένα θεμελιώδη στοιχεία για την μηχανική μάθηση και πιο συγκεκριμένα για την παλινδρόμηση και την ταξινόμηση. Ακόμα γίνεται παρουσίαση των αλγορίθμων μηχανικής μάθησης που χρησιμοποιούνται για την πραγματοποίηση των προβλέψεων. Μεταξύ αυτών των αλγορίθμων είναι η γραμμική παλινδρόμηση, οι μηχανές διανυσμάτων υποστήριξης για παλινδρόμηση

(SVR), τα δέντρα απόφασης και τα τυχαία δάση ή ο Xgboost, τα τυχαία δάση, οι μηχανές διανυσμάτων υποστήριξης για ταξινόμηση (SVC), τα δέντρα απόφασης και οι πλησιέστεροι γείτονες. Επιπρόσθετα, στο κεφάλαιο αυτό γίνεται αναφορά στις μετρικές μεθόδους που εφαρμόστηκαν για την αξιολόγηση και την ακρίβεια των αλγορίθμων αυτών όπως το μέσο απόλυτο σφάλμα και η ρίζα της μέσης τετραγωνικής απόκλισηw ή η F1-μετρική και η ακρίβεια (accuracy).

Στο κεφάλαιο 4 υπάρχει η ανάλυση για το τεχνικό υπόβαθρο της διπλωματικής. Αναλυτικότερα, παρουσιάζεται η Python ως επιλεγόμενη γλώσσα για την εκπόνηση της διπλωματικής, οι βιβλιοθήκες της και πληροφορίες για τα εργαλεία και τα περιβάλλοντα που χρησιμοποιούνται για την εκπόνηση της εργασίας. Επίσης, γίνεται αναφορά στα αρχεία CSV (Comma Separated Values - Τιμές διαχωρισμένες με κόμματα) που χρησιμοποιήθηκαν για την αποθήκευση των δεδομένων.

Στο κεφάλαιο 5 περιγράφεται η δομή του προφίλ ενός χρήστη στο Twitter, καθώς και κάποιες σημαντικές αλλαγές σε αυτή τη δομή σε σχέση με το παρελθόν. Επιπροσθέτως, γίνεται ανάλυση και της δομής του tweet και των πεδίων που περιλαμβάνει. Σημαντική είναι και η αναφορά στη διαδικασία που ακολουθήθηκε για την επικοινωνία με το API του Twitter μέσω της βιβλιοθήκης Tweepy με σκοπό την εξόρυξη των δεδομένων, παραθέτοντας και ορισμένα μικρά δείγματα κώδικα.

Στο κεφάλαιο 6 γίνεται ανάλυση του προβλήματος και των εμποδίων που εμφανίστηκαν κατά τη διαδικασία της επίλυσης. Παρουσιάζεται ο σχεδιασμός των μεθόδων που επιλέχθηκαν για την πραγματοποίηση των προβλέψεων, όπου η διαλογή τους βασίστηκε σε στοιχεία που εφαρμόστηκαν σε προσεγγίσεις από τη συναφή βιβλιογραφία.

Στο κεφάλαιο 7 περιγράφεται αναλυτικά η διαδικασία συλλογής και αποθήκευσης των δεδομένων από το Twitter. Παράλληλα, αναλύονται οι μέθοδοι που ακολουθήθηκαν για την επεξεργασία του κειμένου των tweets και την εξαγωγή χρήσιμων πληροφορίων και χαρακτηριστικών από αυτά. Ακόμη, παρουσιάζονται τα γνωρίσματα που εξάγονται από τα μη λεξικογραφικά δεδομένα του προφίλ των χρηστών πάνω στα οποία γίνεται η διαδικασία της προεπεξεργασίας και της μετεπεξαργασίας. Ο συνδυασμός των χαρακτηριστικών που προκύπτουν από τα tweets με αυτά του προφίλ των χρηστών συνθέτουν το τελικό σύνολο δεδομένων που τροφοδοτεί τους αλγορίθμους μηχανικής μάθησης.

Στο κεφάλαιο 8 παρουσιάζονται τα αποτελέσματα των αλγορίθμων μηχανικής μάθησης που εφαρμόστηκαν για την πραγματοποίηση της πρόβλεψης της ηλικίας των χρηστών και της ηλικιακής ομάδας που ανήκουν οι χρήστες. Παράλληλα, προτείνεται το βέλτιστο μοντέλο μηχανικής μάθησης που προέκυψε βάση των μετρικών καθώς και τα πιο σημαντικά χαρακτηριστικά εισόδου για την τροφοδότηση του. Στο τέλος γίνεται μια σύνοψη της πειραματικής μελέτης.

Στο κεφάλαιο 9 παρουσιάζονται τα τελικά συμπεράσματα που προκύπτουν από την παρούσα διπλωματική εργασία και προτείνονται μελλοντικές επεκτάσεις.

Κεφάλαιο 2

2 Συναφής βιβλιογραφία

Το πρόβλημα της ανίχνευσης της ηλικίας ή της ηλικιακής ομάδας που ανήκουν οι άνθρωποι αποτελεί ένα αρκετά ενδιαφέρον και πολυμελετημένο πεδίο έρευνας. Σε αυτό το κεφάλαιο γίνεται αναφορά σε μελέτες που έχουν πραγματοποιηθεί με στόχο να προτείνουν λύσεις για το πρόβλημα. Στην ενότητα 2.1 παρουσιάζεται γενικά το πρόβλημα της ανίχνευσης δημογραφικών στοιχείων από τα μέσα κοινωνικής δικτύωσης ενώ στην ενότητα 2.2 επικεντρωνόμαστε στις μελέτες που έχουν γίνει για την ανίχνευση της ηλικίας των χρηστών του Twitter.

2.1 Ανίχνευση δημογραφικών στοιχείων χρηστών διαφόρων κοινωνικών δικτύων

Οι ερευνητές έχουν διεξάγει πολλές μελέτες τα τελευταία χρόνια, όπου επεξεργάζονται και αξιοποιούν τα δεδομένα των μέσων κοινωνικής δικτύωσης προκειμένου να πραγματοποιήσουν προβλέψεις για τα δημογραφικά στοιχεία των χρηστών τους, όπως είναι το φύλο, η ηλικία ή η κοινωνική ομάδα που ανήκουν. Αντλούν πληροφορίες από τα πιο διαδεδομένα social media όπως το Facebook, το Twitter, το LinkedIn και άλλα. Οι ερευνητές εστιάζουν στην μελέτη των στοιχείων των προφίλ των χρηστών καθώς και των γλωσσολογικών τους χαρακτηριστικών.

Οι συγγραφείς του άρθρου [15] ανέπτυξαν μία τεχνική ανοικτού λεξιλογίου, open-vocabulary, όπως το αναφέρουν, κατά την οποία τα ίδια τα δεδομένα οδηγούν σε μια πιο ολοκληρωμένη εξερεύνηση της γλώσσας που είναι ικανή να διακρίνει τις διαφορές των ανθρώπων και να βρει διασυνδέσεις που οι παραδοσιακές μέθοδοι ανάλυσης κλειστών-λεξιλογίων (closed-vocabulary) αδυνατούν. Το σκεπτικό των ερευνητών βασίστηκε στο γεγονός πως εξετάζοντας τις λέξεις που χρησιμοποιούν οι χρήστες δίνεται η δυνατότητα για καλύτερη κατανόηση της ανθρώπινης ψυχολογίας. Η έρευνα πραγματοποιήθηκε λαμβάνοντας δεδομένα κειμένου από το Facebook με τη βοήθεια χρηστών εθελοντών οι οποίοι απάντησαν και σε ερωτηματολόγια σχετικά με την προσωπικότητά τους. Ορισμένοι περιορισμοί που ελήφθησαν από τους συγγραφείς, ώστε να υπάρχουν επαρκή και αντιπροσωπευτικά δεδομένα. Τέτοιοι ήταν οι χρήστες να έχουν γράψει τουλάχιστον 1000 λέξεις συνολικά στις δημοσιεύσεις που έχουν αναρτήσεις και να είναι κάτω των 65 ετών. Για την εξαγωγή των features βασίστηκαν σε απλές λέξεις ή μικρές φράσεις μήκους μίας έως τριών λέξεων χρησιμοποιώντας φίλτρα που αναγνωρίζουν την ύπαρξη emoticons και συμφράσεων. Επίσης, τα features περιλαμβάνουν και topics δηλαδή ομάδων λέξεων (word clusters) που έχουν ένα κοινό θέμα (topic), όπου το topic modelling υλοποιήθηκε με τον αλγόριθμο Latent Dirichlet Allocation (LDA) που αναλύεται στην ενότητα 3.4.3 της παρούσας εργασίας. Τα αποτελέσματα του open-vocabulary για το φύλο των χρηστών δημιούργησαν δύο μεγάλες ομάδες λέξεων που περιλάμβαναν εκφράσεις που διαχωρίζουν γυναίκες και άνδρες. Η πρόβλεψη του φύλου σημείωσε accuracy περίπου 90% και έγινε με τον αλγόριθμο ταξινόμησης SVM που περιγράφεται στην ενότητα 3.3.3. Όσον αφορά, την ηλικία των χρηστών, η τεχνική αυτή τους χώρισε σε τέσσερις ομάδες με βάση την κατηγοριοποίηση των λέξεων που τις χαρακτηρίζουν. Οι ηλικιακές ομάδες ήταν 13 με 18, 19 με 22, 23 με 29 και 30 έως 65 ετών και για την πρόβλεψη εφαρμόστηκε ο αλγόριθμος της

παλινδρόμησης ridge που αναλύεται στην ενότητα 3.2.3. Η αξιολόγηση των αποτελεσμάτων έγινε με τη μέθοδο του συντελεστή προσδιορισμού R-squared που αναφέρεται στην ενότητα 3.5.2.5, ο οποίος υπολογίστηκε 0.8 περίπου. Παρομοίως έγινε και ο διαχωρισμός για τα στοιχεία της προσωπικότητας των χρηστών. Οι τέσσερις κατηγορίες που δημιουργήθηκαν ήταν η εξωστρέφεια, η εσωστρέφεια, ο νευρωτισμός και η συναισθηματική σταθερότητα. Η κατάταξη των χρηστών έγινε επίσης με την παλινδρόμηση ridge και ο συντελεστής προσδιορισμού ήταν περίπου 0.35 κατά μέσο όρο για τα τέσσερα χαρακτηριστικά.

Το 2015 οι Quan Fang, Jitao Sang, Changsheng Xu, και M. Shamim Hossain με την έρευνά τους [67] επιδίωξαν την ανίχνευση έξι δημογραφικών στοιχείων των χρηστών της πλατφόρμας Google+. Πιο συγκεκριμένα θέλησαν να προβλέψουν το φύλο, την ηλικία, το επάγγελμα, τα ενδιαφέροντα, τη συζυγική κατάσταση και το συναισθηματικό προσανατολισμό των χρηστών. Για την διεξαγωγή της μελέτης χρησιμοποίησαν 20000 προφίλ τόσο διάσημων όσο και γενικών χρηστών, διότι τα δεδομένα τους παρουσιάζουν διαφορές λαμβάνοντας τα δεδομένα από το API που παρείχε η πλατφόρμα. Τα χαρακτηριστικά που εξέτασαν και είχαν ως είσοδο για τους αλγορίθμους ήταν τα στοιχεία του προφίλ, οι φωτογραφίες προφίλ, το βιογραφικό και οι 500 πιο πρόσφατες δημοσιεύσεις που είχαν αναρτήσει, λαμβάνοντας υπόψη τις φωτογραφίες, το κείμενο ή τους συνδέσμους που περιείχαν. Τέθηκαν κάποιοι περιορισμοί ώστε να υπάρχει ικανό και ίδιας σημασίας λεξικογραφικό δείγμα για κάθε χρήστη και έτσι μελετήθηκαν άτομα που είχαν κάνει τουλάχιστον 20 αναρτήσεις. Οι χρήστες χωρίστηκαν σε δύο ηλικιακές κατηγορίες, οι οποίες ήταν νέοι, δηλαδή άτομα κάτω των 30 ετών, και μεγάλοι, δηλαδή χρήστες των 30 ετών και άνω. Για το επάγγελμα χρησιμοποιήσαν 15 ομάδες η μία εκ των οποίων περιλάμβανε τους μαθητές, ενώ όρισαν δύο κατηγορίες έγγαμος και άγαμος για τη συζυγική κατάσταση. Όσον αφορά τα ενδιαφέροντα των χρηστών, θέλησαν να τους ταξινομήσουν σε 12 διαφορετικές κατηγορίες. Τρεις κατηγορίες, δηλαδή θετικός, αρνητικός και ουδέτερος επιλέχθηκαν για τον συναισθηματικό προσανατολισμό. Για την πραγματοποίηση των προβλέψεων εφαρμόστηκε ο αλγόριθμος SVM, ο Stacked SVM, ενώ προτάθηκε και μία νέα προσέγγιση, ο Relational LSVM που αξιοποιεί και τις συσχετίσεις μεταξύ των χαρακτηριστικών των χρηστών και ενισχύει την ανίχνευση των δημογραφικών τους στοιχείων. Ο SVM χρησιμοποιήθηκε για την πρόβλεψη μέσω χρήσης μεμονωμένων χαρακτηριστικών όπως η φωτογραφία προφίλ ή τα μονογράμματα (unigrams) ή τα λεξικογραφικά στοιχεία. Την καλύτερη επίδοση για την πρόβλεψη κάθε δημογραφικού χαρακτηριστικού είχε ο προτεινόμενος Relational LSVM. Πιο συγκεκριμένα σημείωσε περίπου 72% accuracy στην πρόβλεψη της ηλικίας, περίπου 80% στην πρόβλεψη του φύλου και 62% τόσο στην πρόβλεψη της συζυγικής κατάστασης, όσο και στην πρόβλεψη για τα ενδιαφέροντα. Μικρότερη επιτυχία είχε στο επάγγελμα όπου κατέγραψε accuracy 25% και στο συναισθηματικό προσανατολισμό όπου είχε 41% επιτυχία.

Οι Jorge Brea, Javier Burroni και Carlos Sarraute το 2015 πραγματοποίησαν μία μελέτη με σκοπό τον διαχωρισμό σε ηλικιακές ομάδες, ενός πλήθους χρηστών κινητών τηλεφώνων ανάλογα με τη χρήση που κάνανε [68]. Οι πληροφορίες που επεξεργάστηκαν ήταν συνδυασμός ενός συνόλου δεδομένων που προήλθαν από ένα πάροχο υπηρεσιών κινητής τηλεφωνίας από μία χώρα της Λατινικής Αμερικής και δεδομένων σχετικών με τα δημογραφικά στοιχεία του πληθυσμού. Περιείχαν πληροφορίες για τις κλήσεις που είχαν πραγματοποιήσει και για τα μηνύματα (SMS) που είχαν αποστείλει. Για την επίλυση του προβλήματος κατασκευάστηκε ένας γράφος με 70 εκατομμύρια κορυφές και 250 εκατομμύρια ακμές ώστε να εκφραστούν οι συσχετίσεις μεταξύ των ηλικιών και των δεδομένων. Προέκυψαν 4 ηλικιακές ομάδες οι οποίες

ήταν άτομα κάτω των 24 ετών, 25 έως 34 , 35 έως 50 και χρήστες άνω των 50. Η τελική επίδοση ήταν 81% επιτυχία για τιμή 0,55 του διανύσματος πιθανότητας.

Οι Marco Pennacchiotti και Ana-Maria Popescu στην έρευνα που διενέργησαν το 2011 [69] επιχείρησαν την κατηγοριοποίηση των χρηστών του Twitter σχετικά με τις πολιτικές τους πεποιθήσεις, την εθνικότητά τους και αν τους αρέσουν τα καταστήματα Starbucks. Επικεντρώθηκαν αρχικά στην συμπεριφορά του χρήστη στο Twitter, δηλαδή στη συχνότητα που κάνει δημοσιεύσεις, στο συνολικό αριθμό των tweets και των retweets του και στο αν δημοσιεύει tweets με σκοπό να απαντήσει σε άλλους χρήστες. Ακόμη, οι ερευνητές έδωσαν βάση στο περιεχόμενο των tweets, δηλαδή στο γλωσσικό επίπεδο και τις λέξεις-εκφράσεις που χρησιμοποιεί, στην ανάλυση αισθήματος (sentiment analysis) των κειμένων καθώς και στα θέματα που τον ενδιαφέρουν. Το κοινωνικό δίκτυο του χρήστη, δηλαδή ποιους ακολουθεί ή ποιοι τον ακολουθούν καθώς και σε ποιους απαντά ή ποιων τα posts αναδημοσιεύει, αποτέλεσε επίσης ένα αντικείμενο που εξέτασαν. Σημαντικά features ήταν και τα στοιχεία του προφίλ όπως το όνομα, η τοποθεσία, το βιογραφικό, ο αριθμός των tweets, των φίλων και των ακολούθων του. Για την ανάλυση των λέξεων ως προς τα συναισθήματα χρησιμοποιήθηκαν τρεις κατηγορίες που ήταν θετικές, αρνητικές και ουδέτερες λέξεις. Ο αλγόριθμος LDA και κάποιες παραλλαγές του εφαρμόστηκαν για την μελέτη των κειμένων, ώστε να ομαδοποιηθούν σημασιολογικά οι λέξεις και να εξαχθούν παραπάνω πληροφορίες σχετικές με τις απόψεις και τις προτιμήσεις των χρηστών. Οι χρήστες χωρίστηκαν σε δημοκράτες (democrats) και ρεπουμπλικανούς (republicans) σχετικά με τις πολιτικές τους πεποιθήσεις, σε αφροαμερικανούς (African-Americans) ή όχι όσον αφορά την εθνικότητα τους και σε άτομα που τους αρέσουν ή όχι τα Starbucks. Ο αλγόριθμος ταξινόμησης Gradient Boosted Decision Trees (GBDT) εκτελέστηκε για την εξαγωγή των προβλέψεων. Οι μετρικές accuracy, precision, recall και F1-score, που περιγράφονται αναλυτικά στην ενότητα 3.5.3, χρησιμοποιήθηκαν για την αξιολόγηση του μοντέλου. Το πρόβλημα της πρόβλεψης των πολιτικών πεποιθήσεων παρουσίασε περίπου 88% accuracy. Η πρόβλεψη για την εθνικότητα σημείωσε τιμές περίπου 0,65 για τις μετρικές precision, recall και F1-score, ενώ για τις ίδιες μετρικές στο πρόβλημα των ατόμων που τους αρέσουν τα Starbucks η τιμή ήταν 0,76 περίπου.

Το 2010 οι Delip Rao, David Yarowsky, Abhishek Shreevats και Manaswi Gupta [71] επιχείρησαν να μελετήσουν τον προσδιορισμό του φύλου, της ηλικίας, της καταγωγής και των πολιτικών πεποιθήσεων των χρηστών του Twitter. Για τη μελέτη του φύλου δημιούργησαν ένα σύνολο δεδομένων που περιλάμβανε 500 άντρες και 500 γυναίκες ταυτοποιημένους χρήστες, έχοντας για πηγές αντρικές και γυναικείες πανεπιστημιακές αδελφότητες, αλλά και αντρικά και γυναικεία προϊόντα υγιεινής. Για τη μελέτη της ηλικίας, οι χρήστες χωρίστηκαν σε αυτούς που ήταν κάτω των 30 ετών και σε αυτούς που ήταν άνω των 30 ετών, έχοντας σε κάθε κατηγορία από 1000 άτομα. Τα δεδομένα ηλικίας ελήφθησαν μέσω άλλων μέσων κοινωνικής δικτύωσης, όπως το MySpace και το LinkedIn, τα οποία είχαν επισυνάψει οι χρήστες ως επιπλέον πληροφορίες στο προφίλ τους στο Twitter. Οι χρήστες χωρίστηκαν σε άτομα από τη βόρεια και τη νότια Ινδία για το πρόβλημα της καταγωγής και σε δημοκρατικούς και ρεπουμπλικανούς για το πρόβλημα των πολιτικών πεποιθήσεων. Οι ερευνητές εξέτασαν ως πιθανά features τη δομή του δικτύου ενός χρήστη, δηλαδή τους followers και τους following, αλλά και τη συμπεριφορά του στην επικοινωνία, δηλαδή τις απαντήσεις, τα retweets και το πόσο συχνά δημοσιεύει κάτι. Επίσης, αξιοποίησαν ως δεδομένα εισόδου και τη χρήση emoticons ή αποσιωπητικών και την επανάληψη ίδιων γραμμάτων σε μια λέξη. Τα πειράματα έγιναν με τρία SVM μοντέλα. Το πρώτο είχε ως είσοδο τα κοινωνιογλωσσικά (sociolinguistic) χαρακτηριστικά του τρόπου γραφής των χρηστών, το δεύτερο, το περιεχόμενο του κειμένου

των tweets (unigrams, bigrams) και το τρίτο, τις προβλέψεις των δυο προηγούμενων μαζί με τα αντίστοιχα βάρη τους. Για το πρόβλημα του φύλου είχε την καλύτερη ακρίβεια με περίπου 72% μέσω του συνδυαστικού μοντέλου. Στο πρόβλημα της ηλικίας 74% το συνδυαστικό μοντέλο κατέγραψε ακρίβεια περίπου 74%. Για την καταγωγή το καλύτερο μοντέλο ήταν το γλωσσολογικό που είχε 77% accuracy, ενώ το ngram μοντέλο παρουσίασε περίπου 82% ακρίβεια στην εργασία για τις πολιτικές πεποιθήσεις.

Οι συγγραφείς του άρθρου [18] το 2014 θέλησαν να επισημάνουν τη δυσκολία της ανίχνευσης του φύλου και της ηλικίας από το περιεχόμενο των tweets. Πραγματοποίησαν ένα crowdsourcing πείραμα με τη μορφή online παιχνιδιού ζητώντας από τους παίχτες να μαντέψουν το φύλο και την ηλικία ενός συνόλου χρηστών του Twitter διαβάζοντας ορισμένα tweets τους. Το πείραμα διεξήχθη σε μία βάση που αποτελούνταν από 3000 Ολλανδούς χρήστες του Twitter, από τους οποίους οι 200 επιλέχθηκαν τυχαία ως test set για το online παιχνίδι και τη μέτρηση της απόδοσης του μοντέλου. Οι ερευνητές τόνισαν, ότι ο τρόπος κατασκευής του συνόλου δεδομένων με τους χρήστες είναι μεγάλης σημασίας, διότι πρέπει να περιλαμβάνει όσο το δυνατόν πιο αντιπροσωπευτικό δείγμα του γενικού πληθυσμού. Επίσης, χρησιμοποίησαν κάθε πιθανή πληροφορία που συλλέξανε από τα tweets καθώς και από τα προφίλ των χρηστών σε άλλα μέσα κοινωνικής δικτύωσης όπως το Facebook και το LinkedIn. Στο δείγμα τέθηκε περιορισμός για την ηλικία των χρηστών με μικρότερη τιμή τα 8 έτη και μεγαλύτερη τα 80 και χρησιμοποιήθηκαν από 20 έως 40 tweets για κάθε χρήστη. Δόθηκε έμφαση και στις λεξικογραφικές συνήθειες των χρηστών ώστε να επισημανθεί ότι η ηλικία τους μπορεί να διαφέρει από τον τρόπο γραφής τους, το μήκος των tweets τους και τις λέξεις που χρησιμοποιούν. Το μοντέλο της γραμμικής παλινδρόμησης που δημιουργήθηκε για την ηλικία σημείωσε για τη μετρική MAE τιμή περίπου 6,1 χρόνια, η οποία περιγράφεται στην ενότητα 3.5.2.1, ενώ οι προβλέψεις των παικτών είχαν MAE 4,8 περίπου. Στο πρόβλημα προσδιορισμού του φύλου το μοντέλο Logistic Regression που προτάθηκε παρουσίασε περίπου 69% ακρίβεια, ενώ το κοινό είχε ακρίβεια περίπου 84% στις προβλέψεις του. Τελικά, το συμπέρασμα που εξήγαγαν οι ερευνητές είναι ότι η συνήθης προσέγγιση με αλγορίθμους μηχανικής μάθησης προβλέπει το φύλο και την ηλικία για τους περισσότερους χρήστες μαθαίνοντας μια στερεοτυπική συμπεριφορά και παρέχοντας μια απλή ανάλυση για τον ορισμό των δύο εννοιών. Αυτός ήταν ο λόγος που πρότειναν η πρόβλεψη του φύλου και της ηλικία να αντιμετωπίζονται σε παρόμοιες έρευνες ως κοινωνικές μεταβλητές και όχι ως στατικές βιολογικές μεταβλητές.

Οι Molteni και Ponce De Leon το 2016 [72] εκμεταλλευόμενοι δεδομένα από το Twitter, ανέλυσαν την προτίμηση των χρηστών του Twitter και την τηλεθέαση διαφόρων τηλεοπτικών προγραμμάτων. Ο κύριος όγκος των δεδομένων που χρησιμοποίησαν αφορούσαν το πλήθος και το περιεχόμενο των tweets που δημοσίευσαν σχετικά με τα τηλεοπτικά προγράμματα. Επιπλέον, είχαν ως δεδομένα εισόδου την ημερομηνία και την τοποθεσία που έγιναν τα tweets. Ομαδοποίησαν τα δεδομένα με βάση τις ομοιότητές τους και έπειτα εφάρμοσαν ένα μοντέλο πρόβλεψης για την κάθε ομάδα προγραμμάτων. Για την πραγματοποίηση των πειραμάτων, εφαρμόστηκε η γραμμική παλινδρόμηση, που αναλύεται στην ενότητα 3.2.1, σε συνδυασμό με κάποιες μετρικές ανάλυσης συναισθήματος (sentiment analysis). Για την ανάλυση συναισθήματος επιλέχθηκαν τρεις κατηγορίες, θετικό, αρνητικό και ουδέτερο. Ο συντελεστής προσδιορισμού R-squared κυμάνθηκε μεταξύ 0,73 και 0,94 για την κάθε ομάδα.

Το 2017 οι Alfonso Crisci, Valentina Grasso, Paolo Nesi, Gianni Pantaleo, Irene Paoli, Imad Zaza [73] βασίστηκαν σε δεδομένα που εξόρυξαν από το Twitter για να μελετήσουν τις

προτιμήσεις των τηλεθεατών σχετικά με τηλεοπτικές εκπομπές reality. Τα δεδομένα που επεξεργάστηκαν αφορούσαν το πλήθος των δημοσιεύσεων και των αναδημοσιεύσεων (retweets), ενώ στάθηκαν και στο πλήθος των μοναδικών χρηστών που έκαναν retweets σχετικά με αυτά τα προγράμματα. Επίσης, επεξεργάστηκαν και τις λέξεις ή τις λέξεις κλειδιά που ήταν δημοφιλείς στις αναζητήσεις. Ακόμη, έκαναν ανάλυση συναισθήματος για το περιεχόμενο των κειμένων που αναρτήθηκαν μέσω τεχνικών NLP. Για την δημιουργία των μοντέλων πρόβλεψης εφάρμοσαν διάφορες μεθόδους παλινδρόμησης, όπως η γραμμική, η Ridge, η Lasso και η ElasticNet. Η έρευνα τους σημείωσε ακρίβεια για τις προβλέψεις τους μεταξύ 80-94%.

2.2 Ανίχνευση ηλικιακής ομάδας χρηστών του Twitter

Ειδικότερα τα τελευταία χρόνια οι μελέτες και οι προσπάθειες για την ανίχνευση της ηλικίας των χρηστών του Twitter έχουν ενταθεί, καθώς όλο και περισσότεροι ερευνητές ασχολούνται με αυτό το σημαντικό πρόβλημα. Σκοπός τους είναι να εκμεταλλευτούν όλα τα χρήσιμα δεδομένα που δημιουργούνται από την ενασχόληση του χρήστη στην πλατφόρμα και σχετίζονται τόσο με τα στοιχεία του προφίλ του που αναλύονται στην ενότητα 5.1, όσο και με τα γλωσσολογικά δεδομένα που ανακύπτουν από τα κείμενα των tweets που δημοσιεύει ο κάθε χρήστης. Υπάρχουν αρκετές προσεγγίσεις για το πρόβλημα της ανίχνευσης της ηλικίας του χρήστη. Κάποιες από αυτές εστιάζουν μόνο στα στοιχεία του προφίλ του. Επίσης, ορισμένες επικεντρώνονται στα γλωσσικά χαρακτηριστικά του χρήστη, αφού εξάγουν την πληροφορία από το βιογραφικό του και από τα tweets που δημοσιεύει για να πραγματοποιήσουν τις προβλέψεις τους, ανάγοντας έτσι το πρόβλημα στην κατηγορία των προβλημάτων Επεξεργασίας Φυσικής Γλώσσας (Natural Language Processing - NLP). Ωστόσο, υπάρχουν και μελέτες που συνδυάζουν τις δύο προηγούμενες προσεγγίσεις σε μία λύση ώστε να κάνουν τις προβλέψεις τους.

Οι R. G. Guimarães, R. L. Rosa, D. De Gaetano, D. Z. Rodríguez και G. Bressan στην έρευνα τους το 2017 [70] επιχείρησαν να ομαδοποιήσουν ηλικιακά τους χρήστες του Twitter σε δύο κατηγορίες, έφηβους (teenagers) από 13 έως 20 ετών και ενήλικες (adults) 20 ετών και άνω. Η προσέγγισή τους περιλάμβανε τη μελέτη του προβλήματος τόσο με αλγορίθμους μηχανικής μάθησης όσο και με νευρωνικά δίκτυα. Το σύνολο των δεδομένων αποτελούνταν από 7000 προτάσεις οι οποίες συλλέχθηκαν μέσω αναζήτησης με λέξεις-κλειδιά στα μέσα κοινωνικής δικτύωσης, όμως τελικά θεωρήθηκαν ως έγκυρες και χρησιμοποιήθηκαν οι 6280 από αυτές. Για τα πειράματα τα δεδομένα χωρίστηκαν σε 80% training set για την εκμάθηση του μοντέλου και σε 20% test set για την αξιολόγηση της απόδοσης του. Τα χαρακτηριστικά που συνέθεσαν την είσοδο σχετίστηκαν με τα στοιχεία του προφίλ του χρήστη και με τις λεξικογραφικές του συνήθειες. Έτσι ορισμένα features ήταν το φύλο, ο αριθμός των tweets που έχουν ποστάρει, το πλήθος των ατόμων που ακολουθούν και ο αριθμός των followers τους. Σχετικά με τον τρόπο γραφής των χρηστών το μοντέλο επεξεργάστηκε τη χρήση των tags-mentions (σύμβολο @), των hashtags και των σημείων στίξης (punctuation). Ακόμη, μερικά χαρακτηριστικά ήταν το μήκος των μηνυμάτων, η χρήση συντομογραφιών (abbreviations), η αργκό γλώσσα (slang) και αν η δημοσίευση είναι retweet. Σημαντικό feature για τη μελέτη θεωρήθηκε και το θέμα (topic) του κειμένου καθώς και αν περιέχει συνδέσμους (URL) για ανακατεύθυνση σε άλλη σελίδα. Η πειραματική διαδικασία περιλάμβανε την εκπαίδευση τριών αλγορίθμων ταξινόμησης και δύο νευρωνικών δικτύων. Οι αλγόριθμοι μηχανικής μάθησης που χρησιμοποιήθηκαν ήταν τα δέντρα απόφασης, ο SVM και τα τυχαία δάση. Όσον αφορά τα νευρωνικά δίκτυα υλοποιήθηκε ένα Multilayer Perceptron (MLP) νευρωνικό δίκτυο και ένα βαθύ συνελικτικό νευρωνικό

δίκτυο (DCNN). Η απόδοση των μοντέλων αξιολογήθηκε με τις μετρικές Precision, Recall και F1-score. Για τη βελτίωση της απόδοσης εφαρμόστηκε ένας αλγόριθμος δέντρων απόφασης, ώστε να επιλεχθούν οι πιο σχετικές παράμετροι στην εκπαίδευση και να μη ληφθούν υπόψη αυτές με το πιο χαμηλό βάρος. Για τους τρεις αλγορίθμους ταξινόμησης το F1-score έλαβε την ίδια τιμή 0,85 και για το MLP η επίδοση ήταν 0,88. Το καλύτερο μοντέλο ήταν αυτό του DCNN που κατέγραψε τιμή 0,95 στο F1-score. Τέλος, για την επιβεβαίωση της χρησιμότητας του προτεινόμενου μοντέλου δοκιμάστηκε μία enhanced Sentiment Metric (eSM). Σκοπός ήταν να επισημάνει τις διαφορές στα αποτελέσματα της πρόβλεψης με την εφαρμογή ή όχι του μοντέλου, όπου καταγράφηκε σαφώς καλύτερη επίδοση σε περιπτώσεις χρήσης του με τιμή MAE 0,10 αντί για 0,15, τιμή μεγίστου λάθους 0,15 αντί για 0,19 και τιμή RMSE 0,25 αντί για 0,29 χωρίς αυτό. Αυτό κρίθηκε εξαιρετικά σπουδαίο, παρότι πολλές φορές τα δεδομένα ηλικίας δεν είναι διαθέσιμα στα social media το μοντέλο αυξάνει σημαντικά την eSM.

Οι Antonio Morgan-Lopez, Annice Kim, Robert Chew, Paul Ruddle το 2017 [74] δοκίμασαν μέσω της έρευνάς τους, να πραγματοποιήσουν προβλέψεις για τις ηλικιακές ομάδες που ανήκουν οι χρήστες στην πλατφόρμα του Twitter με σκοπό να παρέχουν περισσότερες πληροφορίες σε ιατρικούς οργανισμούς. Για την διεξαγωγή της έρευνας βασίστηκαν στα γλωσσολογικά στοιχεία και στις πληροφορίες του προφίλ των χρηστών. Συνέλεξαν μέσω του Twitter API, τα 200 πιο πρόσφατα tweets περίπου 3200 χρηστών σε συνδυασμό με τα στοιχεία των προφίλ τους. Όσον αφορά τα μεταδεδομένα (metadata) του προφίλ, δημιούργησαν 21 features από αυτά. Κάποιες ενδεικτικές πληροφορίες που αξιοποίησαν είναι η συχνότητα που δημοσιεύουν, ο αριθμός των followers, ο αριθμός των ατόμων που ακολουθούν (followings) καθώς και το λόγο followers-followings. Επιπλέον, δημιούργησαν 23 features με βάση τα λεξικογραφικά δεδομένα του χρήστη. Για τα δεδομένα τις ηλικίας εξόρυξαν δημοσιεύσεις σχετικές με γενέθλια ή σχετικές ανακοινώσεις. Οι χρήστες ταξινομήθηκαν σε τρεις κατηγορίες τους νέους (youth) από 13 έως 17 ετών, τους νέους ενήλικες (young adults) από 18 έως 24 ετών και τους ενήλικες (adults) με ηλικία άνω των 25 ετών. Στην έρευνα εξετάστηκαν τέσσερα μοντέλα, ένα που περιλάμβανε μόνο τα γλωσσικά δεδομένα, ένα που περιλάμβανε μόνο τα στοιχεία του προφίλ, ένα συνδυαστικό που περιείχε τα γλωσσικά και τα metadata ως χαρακτηριστικά μαζί και ένα με το λεξικό που δημιουργήθηκε από τους συγγραφείς του άρθρου [15]. Η πειραματική μελέτη έγινε με την εκτέλεση έξι αλγορίθμων ταξινόμησης, την λογιστική παλινδρόμηση, τα SVM, τα τυχαία δάση, τον adaBoost, τα extra trees και έναν dummy ταξινομητή. Εφαρμόστηκαν επίσης, οι τεχνικές βελτίωσης υπερπαραμέτρων και σπουδαιότητας των χαρακτηριστικών που περιγράφονται στις ενότητες 3.5.1.3 και 3.5.1.2 αντίστοιχα, μέσω των μοντέλων παλινδρόμησης Ridge, Lasso και ElasticNet αλλά και αλγορίθμων δέντρων. Αυτές συνέβαλλαν στην επιλογή του βέλτιστου μοντέλου, την αποφυγή overfitting και ανέδειξαν την ηλικία του λογαριασμού του χρήστη στην πλατφόρμα ως το πιο σημαντικό feature για την πρόβλεψη της ηλικίας των νέων με τιμή 0,336 για τον Cohen’s d συντελεστή. Το συνδυαστικό μοντέλο, που εξήχθη με την λογιστική παλινδρόμηση, είχε την καλύτερη απόδοση και αξιολογήθηκε με τις μετρικές Precision, Recall και F1-score όπου πέτυχε 74% και στις τρεις. Αντίθετα, το μοντέλο που χρησιμοποίησε μόνο τα δεδομένα του προφίλ σημείωσε 58% Precision, 60% Recall και 58% F1-score.

Το 2015 οι Luke Sloan, Jeffrey Morgan, Pete Burnap, Matthew Williams [75] θέλησαν να εξορύξουν δημογραφικά στοιχεία σχετικά με την ηλικία, το επάγγελμα και την κοινωνική τάξη κατοίκων του Ηνωμένου Βασιλείου, μελετώντας τα προφίλ τους στο Twitter. Για αυτό το λόγο έλαβαν και επεξεργάστηκαν τα tweets και το βιογραφικό κάθε χρήστη. Εστιάζοντας στην προσέγγιση τους για τον προσδιορισμό της ηλικίας παρατηρούνται αρκετά ενδιαφέροντα

συμπεράσματα. Από ένα σύνολο δεδομένων περίπου 390000 χρηστών κατάφεραν να βρουν και να ταυτοποιήσουν περίπου το 0,37%, δηλαδή 1470 χρήστες, με την ηλικία τους. Η πρόβλεψη έγινε με την λεξικογραφική μελέτη των βιογραφικών στο Twitter. Αναζήτησαν και δοκίμασαν διάφορα μοτίβα με φράσεις σχετικές με ημερομηνίες, χρονολογίες και ηλικίες για να κατασκευάσουν τον ανιχνευτή της ηλικίας. Για παράδειγμα έψαχναν για ημερομηνίες κάποια από τις μορφές "DD/MM/YYYY" ή "DD/MM/YY". Δημιούργησαν bigrams για λέξεις που αναφέρονται στην ηλικία για να εξάγουν την εννοιολογική σημασία της έκφρασης συνολικά μαζί με μορφές ηλικίας ή χρονολογίας. Έτσι όρισαν τρία είδη φράσεων με τις μορφές "Born in X", "I am X years old" και "X years old" και έκαναν τις αναζητήσεις τους στηριζόμενες σε αυτές. Ωστόσο ,πολλές φορές μπορεί το νόημα και η μορφή των φράσεων να διαφέρουν και να μην εξασφαλίζουν την εξαγωγή της ηλικίας. Για παράδειγμα η πρόταση "17 years working as a lecturer" περιγράφει τα χρόνια που ασκεί την επαγγελματική του ιδιότητα και όχι την ηλικία του. Το γεγονός αυτό τους ώθησε να λάβουν κάποιους επιπλέον περιορισμούς και να ορίσουν μοτίβα που δηλώνουν ταυτοποίηση ηλικία και κανόνες που ταυτοποιούν κάποιο ρόλο. Στην Εικόνα 2.1[5] και στην Εικόνα 2.2[6] παρακάτω απεικονίζονται οι εκφράσεις που χρησιμοποίησαν για κάθε περίπτωση. Σύμφωνα με στοιχεία που είχαν οι ερευνητές το 2014 για τους χρήστες του Twitter το κατώτερο όριο ηλικίας ήταν τα 13 έτη και το ανώτερο τα 90 έτη ενώ η ηλικιακή τους κατανομή δεν ήταν ομοιόμορφη και εμφάνιζε σημαντικές διαφορές με αυτήν της απογραφής των κατοίκων του Ηνωμένου Βασιλείου. Πιο συγκεκριμένα περίπου το 60% των χρηστών του Twitter είναι από 13 έως 20 ετών, το 35% περίπου είναι μεταξύ 21 και 40 ετών, ενώ μόλις το 5% έχουν ηλικία άνω των 40 ετών. Το γεγονός αυτό επηρέασε σημαντικά τις προβλέψεις τους, αφού τα δεδομένα είναι ελλιπή για τις μεγάλες ηλικίες ενώ αντίθετα είναι πολλά για τους νέους.

Pre-Integer:	Post-Integer:
'age'	*'years old'*
'aged'	*'yrs'*
'I'm'	*'yrs old'*
'I am'	*'years'*
'born'	
'born in'	

doi:10.1371/journal.pone.0115545.t004

Εικόνα 2.1: Κανόνες ταυτοποίησης ηλικίας

***Pre-Integer*:**	***Post-Integer*:**
'for'	*'years as'*
'spent'	*'years working'*
	'years in'
	Any of the post-integer terms listed in Table 4 when followed by 'son', 'daughter'

doi:10.1371/journal.pone.0115545.t005

Εικόνα 2.2: Κανόνες ταυτοποίησης ρόλου

Οι Benjamin Chamberlain, Clive Humby και Marc Deisenroth προσπάθησαν με την έρευνά τους το 2017 [76] να προσδιορίσουν την ηλικία των χρηστών του Twitter στηριζόμενοι στο ποιους λογαριασμούς και ποια άτομα ακολουθούν. Το βασικό τους επιχείρημα ήταν πως οι

[5] Πηγή: https://journals.plos.org/plosone/article/figure?id=10.1371/journal.pone.0115545.t004
[6] Πηγή: https://journals.plos.org/plosone/article/figure?id=10.1371/journal.pone.0115545.t005

χρήστες ακολουθούν λογαριασμούς ή άτομα ανάλογα με τα ενδιαφέροντά τους, τα οποία όμως αποτελούν αρκετά ενδεικτικά στοιχεία για την ηλικία του ανθρώπου. Στο πλαίσιο της έρευνας οι συγγραφείς μελέτησαν 700 εκατομμύρια διαφορετικούς χρήστες ώστε να τονίσουν την κλιμακωσιμότητα της προτεινόμενης λύσης. Ωστόσο, μόνο 133 χιλιάδες χρήστες είχαν φανερή την ηλικία τους, που αποτελεί μόλις το 0,02% του συνολικού δείγματος. Οι ερευνητές ανέπτυξαν έναν web crawler που συνδέθηκε με το Twitter API και κατέβασαν το βιογραφικό των 700 εκατομμυρίων χρηστών. Τα κείμενα της περιγραφής έλαβαν τη μορφή ευρετηρίων με τη χρήση του Apache SOLR αναζητήθηκαν μοτίβα φράσεων σχετικά με την ηλικία ώστε να εξαχθεί η πληροφορία. Για να αντιμετωπίσουν το πρόβλημα παλαιών δεδομένων και να περιόρισαν τη μελέτη μόνο σε ενεργούς λογαριασμούς, δηλαδή σε προφίλ που είχαν κάνει δημοσίευση τους τελευταίους τρεις μήνες. Τα features για την εκπαίδευση του μοντέλου αποτέλεσαν τα 103,722 λογαριασμοί του Twitter που ακολουθούνταν από 10 τουλάχιστον χρήστες. Η προσέγγισή τους ήταν να χωρίσουν τους νέους χρήστες κάτω των 18 ετών σε πολλές ομάδες ανά δύο έτη και τους μεγαλύτερους σε ομάδες με μεγαλύτερο εύρος. Έτσι δημιουργήθηκαν 10 ηλικιακές κατηγορίες που ήταν χρήστες μέχρι 11 ετών, έπειτα 12 με 13 ετών, 14 και 15 ετών, 16 έως 17 ετών, άτομα από 18 μέχρι 24 ετών, χρήστες 25 με 34 ετών, 35 ετών έως 44, 45 με 54, 55 έως 64 και τέλος όσοι ήταν άνω των 65 ετών. Οι αλγόριθμοι ταξινόμησης Naive Bayes και Bernoulli εκπαιδεύτηκαν για την παραγωγή των προβλέψεων. Η αξιολόγηση των αλγορίθμων πραγματοποιήθηκε με τη χρήση των μετρικών μεθόδων Precision, Recall και F1-score. Η επίδοση των τριών μετρικών ήταν περίπου στο 0,3 κατά μέσο όρο για κάθε κλάση. Επίσης, θέλησαν να παρουσιάσουν τα αποτελέσματά τους και για τρεις ηλικιακές ομάδες, οι οποίες ήταν άτομα κάτω των 18 ετών, από 18 έως 44 και χρήστες άνω των 45. Σε αυτή την περίπτωση επιτεύχθηκαν καλύτερες τιμές στις μετρικές που για τους χρήστες άνω των 45 ετών έφτασαν το 0,95 και 0,96 σε Recall και Precision αντίστοιχα.

Οι Vasiliki Simaki, Iosif Mporas και Vasileios Megalooikonomou επιχείρησαν με τη μελέτη τους το 2016 [77] να ταυτοποιήσουν ηλικιακά τους χρήστες του Twitter μέσω κοινωνικογλωσσολογικής (sociolinguistic) ανάλυσης των δημοσιεύσεων που έχουν αναρτήσει. Επισήμαναν τις θεωρίες που δηλώνουν ότι υπάρχει στενή συσχέτιση της ηλικίας του χρήστη με τον τρόπο που γράφει, για να ενισχύσουν τη βαρύτητα της έρευνάς τους. Τα δεδομένα που εξόρυξαν από το Twitter αποτελούνταν από 19,377 tweets γραμμένα στην αγγλική γλώσσα. Σκοπός τους ήταν να βρουν την ηλικία του συγγραφέα του tweet μελετώντας το περιεχόμενο του. Η ανάλυση έγινε με βάση τις δημοσιεύσεις οι οποίες χωρίστηκαν σε 6 ηλικιακές τάξεις, όπου κάθε τάξη αντιστοιχεί σε διαφορετικό ηλικιακό εύρος των συγγραφέων. Οι κλάσεις που προέκυψαν ήταν 14 με 19 ετών, 20 έως 24 χρόνων, 25 με 34 ετών, από 35 μέχρι 44 ετών, 45 με 59 χρόνων και τέλος άνω των 60 ετών. Τα χαρακτηριστικά που συνέθεσαν την είσοδο των μοντέλων ήταν 49 συνολικά και πρόκειται για 40 text mining features, δηλαδή στατιστικά για τη χρήση λέξεων και χαρακτήρων ή συμβόλων, 6 sociolinguistic-based features που αφορούσαν το γλωσσικό επίπεδο, και 3 content-based features, σχετικά με τη χρήση ορισμένων εκφράσεων. Η επεξεργασία των κειμένων για την κατασκευή των χαρακτηριστικών έγινε μέσω του πακέτου NLTK. Ο αλγόριθμος ReleifF πραγματοποίησε την αξιολόγηση των features ώστε να επισημανθούν τα πιο σημαντικά. Από αυτά κρίθηκαν σπουδαιότερα το πλήθος των λέξεων που ξεκινούν με κεφαλαίο γράμμα και η εμφάνιση ειδικών χαρακτήρων. Οι αλγόριθμοι ταξινόμησης εφαρμόστηκαν μέσω της μεθόδου 10-fold cross validation, που αναλύεται στην ενότητα 3.5.1.1. Πιο συγκεκριμένα χρησιμοποιήθηκαν ο SVM, τα τυχαία δάση, τα δέντρα απόφασης, ο adaboost, ο Bayes και ένας Bagging. Δοκιμάστηκε επίσης ένα MLP νευρωνικό δίκτυο. Η αξιολόγηση των μοντέλων έγινε με την μετρική accuracy, και τις μεθόδους sensitivity και specificity που είναι το ποσοστό των θετικών που ταυτοποιήθηκαν σωστά και

το ποσοστό των αρνητικών που ταυτοποιήθηκαν σωστά αντίστοιχα. Το πιο αποδοτικό μοντέλο ήταν αυτό του τυχαίου δάσους που παρουσίασε accuracy 61%, sensitivity 60.6% και specificity 84.4%. Ωστόσο, ο αλγόριθμος Bayes πέτυχε το καλύτερο specificity με ποσοστό 86,5%, ενώ γενικά για κάθε μοντέλο που εξετάστηκε η μετρική αυτή σημείωσε υψηλές τιμές.

Συνοπτικά ο Πίνακας 2.1 παρουσιάζει τα επικρατέστερα στοιχεία που χρησιμοποιούνται στις μελέτες για την ανίχνευση της ηλικίας των χρηστών σε κοινωνικά δίκτυα.

Πίνακας 2.1: Συχνότητα χρήσης στοιχείων για την ανίχνευση χαρακτηριστικών ηλικίας σε κοινωνικά δίκτυα

Χαρακτηριστικό	**Συχνότητα χρήσης για προσδιορισμό ηλικίας**	**Social media που εμφανίζονται**
Στοιχεία προφίλ	6	Twitter, Facebook
Αριθμός posts-tweets	5	Twitter, Facebook
Αριθμός followers	6	Twitter
Αριθμός φίλων (followings, friends)	6	Twitter, Facebook
Αριθμός likes	4	Twitter
Φωτογραφία προφίλ	3	Twitter
Βιογραφικό - Description	5	Twitter
Όνομα χρήστη (username)	2	Twitter, Facebook, Google+
Χρώματα προφίλ	1	Twitter
Ενδιαφέροντα χρήστη (interests)	7	Twitter, Facebook, Google+
Τοποθεσία χρήστη (location)	4	Twitter, Facebook
Λεξικογραφικά δεδομένα - sociolinguistics	11	Twitter, Facebook
Πλήθος retweets	5	Twitter
Πλήθος hashtags	5	Twitter
Πλήθος tags-mentions	5	Twitter, Facebook
Μέγεθος κειμένων (tweets, posts)	5	Twitter, Facebook
Λόγος followers-followings	1	Twitter
Πλήθος συνδέσμων (URL, links)	5	Twitter, Facebook
Χρήση emoticons	5	Twitter
Topic modelling[7]	6	Twitter
Ημερομηνία tweets	3	Twitter
Συχνότητα που δημοσιεύει tweets-posts	4	Twitter, Facebook
Sentiment Analysis[7]	5	Twitter, Facebook
Ηλικία account	1	Twitter
Πλήθος σελίδων που είναι μέλος (listed)	1	Twitter

[7] Πρόκειται για δευτερογενή χαρακτηριστικά που προέρχονται από την εκτέλεση καταλλήλων αλγοριθμικών τεχνικών για την δημιουργία τους.

Κεφάλαιο 3

3 Θεωρητικό υπόβαθρο

3.1 Εισαγωγή

Η Μηχανική μάθηση είναι ένα σημαντικό πεδίο της επιστήμης των υπολογιστών και αποτελεί μία εφαρμογή της Τεχνητής Νοημοσύνης, κατά την οποία ένα υπολογιστικό σύστημα μπορεί να εκπαιδεύεται και να βελτιώνεται αυτόματα χωρίς να απαιτείται ο συνεχής προγραμματισμός του [20]. Η Μηχανική Μάθηση αναπτύχθηκε μέσω της μελέτης της αναγνώρισης προτύπων και η κύρια χρήση της γίνεται με σκοπό την ανάπτυξη υπολογιστικών προγραμμάτων και αλγορίθμων, τα οποία έχοντας πρόσβαση σε διάφορα δεδομένα, έχουν τη δυνατότητα να μαθαίνουν και να κάνουν προβλέψεις πάνω σε προγραμματισμός αυτά. Οι αλγόριθμοι αυτοί λειτουργούν εξάγοντας μοντέλα από πειραματικά δεδομένα, ώστε να δώσουν ως αποτέλεσμα κάποιες προβλέψεις βασιζόμενες στα δεδομένα ή να εξάγουν αποφάσεις. Οι αλγόριθμοι της Μηχανικής Μάθησης χωρίζονται σε τρεις κατηγορίες και πρόκειται για τους αλγορίθμους επιβλεπόμενης μάθησης (supervised learning), μη επιβλεπόμενης μάθησης (unsupervised learning) και ημι-επιβλεπόμενης μάθησης (semi-supervised learning) που αναλύονται στη συνέχεια.

Η εκμάθηση των αλγορίθμων βασίζεται σε παρατηρήσεις και δεδομένα, τα οποία αποτελούν τα χαρακτηριστικά (features) εισόδου και πρόκειται για μετρήσιμες ιδιότητες των δεδομένων όπου μπορεί να είναι συνεχείς ή διακριτές τιμές. Αυτά τα χαρακτηριστικά των δεδομένων παρέχουν την γνώση στους αλγορίθμους ώστε να εκτελέσουν με μεγαλύτερη επιτυχία και ακρίβεια την ζητούμενη αναζήτηση μοτίβων ή να λάβουν τις καλύτερες αποφάσεις. Αυτή η διαδικασία στοχεύει στην αυτόματη εκπαίδευση των υπολογιστών και στην προσαρμογή τους ανάλογα τις συνθήκες χωρίς την ανθρώπινη παρέμβαση.

Η μηχανική μάθηση είναι μια μέθοδος που χρησιμοποιείται για την επινόηση πολύπλοκων μοντέλων και αλγορίθμων που οδηγούν στην πρόβλεψη και μπορεί να λειτουργήσει σε μεγάλο όγκο δεδομένων παράγοντας αποτελέσματα με μεγάλη ακρίβεια σε μικρό χρόνο. Ωστόσο είναι πιθανό αρκετές φορές τα μοντέλα να χρειαστούν μεγάλο χρονικό διάστημα ώστε να εκπαιδευτούν σωστά.

Η επιβλεπόμενη μάθηση (supervised learning) αποτελεί κατηγορία της μηχανικής μάθησης που σκοπεύει στον χαρακτηρισμό των δεδομένων και στην εξαγωγή προβλέψεων με βάση κάποια δεδομένα εκπαίδευσης και επισημασμένα (annotated-labeled) παραδείγματα, δηλαδή χρησιμοποιεί όλη την πληροφορία που έχει εξάγει στο παρελθόν στα νέα δεδομένα. Στην προκειμένη περίπτωση, η εκπαίδευση των μοντέλων γίνεται σε δεδομένα εισόδου όπου είναι γνωστή η επιθυμητή έξοδος για κάθε στοιχείο. Οι αλγόριθμοι που χρησιμοποιούνται, αναλύοντας τα δεδομένα εκμάθησης, οδηγούνται στην εξαγωγή μιας συνάρτησης που προβλέπει τιμές για την έξοδο. Μετά την εκπαίδευση το σύστημα μπορεί να παρέχει νέες μεταβλητές στόχου (target variables) για κάθε νέα είσοδο . Τέλος στην επιβλεπόμενη μάθηση είναι εύκολη η σύγκριση της αληθινής τιμής εξόδου με την προβλεπόμενη έξοδο ώστε να γίνονται περαιτέρω τροποποιήσεις στους αλγορίθμους και να οδηγήσουν σε ένα καλύτερο και πιο ακριβές αποτέλεσμα.

Η μη-επιβλεπόμενη μάθηση (unsupervised learning) είναι κατηγορία της μηχανικής μάθησης και εφαρμόζεται σε περιπτώσεις που τα δεδομένα δεν είναι ούτε κατηγοριοποιημένα, ούτε

επισημασμένα (annotated-labeled) [20]. Ο στόχος της είναι η ανακάλυψη μίας πιθανής κρυφής δομής από τα μη χαρακτηρισμένα δεδομένα και σε αντίθεση με την επιβλεπόμενη μάθηση οι αλγόριθμοι εκπαιδεύονται χωρίς την επιθυμητή έξοδο. Όπως γίνεται αντιληπτό ο σκοπός του συστήματος είναι να μελετήσει τα δοθέντα δεδομένα ώστε να προκύψουν συμπεράσματα, ομάδες και αλληλουχίες (clustering) που θα μπορούν να περιγράψουν τα μη κατηγοριοποιημένα δεδομένα, και όχι να δώσει μία σωστή έξοδο. Από τη στιγμή που σε αυτή την εφαρμογή τα παραδείγματα που χρησιμοποιούνται δεν είναι επισημασμένα δεν μπορούν να αξιολογηθούν εύκολα οι πιθανές λύσεις.

Η ημι-επιβλεπόμενη μάθηση (semi-supervised learning) αποτελεί μία ακόμη κατηγορία μηχανικής μάθησης η οποία κινείται μεταξύ της επιβλεπόμενης και της μη επιβλεπόμενης μάθησης. Αυτό συμβαίνει διότι αυτή η προσέγγιση συνδυάζει κατά τη διαδικασία της εκπαίδευσης του μοντέλου ένα μικρό μέρος επισημασμένων δεδομένων με ένα μεγάλο σύνολο μη επισημασμένων. Μέσω της τεχνικής αυτής είναι δυνατό να αυξηθεί σημαντικά η ακρίβεια του παραγόμενου μοντέλου [78].

Στην παρούσα διπλωματική εργασία χρησιμοποιείται η επιβλεπόμενη μάθηση και συγκεκριμένα δοκιμάζονται αλγόριθμοι παλινδρόμησης (regression) και αλγόριθμοι ταξινόμησης (classification). Επίσης χρησιμοποιούνται τεχνικές NLP που ορισμένες εμπίπτουν στην επιβλεπόμενη μάθηση και άλλες στην ημί-επιβλεπόμενη. Όλα τα παραπάνω αναλύονται λεπτομερώς στη συνέχεια αυτού του κεφαλαίου.

3.2 Αλγόριθμοι Παλινδρόμησης

Η Παλινδρόμηση (Regression) είναι μια ιδιαίτερα διαδεδομένη στατιστική τεχνική μοντελοποίησης που εφαρμόζεται για την μελέτη της συσχέτισης μεταξύ μίας εξαρτώμενης μεταβλητής (outcome variable) και μιας ή περισσότερων ανεξάρτητων μεταβλητών (features, predictors) [38]. Τα μοντέλα παλινδρόμησης χρησιμοποιούνται σε προβλήματα όπου η μεταβλητή εξόδου λαμβάνει συνεχόμενες τιμές. Ειδικότερα, η παλινδρόμηση συμβάλλει στην εύρεση της μεταβολής της εξαρτώμενης μεταβλητής όταν οποιαδήποτε ανεξάρτητη μεταβλητή αλλάζει και ταυτόχρονα οι υπόλοιπες ανεξάρτητες μεταβλητές διατηρούνται ίδιες. Ο στόχος της είναι εύρεση της συνάρτησης παλινδρόμησης f (regression function) όπως ονομάζεται, όπου πρόκειται για μια συνάρτηση που αντιστοιχίζει τις ανεξάρτητες μεταβλητές, δηλαδή την είσοδο x σε μία συνεχόμενη μεταβλητή y. Σε περιπτώσεις εφαρμογής της παλινδρόμησης ως τεχνικής εξόρυξης δεδομένων, εξάγεται ένα μοντέλο που χρησιμοποιείται για την πρόβλεψη των συνεχόμενων τιμών εξόδου για νέα δεδομένα. [35], [36]

Στην παρούσα προσέγγιση χρησιμοποιήθηκαν, διάφοροι αλγόριθμοι για το πρόβλημα της παλινδρόμησης. Πιο συγκεκριμένα, εφαρμόστηκε η γραμμική παλινδρόμηση, η παλινδρόμηση Lasso, Ridge, ElasticNet, καθώς και η παλινδρόμηση XGBoost, η παλινδρόμηση με τυχαία δάση, η παλινδρόμηση με διανύσματα υποστήριξης και τέλος η παλινδρόμηση με στοχαστικό φθίνον βαθμωτό διάνυσμα. Οι αλγόριθμοι αυτοί αναλύονται στις ακόλουθες ενότητες [32].

3.2.1 Γραμμική Παλινδρόμηση

Η γραμμική παλινδρόμηση (linear regression) είναι μία προσέγγιση παλινδρόμησης που μοντελοποιεί τη σχέση μεταξύ μίας εξαρτημένης μεταβλητής y και μίας ή περισσοτέρων

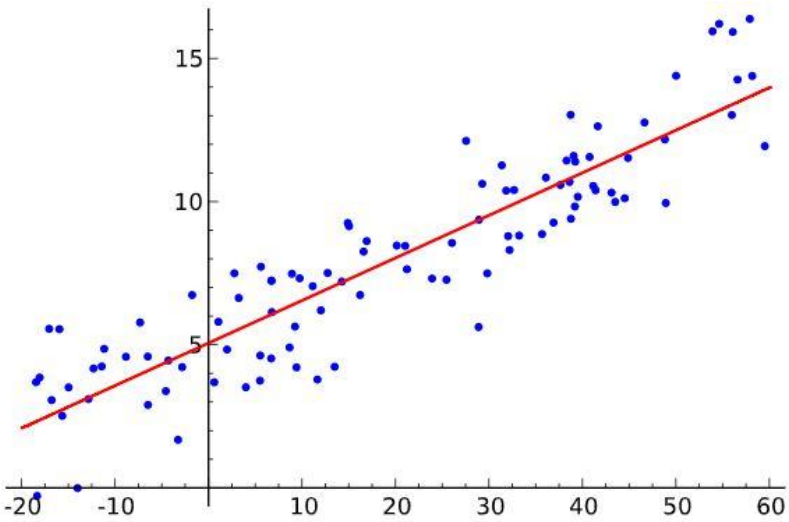
ανεξαρτήτων μεταβλητών x. Για τη χρήση περισσότερων από μίας επεξηγηματικών ή ανεξάρτητων μεταβλητών υπάρχει η πολλαπλή γραμμική παλινδρόμηση (multiple linear regression) αφορά. Στο μοντέλο αυτό τα δεδομένα εισόδου, δηλαδή οι ανεξάρτητες μεταβλητές, μοντελοποιούνται χρησιμοποιώντας γραμμικές λειτουργίες, ενώ μέσω αυτών υπολογίζονται οι επιθυμητές τιμές εξόδου, δηλαδή οι άγνωστες παράμετροι. Τέτοιου είδους μοντέλα καλούνται γραμμικά μοντέλα και συνεπώς η εξαρτημένη μεταβλητή y αποτελεί έναν γραμμικό συνδυασμό των ανεξάρτητων μεταβλητών x [35]. Το μοντέλο της πολλαπλής γραμμικής παλινδρόμησης έχει την παρακάτω μορφή:

$$y = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_i x_i + c$$

με συνάρτηση λάθους (cost function):

$$\sum_{i=1}^{M} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{M} (y_i - \sum_{j=0}^{p} w_j * x_{ij})^2$$

Στην εξίσωση τα x_1, x_2, x_i αποτελούν τις ανεξάρτητες μεταβλητές που εισέρχονται ως στο σύστημα ως δεδομένα εισόδου για την εκπαίδευση του μοντέλου. Όσον αφορά τον υπολογισμό της καλύτερης γραμμής προσαρμογής για τα δεδομένα με βάση το μοντέλο των ελαχίστων τετραγώνων, εφαρμόζονται ελαχιστοποιήσεις για το άθροισμα των τετραγώνων των κάθετων αποκλίσεων από κάθε σημείο δεδομένων με τη γραμμή. Οι αποκλίσεις αυτές δεν εμφανίζουν διαφορές μεταξύ θετικών και αρνητικών τιμών, δηλαδή αν το σημείο είναι πάνω ή κάτω από τη γραμμή διότι είναι τετραγωνισμένες και άρα μπορούν να συγκεντρωθούν συνολικά. Όταν ένα σημείο βρίσκεται ακριβώς πάνω στην προσαρμοσμένη γραμμή τότε έχει κάθετη απόκλιση ίση με το 0. Οι τιμές b_0, b_1, b_i των ελάχιστων τετραγώνων υπολογίζονται συνήθως μέσω κάποιου στατιστικού λογισμικού. Στην Εικόνα 1.1[8] παρουσιάζεται ένα γράφημα μοντέλου γραμμικής παλινδρόμησης.



Εικόνα 3.1: Γραμμική Παλινδρόμηση

Ωστόσο, είναι πιθανό η γραμμική παλινδρόμηση να οδηγήσει σε overfitting σε περιπτώσεις όπου υπάρχει μεγάλος αριθμός features.

[8] Πηγή: https://en.wikipedia.org/wiki/Linear_regression#/media/File:Linear_regression.svg

3.2.2 Παλινδρόμηση Lasso

Η παλινδρόμηση Lasso (least absolute shrinkage and selection operator - Lasso Regression) που ονομάζεται και L1 Regularization αποτελεί μία παραλλαγή του γραμμικού μοντέλου. Παρατηρείται συχνά στη γραμμική παλινδρόμηση το φαινόμενο όπου μία ή περισσότερες ανεξάρτητες μεταβλητές x_i είναι γραμμικά συσχετισμένες μεταξύ τους. Αυτό οδηγεί πολλές φορές σε αυξημένα τυπικά σφάλματα και έτσι περιορίζει την επίδραση κάθε ανεξάρτητης μεταβλητής x_i στην εξαρτημένη μεταβλητή y. Εξαιτίας αυτού γίνεται σχεδόν αδύνατο να εντοπιστούν οι στατιστικά σημαντικές μεταβλητές εισόδου. Σε αυτές τις περιπτώσεις η ανάλυση μέσω παλινδρόμησης γίνεται αφαιρώντας μια ή κάποιες μεταβλητές από το γραμμικά εξαρτημένο σύνολο και πραγματοποιείται συρρίκνωση (shrinkage) του μοντέλου. Η παλινδρόμηση Lasso είναι μία τέτοιου είδους προσέγγιση, κατά την οποία μηδενίζονται οι συντελεστές των χαρακτηριστικών του μοντέλου που δεν είναι εμφανίζουν μεγάλη συσχέτιση. Η συνάρτηση που εκφράζει το λάθος στην παλινδρόμηση Lasso έχει την παρακάτω μορφή:

$$\sum_{i=1}^{M}(y_i - \hat{y_i})^2 = \sum_{i=1}^{M}(y_i - \sum_{j=0}^{p} w_j * x_{ij})^2 + \lambda \sum_{j=0}^{p}|w_j|$$

Στην παραπάνω συνάρτηση θέτοντας το συντελεστή λ=0 έχει την ίδια μορφή με τη γραμμική παλινδρόμηση. Όμως σε αυτήν την περίπτωση η παλινδρόμηση λαμβάνει υπόψη τις απόλυτες τιμές των συντελεστών και εκτελείται με κανονικοποίηση τύπου L1 που μπορεί να οδηγήσει σε μηδενικούς συντελεστές σε κάποια χαρακτηριστικά τα οποία δε θα ληφθούν υπόψη κατά την αξιολόγηση της εξόδου. Με αυτόν τον τρόπο η παλινδρόμηση Lasso βοηθά στην αποφυγή του overfitting[14] καθώς και στην επιλογή των κατάλληλων features [39].

3.2.3 Παλινδρόμηση Ridge

Η παλινδρόμηση Ridge (παλινδρόμηση κορυφογραμμής) που ονομάζεται και L2 Regularization είναι μία ακόμη παραλλαγή του γραμμικού μοντέλου. Σκοπός αυτής της προσέγγισης είναι ο περιορισμός των τυπικών σφαλμάτων που δημιουργούνται από τα χαρακτηριστικά των δεδομένων εισόδου με μικρή συσχέτιση, όπως αναφέρεται και στην παλινδρόμηση Lasso. Αποτελεί μία βασική τεχνική που επιδιώκει τον επαναπροσδιορισμό του μοντέλου και την εκτίμηση διαφορετικών εκτιμητικών μεθόδων εκτός αυτής των ελαχίστων τετραγώνων, όπου προσπαθεί να ελαττώσει την επίδραση των μη σχετικών features στο μοντέλο εκπαίδευσης χωρίς ωστόσο να τα εκμηδενίζει. Η cost function που προσδιορίζει την παλινδρόμηση Ridge προσθέτει μία ποινή (penalty) στους συντελεστές ίση με το τετράγωνο των συντελεστών και δίνεται από τον παρακάτω τύπο:

$$\sum_{i=1}^{M}(y_i - \hat{y_i})^2 = \sum_{i=1}^{M}(y_i - \sum_{j=0}^{p} w_j * x_{ij})^2 + \lambda \sum_{j=0}^{p} {w_j}^2$$

Η παλινδρόμηση Ridge περιορίζει τους συντελεστές w όπως φαίνεται στον ανωτέρω τύπο και με το σφάλμα που ορίζει κανονικοποιεί τους συντελεστές με τέτοιο τρόπο ώστε όταν αυτοί λαμβάνουν μεγάλες τιμές να υπάρχει ανάλογη ποινή και στην συνάρτηση υπολογισμού της εξόδου. Άρα συρρικνώνει τους συντελεστές και βοηθά στην μείωση της πολυπλοκότητας και

της γραμμικότητας του μοντέλου. Ελαττώνοντας όλο και περισσότερο τον συντελεστή λ στην συνάρτηση σφάλματος το μοντέλο της παλινδρόμησης Ridge τείνει να μοιάσει στη γραμμική παλινδρόμηση [39].

3.2.4 Παλινδρόμηση ElasticNet

Επίσης μία μέθοδος είναι η παλινδρόμηση ElasticNet (ElasticNet Regression) η οποία συνδυάζει γραμμικά τις ποινές που επιβάλουν οι μέθοδοι Lasso και Ridge. Η συνάρτηση σφάλματος δίνεται από τον τύπο:

$$\sum_{i=1}^{M}(y_i - \widehat{y_i})^2 = \sum_{i=1}^{M}(y_i - \sum_{j=0}^{p} w_j * x_{ij})^2 + \lambda_1 \sum_{j=0}^{p}|w_j| + \lambda_2 \sum_{j=0}^{p} {w_j}^2$$

$$\text{όπου } \lambda_1 = \alpha \text{ και } \lambda_2 = \frac{1-\alpha}{\alpha} \quad \text{με } 0 < \alpha < 1 .$$

Παρατηρείται ότι στην ανωτέρω εξίσωση περιλαμβάνονται οι μέθοδοι Ridge και Lasso και εξάγεται η καθεμία μηδενίζοντας είτε το λ_1 είτε το λ_2 αντίστοιχα. Άρα για $\alpha = 0$ γίνεται παλινδρόμηση Ridge ενώ για $\alpha = 1$ γίνεται παλινδρόμηση Lasso. Σκοπός της ElasticNet παλινδρόμησης είναι να επιλέξει τη βέλτιστη τιμή για το α ώστε να βελτιστοποιηθεί το παραγόμενο μοντέλο.

3.2.5 Παλινδρόμηση XGBoost

Ο αλγόριθμος XGBoost ή αλλιώς Extreme Gradient Boosting είναι ένας νέος αλγόριθμος που εφαρμόζεται όλο και περισσότερο σε προβλήματα μηχανικής μάθησης λόγω της εξαιρετικής απόδοσης που παρουσιάζει ενώ συνδυάζει δύο τεχνικές, την Boosting και την Gradient Boosting. Πιο συγκεκριμένα η μέθοδος Boosting είναι μία μέθοδος machine learning που αναπτύχθηκε και χρησιμοποιείται τα τελευταία χρόνια και βασίζεται σε ένα σύνολο ατομικών μοντέλων. Ο κύριος σκοπός του Boosting είναι ο συνδυασμός των αποτελεσμάτων πολλών αλγορίθμων μηχανικής μάθησης που σημειώνουν χαμηλές επιδόσεις, δηλαδή αλγορίθμων που έχουν σφάλμα (error rate) ελαφρώς καλύτερο από αυτό της τυχαίας επιλογής. Μέσω αυτής της τεχνικής ελαττώνονται η διακύμανση (variance) και η μεροληψία (bias). Η Gradient Boosting αποτελεί μία μέθοδο που προσθέτει νέα μοντέλα με στόχο τη διόρθωση των σφαλμάτων που παρουσιάστηκαν στα ήδη υπάρχοντα μοντέλα. Η προσθήκη νέων μοντέλων γίνεται διαδοχικά μέχρι το σημείο που δεν μπορούν να γίνουν άλλες βελτιώσεις. Ο XGBoost αλγόριθμος είναι μία παραλλαγή του gradient boosting αλγορίθμου. Στην παλινδρόμηση με τον XGBoost τα αδύναμα μοντέλα είναι δέντρα απόφασης τα οποία προστίθενται διαδοχικά για να προβλέψουν τις διαφορές που εμφανίστηκαν από τις προβλέψεις δέντρων σε προηγούμενο επίπεδο. Αυτά στη συνέχεια συνδυάζονται με τα προηγούμενα δέντρα για να εξάγουν την τελική πρόβλεψη [33], [34]. Η ταχύτητα εκτέλεσης και η κλιμακωσιμότητα του XGBoost, που προκύπτουν από το γεγονός ότι έχει αναπτυχθεί εστιάζοντας στην παραλληλοποίηση της διαδικασίας, είναι σημαντικά χαρακτηριστικά που ωθούν στην χρησιμοποίηση του. Τέλος, δίνει τη δυνατότητα στον προγραμματιστή να εφαρμόσει ευρέως την τεχνική hyperparameter tuning (ενότητα 3.5.1.3) λόγω των πολλών παραμέτρων που περιλαμβάνει όπως για δέντρα, για cross-validation (ενότητα 3.5.1.1) ή για απουσία τιμών.

3.2.6 Παλινδρόμηση με Τυχαία Δάση (Random Forrest)

Το τυχαίο δάσος (Random Forest) είναι μία συλλογή πολλών ατομικών δέντρων αποφάσεων (ενότητα 3.3.4) που λειτουργούν ως ένα σύνολο. Κατά την εκτέλεσή του αλγορίθμου κάθε δέντρο επιλέγει τυχαία τις παρατηρήσεις και τα χαρακτηριστικά που επεξεργάζεται και παράγει μία πρόβλεψη. Στην περίπτωση της παλινδρόμησης με τυχαία δάση ο αλγόριθμος υπολογίζει και επιστρέφει ως έξοδο τον μέσο όρο των αποτελεσμάτων. Το τυχαίο δάσος παρουσιάζει ευρεία χρήση σε προβλήματα μηχανικής μάθησης διότι, αξιοποιώντας ένα σύνολο δέντρων που δεν σχετίζονται μεταξύ τους, καταφέρνει να οδηγήσει σε καλύτερο αποτέλεσμα από εκείνο που παράγουν τα επιμέρους δέντρα που το συνθέτουν όταν αυτά λειτουργούν μεμονωμένα. Σημαντικό ρόλο στην επιτυχία της πρόβλεψης του μοντέλου, παίζει η χαμηλή συσχέτιση των δέντρων απόφασης γιατί αλληλοκαλύπτονται μεταξύ τους ως προς το αποτέλεσμα. Σε αντίθεση με τα δέντρο απόφασης το τυχαίο δάσος αποφεύγει το overfitting[14] αφού κατασκευάζει τυχαία υποσύνολα των δεδομένων εισόδου και μέσω αυτών δημιουργεί νέα μικρότερα δέντρα [40],[42].

3.2.7 Παλινδρόμηση με Διανύσματα Υποστήριξης – SVR

Ένα ακόμα μοντέλο επιβλεπόμενης μηχανικής μάθησης που εφαρμόζεται συχνά σε προβλήματα παλινδρόμησης είναι οι μηχανές διανυσμάτων υποστήριξης (Support Vector Machines – SVM ή Support Vector Networks). Σκοπός του αλγορίθμου SVM είναι να δημιουργήσει ένα μοντέλο που θα κατατάσσει τα δεδομένα σε δύο κατηγορίες. Η εκπαίδευση του γίνεται χρησιμοποιώντας δεδομένα που έχουν ήδη ομαδοποιηθεί και η παραγόμενη έξοδος τοποθετεί νέα δεδομένα σε κάποια από τις δύο κατηγορίες. Συνεπώς ο SVM αλγόριθμος είναι ένας μη δυαδικός γραμμικός ταξινομητής (non-probabilistic binary linear classifier). Ένα μοντέλο SVM αναπαριστά τα δεδομένα ως ένα σύνολο σημείων στο χώρο ώστε να υπάρχει ένα μεγάλο και ξεκάθαρο χάσμα μεταξύ των δεδομένων των δύο κατηγοριών. Στον ίδιο χώρο και με παρόμοιο τρόπο χαρτογραφούνται και νέα δεδομένα τα οποία θα κατηγοριοποιηθούν ανάλογα με την πλευρά του χάσματος που βρίσκονται [48], [49]. Στην παλινδρόμηση χρησιμοποιείται η αντίστοιχη έκδοση του SVM που λέγεται Support Vector Regressor (SVR). Το μοντέλο που παράγει ο SVR εξαρτάται μόνο από ένα υποσύνολο των δεδομένων, επειδή η συνάρτηση σφάλματος που το δημιουργεί αγνοεί τα δεδομένα που βρίσκονται κοντά στην πρόβλεψη. Η χρήση του SVR αλγορίθμου δίνει τη δυνατότητα στον προγραμματιστή να ορίσει την τιμή του αποδεκτού σφάλματος στο μοντέλο και να βρει την βέλτιστη γραμμική απεικόνιση που ταιριάζει στα δεδομένα [46], [47]. Ο χρόνος εκτέλεσης του SVR είναι τετραγωνικά μεγαλύτερος του πλήθους των δειγμάτων που εξετάζονται, γεγονός που τον καθιστά ακατάλληλο να εφαρμοστεί σε αρκετά μεγάλα datasets αφού είναι δύσκολο να κλιμακωθεί. Σε τέτοιες περιπτώσεις συνιστάται η χρήση του αλγορίθμου LinearSVR ή του αλγορίθμου SGD για παλινδρόμηση (SGD Regressor) ο οποίος αναλύεται στην ενότητα 3.2.8 παρακάτω.

3.2.8 Παλινδρόμηση με Στοχαστικό Φθίνον Βαθμωτό Διάνυσμα – SGD

Ο αλγόριθμος SGD (Stochastic Gradient Descent - Στοχαστικό Φθίνον Βαθμωτό Διάνυσμα) είναι μια προέκταση του γραμμικού μοντέλου. Πρόκειται για μία καλύτερη εφαρμογή και βελτιστοποίηση του γραμμικού μοντέλου μέσω της ελαχιστοποίησης του τετραγωνικού σφάλματος. Ο SGD σε κάθε επανάληψη εκτέλεσης του στα δεδομένα πραγματοποιεί εκτιμήσεις για την αντικατάσταση του τρέχοντος βάρους χρησιμοποιώντας την κλίση της

ευθείας πολλαπλασιασμένη με το συντελεστή εκμάθησης (learning rate) και το προηγούμενο βάρος. Στην ουσία αντικαθιστά την πραγματική κλίση με μία εκτίμηση της προσπαθώντας να μειώσει την συνάρτηση σφάλματος όσο περισσότερο γίνεται. Η εξίσωση που εκφράζει το σφάλμα στη μέθοδο φθίνοντος διανύσματος έχει την παρακάτω μορφή:

$$w_{i+1} = w_i - a \sum_{i=1}^{M} \nabla Q_i(w)$$

όπου a είναι το learning rate.

Σε περιπτώσεις όπου το σύνολο των δεδομένων είναι αρκετά μεγάλο συμβάλλει στη μείωση του χρόνου υπολογισμού και εκμεταλλεύεται καλύτερα τους διαθέσιμους υπολογιστικούς πόρους, όμως έχει ως αντάλλαγμα μειωμένη ακρίβεια [41].

3.3 Αλγόριθμοι Ταξινόμησης

Η Ταξινόμηση (Classification) είναι μία ευρέως διαδεδομένη τεχνική που χρησιμοποιείται σε προβλήματα επιβλεπόμενης μηχανικής μάθησης με σκοπό τη δημιουργία ενός μοντέλου που θα μπορεί να πραγματοποιεί προβλέψεις για την αξία μίας εξαρτημένης μεταβλητής αξιοποιώντας ορισμένες ανεξάρτητες μεταβλητές [20]. Τα μοντέλα ταξινόμησης χρησιμοποιούνται σε προβλήματα όπου η έξοδος είναι διακριτή, δηλαδή χωρίζεται σε κατηγορίες όπως για παράδειγμα το φύλο άνδρας ή γυναίκα [50]. Πιο συγκεκριμένα, ένα μοντέλο ταξινόμησης έχει στόχο να προβλέψει την κατηγορία όπου ανήκει κάθε στοιχείο ενός συνόλου δεδομένων. Ένας αλγόριθμος ταξινόμησης λαμβάνει ως είσοδο ένα σύνολο δεδομένων όπου περιέχονται τα σχετικά χαρακτηριστικά (features) και η κατηγορία που ανήκει (target) η κάθε παρατήρηση. Σε περιπτώσεις όπου εφαρμόζεται η ταξινόμηση για την ανάλυση ενός dataset, το μοντέλο που παράγεται μετά την εκπαίδευση μπορεί να προβλέψει τις κατηγορίες που ανήκουν νέα δείγματα δεδομένων.

Στην παρούσα προσέγγιση χρησιμοποιήθηκαν διάφοροι αλγόριθμοι για την επίλυση του προβλήματος με την μέθοδο της ταξινόμησης. Πιο συγκεκριμένα εφαρμόστηκε η ταξινόμηση XGBoost, η λογιστική παλινδρόμηση, η ταξινόμηση με δέντρα απόφασης, η ταξινόμηση με διανύσματα υποστήριξης, η ταξινόμηση με Κ-Πλησιέστερους γείτονες, η ταξινόμηση με τυχαία δάση και τέλος η ταξινόμηση με Bernoulli. Όλοι αυτοί οι αλγόριθμοι αναλύονται στις ακόλουθες ενότητες.

3.3.1 Ταξινόμηση XGBoost

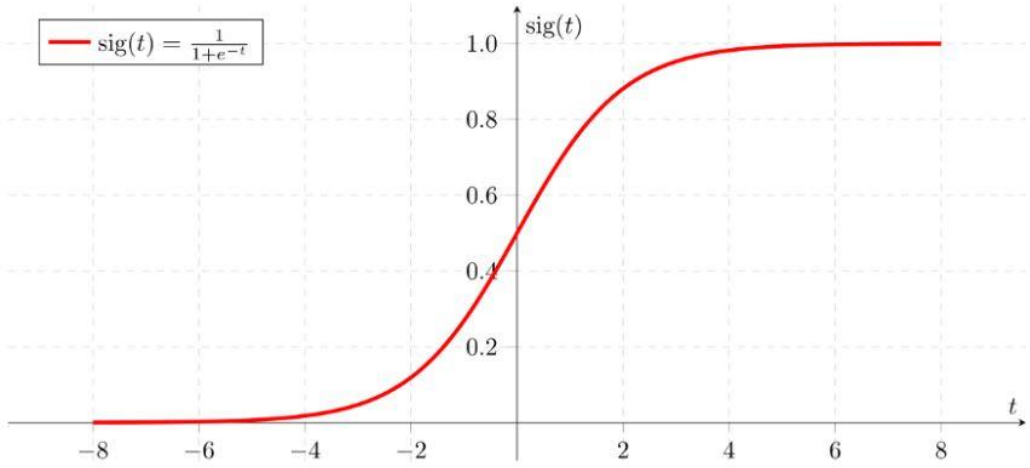
Ο αλγόριθμος XGBoost χρησιμοποιείται τα τελευταία χρόνια όλο και περισσότερο και σε προβλήματα ταξινόμησης. Η μέθοδος αυτή, όπως αναλύθηκε και αναφέρθηκε στην παλινδρόμηση με τον XGBoost στην ενότητα 3.2.5, χρησιμοποιεί τυχαία δέντρα για την πρόβλεψη της εξόδου με ταξινόμηση. Επίσης, παρουσιάζει εξαιρετικές επιδόσεις ακόμα και σε προβλήματα όπου εμφανίζονται πολλές κατηγορίες για την ταξινόμηση των δεδομένων [33], [34].

3.3.2 Λογιστική Παλινδρόμηση (Logistic Regression)

Η λογιστική παλινδρόμηση (Logistic Regression) είναι ένας αλγόριθμος που εφαρμόζεται για την επίλυση προβλημάτων ταξινόμησης παρά το όνομα του. Ο αλγόριθμος αυτός για να πραγματοποιήσει τις προβλέψεις του βασίζεται στη θεωρία των πιθανοτήτων. Μελετά το μη γραμμικό αποτέλεσμα μιας εξαρτημένης μεταβλητής y σχετικά με πολλές ανεξάρτητες μεταβλητές εισόδου x_1, x_2, x_i. Συνήθως εφαρμόζεται σε προβλήματα δυαδικού χαρακτήρα (binary) όπου η έξοδος μπορεί να λάβει δύο διακριτές τιμές. Η λογιστική παλινδρόμηση χρησιμοποιεί τη σιγμοειδή συνάρτηση (Sigmoid function) ως συνάρτηση κόστους για να κάνει τις προβλέψεις της αξιοποιώντας τις πιθανοτικές τιμές των παραμέτρων. Η συνάρτηση κόστους έχει την παρακάτω μορφή:

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

Στην Εικόνα 3.2[9] διαπιστώνεται από τη γραφική παράσταση της συνάρτησης ότι αυτή λαμβάνει τιμές από το 0 ως το 1. Αυτό συμβαίνει γιατί στόχος της λογιστικής παλινδρόμησης είναι να προβλέψει την παρουσία ή την απουσία μίας μεταβλητής μετατρέποντας την πιθανότητα εμφάνισης της σε μία τιμή μεταξύ του 0 και του 1. Το 0 αντιστοιχεί την απουσία του χαρακτηριστικού και το 1 στην ύπαρξη του. Συνεπώς, η συνάρτηση κόστους έχει περιορισμένο σύνολο τιμών γεγονός, που προκρίνει τη λογιστική παλινδρόμηση έναντι των γραμμικών λύσεων [51], [52].



Εικόνα 3.2: Γραφική παράσταση για τη σιγμοειδή συνάρτηση

3.3.3 Ταξινόμηση με Διανύσματα Υποστήριξης – SVC

Οι μηχανές διανυσμάτων υποστήριξης (Support Vector Machines – SVM ή Support Vector Networks), που αναλύονται λεπτομερώς στην ενότητα 3.2.7, παρουσιάζουν ευρεία χρήση και σε προβλήματα ταξινόμησης εκτός από προβλήματα παλινδρόμησης. Στην περίπτωση της ταξινόμησης χρησιμοποιείται η αντίστοιχη έκδοση του SVM που λέγεται Support Vector Classifier (SVC). Το SVC μοντέλο έχει στόχο να χωρίσει τα δεδομένα σε δύο ή περισσότερες

[9] Πηγή: https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc

κατηγορίες ανάλογα με τη φύση του προβλήματος. Όπως και στην περίπτωση του SVR, ο χρόνος εκτέλεσης του SVC κλιμακώνεται τετραγωνικά ανάλογα με το πλήθος των δειγμάτων που εξετάζονται, γεγονός που τον καθιστά ακατάλληλο να εφαρμοστεί σε αρκετά μεγάλα datasets. Σε τέτοιες περιπτώσεις συνιστάται η χρήση του αλγορίθμου LinearSVC ή του αλγορίθμου SGD για ταξινόμηση (SGD Classifier).

3.3.4 Ταξινόμηση με Δέντρα Απόφασης (Decision Tree)

Τα δέντρα απόφασης (Decision Trees) είναι μία αρκετά χρησιμοποιούμενη τεχνική για την πραγματοποίηση προβλέψεων σε πολλούς τομείς μεταξύ αυτών και η μηχανική μάθηση. Εφαρμόζονται ευρέως από τους ερευνητές σε προβλήματα ταξινόμησης. Τα δέντρα απόφασης στοχεύουν να προβλέψουν την τιμή εξόδου μέσα από μία σειρά παρατηρήσεων και συμπερασμάτων. Το μοντέλο δημιουργείται ώστε να προβλέπει το τελικό αποτέλεσμα μέσα από την εκμάθηση απλών κανόνων απόφασης που βασίζονται στα χαρακτηριστικά των δεδομένων εισόδου. Τα δέντρα απόφασης αναπαρίστανται μέσω γράφων και έχουν τη μορφή δυαδικών δέντρων όπου τα φύλλα αντιστοιχούν στις ετικέτες των κλάσεων και τα κλαδιά στα χαρακτηριστικά που οδήγησαν στην επιλογή αυτών των κλάσεων. Η ταξινόμηση των δειγμάτων γίνεται ξεκινώντας από τη ρίζα του δέντρου και εκτείνεται στους κόμβους (nodes) όπου γίνεται ο έλεγχος για κάθε μεταβλητή ξεχωριστά μέχρι να φτάσει σε έναν τελικό κόμβο-φύλλο (node-leaf), δηλαδή την πρόβλεψη. Πρόκειται, για μία αναδρομική διαδικασία που επαναλαμβάνεται για κάθε νέο υποδέντρο που δημιουργείται έχοντας ως ρίζες τους νέους κόμβους. Η ταχύτητα εκτέλεσης τους καθώς και η δυνατότητα να χειριστούν πολλαπλές εξόδους τα καθιστούν ιδιαίτερα δημοφιλή. Επίσης, σημαντικό πλεονέκτημα των δέντρων απόφασης είναι ότι οπτικοποιούνται και έτσι γίνεται εύκολη η ερμηνεία και η κατανόησή τους. Ωστόσο, ένα μειονέκτημα τη χρήσης τους είναι ότι κάποιες φορές κατασκευάζουν πολυσύνθετα δέντρα που δεν μπορούν να γενικευθούν αποδοτικά για τα δεδομένα [53].

3.3.5 Ταξινόμηση με Κ-Πλησιέστερους Γείτονες (KNN)

Ο αλγόριθμος Κ-πλησιέστερων γειτόνων (k-Nearest Neighbors - KNN) είναι μια τεχνική που εφαρμόζεται συχνά σε προβλήματα ταξινόμησης. Πιο συγκεκριμένα, πρόκειται για μία μη παραμετρική μέθοδο που προτείνεται από τον Thomas Cover όπου η είσοδος αποτελείται από τα πιο κοντινά δείγματα εκπαίδευσης στο χώρο χαρακτηριστικών. Η έξοδος είναι η κλάση στην οποία ανήκει το δείγμα. Για κάθε στοιχείο η πρόβλεψη της ομάδας που ανήκει γίνεται μελετώντας τα δεδομένα που βρίσκονται κοντά του. Ο KNN για τον υπολογισμό χρησιμοποιεί τις παρακάτω συναρτήσεις απόστασης:

Ευκλείδεια απόσταση

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

Απόσταση Manhattan

$$\sum_{i=1}^{k}|x_i - y_i|$$

Ο αλγόριθμος KNN ουσιαστικά δεν παράγει ένα μοντέλο προβλέψεων για τα δεδομένα εισόδου αλλά υπολογίζει τις αποστάσεις μεταξύ του στοιχείου και των γειτόνων του [55].

3.3.6 Ταξινόμηση με Τυχαία Δάση (Random Forrest)

Το τυχαίο δάσος (Random Forrest) είναι ένα σύνολο δέντρων απόφασης όπως αναλύθηκε στην ενότητα 3.2.6. Στην περίπτωση όμως της ταξινόμησης τα μεμονωμένα δέντρα που το συνθέτουν λειτουργούν διαφορετικά σε σχέση με αυτά στην παλινδρόμηση. Έτσι εδώ κάθε δέντρο πραγματοποιεί μία πρόβλεψη πάνω στο πρόβλημα ταξινόμησης και τελικά ο αλγόριθμος επιστρέφει ως έξοδο την κλάση με τις περισσότερες εμφανίσεις στο σύνολο των τιμών. Στο στάδιο αυτό που συνδυάζει τα υποδέντρα και εφαρμόζει την τεχνική του bagging (bootstrap aggregating), δηλαδή επιλέγει ως έξοδο τη μεταβλητή με τις περισσότερες ψήφους, επιτυγχάνει αξιοσημείωτη μείωση της διακύμανσης (variance) [40],[42].

3.3.7 Ταξινόμηση με Bernoulli

Η ταξινόμηση Bernoulli είναι μία εκδοχή του naive Bayes ταξινομητή που εφαρμόζεται για μοντέλα Bernoulli πολλαπλών μεταβλητών. Πρόκειται, για έναν αλγόριθμο ταξινόμησης που είναι κατάλληλος για διακριτά δεδομένα και έχει σχεδιαστεί για δυαδικά ή λογικά χαρακτηριστικά τα οποία χρησιμοποιούνται για την πραγματοποίηση των προβλέψεων. Ο αλγόριθμος Bernoulli είναι ένα πολυωνυμικό μοντέλο και εκφράζει στην πιθανότητα όπου το δείγμα εισόδου ανήκει σε μία κλάση [54]. Όντας μία απλή μέθοδος ταξινόμησης τα μοντέλα Bernoulli κάνουν πολλές υποθέσεις για τα δεδομένα και είναι πιθανό να οδηγήσουν σε λάθος εκτιμήσεις. Σε περιπτώσεις όπου τα δείγματα εισόδου είναι λίγα ο αλγόριθμος Bernoulli δίνει ικανοποιητικά αποτελέσματα.

3.4 Επεξεργασία Φυσικής Γλώσσας

Η επεξεργασία φυσικής γλώσσας (ΕΦΓ) ή αλλιώς Natural Language Processing (NLP) είναι ένας σημαντικός και ιδιαίτερα διάσημος κλάδος της επιστήμης της πληροφορικής, της μηχανικής μάθησης και της υπολογιστικής γλωσσολογίας. Ασχολείται με τις αλληλεπιδράσεις μεταξύ των υπολογιστών και των ανθρώπινων φυσικών γλωσσών και σκοπός της είναι να κατανοήσει τη φυσική γλώσσα των ανθρώπων μέσω της επεξεργασίας μεγάλου όγκου δεδομένων. Ουσιαστικά η μέθοδος αυτή επιδιώκει να δημιουργήσει μοντέλα που εξάγουν συμπεράσματα για τα λεξικογραφικά και γλωσσολογικά χαρακτηριστικά των ανθρώπων αναλύοντας τα κείμενα που έχουν γράψει και στην παρούσα εργασία η τεχνική NLP θα βοηθήσει στην μελέτη των tweets των χρηστών. Μέσω αυτής της ανάλυσης αναμένεται να προκύψουν χρήσιμα αποτελέσματα για την προσωπικότητα, τις προτιμήσεις και τα ενδιαφέροντα των χρηστών τα οποία με τη σειρά τους θα συμβάλλουν στην πραγματοποίηση των προβλέψεων για την ακριβή ηλικία τους και την ηλικιακή ομάδα που ανήκουν. Συχνές εφαρμογές της τεχνικής NLP εκτός από την κατανόηση της φυσικής γλώσσας αποτελούν η αναγνώριση ομιλίας και η παραγωγή φυσικής γλώσσας [62].

Η επεξεργασία της φυσικής γλώσσας έκανε την εμφάνισή της από τη δεκαετία του 1950 όπου πρωτοπαρουσιάστηκε από τον Alan Turing ως μία μέθοδος για την αξιολόγηση της ικανότητας ενός συστήματος να επιδεικνύει ευφυή συμπεριφορά ανάλογη με αυτή του ανθρώπου που πλέον καλείται Turing test. Στην τελευταία δεκαετία η διαδικασία εκμάθησης μέσω της

αναπαράστασης των δεδομένων με χαρακτηριστικά (features) και τα βαθιά νευρωνικά δίκτυα είναι δύο μέθοδοι που χρησιμοποιούνται ως επί το πλείστον για την επίλυση προβλημάτων NLP. Αυτό συμβαίνει λόγω των εντυπωσιακών αποτελεσμάτων που παρουσιάζουν σε διεργασίες επεξεργασίας της γλώσσας όπως το language modeling, το topic modelling και το parsing [63], [64].

Η τεχνική του NLP περιλαμβάνει έναν βασικό κύκλο εργασιών πάνω στα δεδομένα και ακολουθεί συγκεκριμένα στάδια επεξεργασίας κατά τη ροή εκτέλεσης της, τα οποία ωστόσο μπορούν να διαφοροποιούνται ανάλογα με την υλοποίηση της κάθε λύσης. Κάθε ένα από τα κείμενα επεξεργάζεται και αναλύεται λέξη προς λέξη για την εξαγωγή των συμπερασμάτων. Ορισμένα από τα στάδια επεξεργασίας είναι τα εξής:

- **Tokenization:** Αποτελεί τη διεργασία που επεξεργάζεται και χωρίζει τις φράσεις σε ξεχωριστές λέξεις τις οποίες επιστρέφει σε μία τούπλα. Για παράδειγμα η φράση “This is a red car” μέσω της μεθόδου tokenization θα γίνει “(This, is, a, red, car)”.
- **Stop-words:** Είναι η διαδικασία κατά την οποία αφαιρούνται από το κείμενο λέξεις χωρίς κάποια ιδιαίτερη σημασία καθώς και λέξεις που χρησιμοποιούνται πάρα πολύ συχνά στο λόγο όπως οι “the”, “a” και “is” της αγγλικής γλώσσας.
- **Lemmatization:** Η τεχνική αυτή μετατρέπει την υπάρχουσα λέξη στην ρίζα από την οποία προέρχεται. Οι λέξεις της αγγλικής γλώσσας “studies” και “studying” θα αντικατασταθούν και οι δύο με τη ρίζα “study”.
- **Stemming:** Είναι παρόμοια τεχνική με την Lemmatization, αλλά σε αυτή την περίπτωση η αντικατάσταση των λέξεων γίνεται αφαιρώντας κάποια κατάληξη ή κάποιο αχρείαστο γράμμα και διατηρώντας την υπάρχουσα ρίζα της λέξης. Άρα σύμφωνα με τη μέθοδο Stemming οι λέξεις της αγγλικής γλώσσας “studies” και “studying” θα αντικατασταθούν με τις λέξεις “studi”, “study” αντίστοιχα.
- **Part-of-speech tagging - POS tagging:** Μέσω αυτής της τεχνικής δίνεται ετικέτα σε κάθε λέξη του κειμένου που δηλώνει τη γραμματική της επισήμανση. Έτσι οι λέξεις μπορούν να ταξινομηθούν ως ουσιαστικά (noun), ρήματα (verb), επιρρήματα (adverb-ADV) ή νούμερα (numeral) καθώς και άλλα. Ωστόσο υπάρχουν και λέξεις που ανήκουν σε αρκετά μέρη του λόγου. Κάποια γενικά παραδείγματα στην αγγλική γλώσσα είναι “is” (verb) , “book” (noun - με την έννοια “βιβλίο”), “book” (verb – με την έννοια “κάνω κράτηση”), “$” (symbol) ή “1” (numeral).
- **Named entity recognition - NER:** Η μέθοδος αυτή είναι υπεύθυνη για την ανάθεση μίας οντότητας του αληθινού κόσμου σε κάθε λέξη του κειμένου ανάλογα με τη σημασία της μέσω ήδη εκπαιδευμένου αλγορίθμου. Ωστόσο παρέχεται η δυνατότητα στον προγραμματιστή να αναπτύξει τις εκάστοτε δικές του οντότητες με τη χρήση κατάλληλων λέξεων-κλειδιών και να τις προσθέσει στις ήδη υπάρχουσες. Για παράδειγμα στην αγγλική γλώσσα η διαδικασία NER δίνει τις ετικέτες “animal” για τη λέξη “cat” ή “city” για την λέξη “London”.
- **Removing punctuation and symbols:** Αποτελεί μία διεργασία κατά την οποία αφαιρούνται τα σημεία στίξης ή άλλοι ειδικοί χαρακτήρες (#, @) από τα κείμενα των δεδομένων εισόδου διότι αποτελούν θόρυβο και δεν βοηθούν στην επεξεργασία της φυσικής γλώσσας.

Στην παρούσα εργασία χρησιμοποιήθηκαν ορισμένες από τις προαναφερθείσες τεχνικές για την εφαρμογή της NLP λύσης. Πιο συγκεκριμένα κατά τη ροή εκτέλεσης εφαρμόστηκαν οι

διεργασίες tokenization, stop-words, lemmatization, removing punctuation and symbols και NER. Στη μέθοδο NER έγινε προσθήκη νέων οντοτήτων στις ήδη υπάρχουσες με σκοπό την απόκτηση πιο στοχευμένης πληροφορίας. Όλες αυτές οι μέθοδοι οδήγησαν στο τελικό σύνολο δεδομένων που αναλύθηκε για τα tweets των χρηστών.

3.4.1 Bag of Words (BoW)

Το μοντέλο Bag of Words (BoW) αποτελεί μία τεχνική που χρησιμοποιείται για text modeling σε προβλήματα NLP. Πρόκειται για μία αρκετά απλή και ευέλικτη μέθοδο που έχει στόχο να αναγνωρίσει τα θέματα που αναφέρονται μέσα σε ένα κείμενο μελετώντας τις λέξεις που το συνθέτουν. Είναι μία ευρέως χρησιμοποιούμενη προσέγγιση που μελετά τη συχνότητα εμφάνισης λέξεων μέσα σε ένα κείμενο για την εξαγωγή features από αυτό που στη συνέχεια θα χρησιμοποιηθούν ως είσοδοι για έναν ταξινομητή που πραγματοποιεί topic modeling. Υποθέτει πως όσο μεγαλύτερη είναι η συχνότητα εμφάνισης μίας λέξης τόσο σπουδαιότερη είναι και η σημασία της για κάποιο κείμενο. Η μέθοδος BoW ξεκινά με τη δημιουργία μίας αριθμημένης λίστας που περιέχει όλες τις μοναδικές λέξεις που εξάγονται από το κείμενο εισόδου και καλείται λεξιλόγιο (vocabulary). Όπως είναι γνωστό οι αλγόριθμοι NLP λειτουργούν με αριθμούς και δεν μπορούν να λάβουν ως είσοδο κείμενο. Για αυτό το λόγο το μοντέλο BoW επεξεργάζεται το κείμενο και το μετατρέπει ώστε να περιγράφει αριθμητικά τη συχνότητα εμφάνισης των λέξεων που περιέχει. Ο τελικός πίνακας περιλαμβάνει τις λέξεις και τις αντίστοιχες συνολικές καταμετρήσεις για την εμφάνιση τους σε όλο το κείμενο εισόδου. Επιπλέον, με αυτόν τον τρόπο κάθε πρόταση γίνεται να αναπαρασταθεί ως ένα διάνυσμα όπου η κάθε λέξη θα λαμβάνει την τιμή 1 όταν υπάρχει στο κείμενο και αντίστοιχα την τιμή 0 όταν δεν υπάρχει. Ωστόσο, η μέθοδος BoW δε λαμβάνει υπόψη τη σειρά και τη σημασιολογία των λέξεων μέσα στις προτάσεις αλλά μόνο την εμφάνιση τους, γεγονός που είναι πιθανό να επηρεάσει σημαντικά τα αποτελέσματα του NLP. Τέλος, απαιτείται ιδιαίτερη προσοχή στο σχεδιασμό του λεξιλογίου διότι αυτό ορίζει το πόσο αραιές είναι οι αναπαραστάσεις του κειμένου εισόδου, κάτι που επιδρά σημαντικά τόσο στον χρόνο εκτέλεσης των υπολογισμών όσο και στην αξία της εξαγόμενης πληροφορίας [65].

3.4.2 Term Frequency-Inverse Document Frequency (TF-IDF)

Η τεχνική Term Frequency-Inverse Document Frequency (συχνότητα όρου – αντίστροφη συχνότητα εγγράφου) ή αλλιώς TF-IDF είναι αρκετά διαδεδομένη και χρησιμοποιείται συχνά σε περιπτώσεις ανάκτησης πληροφορίας και text mining. Πρόκειται, για μία στατιστική μέθοδο που έχει στόχο να αξιολογήσει πόσο σημαντική είναι μία λέξη για ένα κείμενο. Η σπουδαιότητα μίας λέξης αυξάνεται ανάλογα με τον αριθμό των εμφανίσεων της στο κείμενο. Παρατηρείται έντονα πως οι πολύ συχνές λέξεις κυριαρχούν, χωρίς όμως να προσδίδουν σημαντική πληροφορία στο μοντέλο, έναντι άλλων πιο σπάνιων που έχουν μεγαλύτερη σημασιολογική βαρύτητα. Η μέθοδος TF-IDF έρχεται για να αντιμετωπίσει αυτό το πρόβλημα αντισταθμίζοντας τη συχνότητα της λέξης με το πόσο συχνά εμφανίζεται σε όλο το σύνολο δεδομένων προσθέτοντας μία ποινή.

Συνεπώς, ορίζεται ως Term Frequency (TF) η μετρική της συχνότητας της λέξης στη φράση που μελετάται και υπολογίζεται ως εξής:

$$TF = \frac{Number\ of\ times\ term\ t\ appears\ in\ a\ document}{Number\ of\ terms\ in\ the\ document}$$

Επίσης, ορίζεται ως Inverse Document Frequency (IDF) η μετρική που δηλώνει πόσο σπάνια εμφανίζεται η λέξη σε όλο το σύνολο των δεδομένων και υπολογίζεται από τον τύπο:

$$IDF = \log\left(\frac{Number\ of\ total\ documents}{Number\ of\ documents\ a\ term\ t\ has\ appeared}\right)$$

Προφανώς το IDF των σπάνιων λέξεων είναι μεγάλο σε αντίθεση με αυτό των συχνών που λαμβάνει χαμηλή τιμή.

Η τελική τιμή του TF-IDF υπολογίζεται ως το γινόμενο των TF και IDF, δηλαδή:

$$TF - IDF = TF * IDF$$

Το αποτέλεσμα της μέτρησης TF-IDF είναι μία στάθμιση όπου όλες οι λέξεις δεν είναι εξίσου σημαντικές ή ενδιαφέρουσες και έχει στόχο την επισήμανση αυτών που είναι διακριτές και περιέχουν χρήσιμες πληροφορίες για ένα κείμενο εισόδου [66].

3.4.3 Latent Dirichlet Allocation (LDA)

Η τεχνική Latent Dirichlet Allocation[10]- LDA (λανθάνουσα κατανομή Dirichlet) είναι ένα γενετικό στατιστικό μοντέλο που εφαρμόζεται αρκετά συχνά σε προβλήματα επεξεργασίας φυσικής γλώσσας. Πρόκειται για μία μέθοδο που επιτρέπει την εξήγηση ενός συνόλου παρατηρήσεων μέσω μη παρατηρούμενων ομάδων οι οποίες επεξηγούν τους λόγους για τους οποίους κάποια από τα δεδομένα παρουσιάζουν ομοιότητες. Αρχικά προτάθηκε από τους J. K. Pritchard, M. Stephens και P. Donnelly για την γενετική των πληθυσμών το 2000 και από το 2003 με ενέργειες των David Blei, Andrew Ng και Michael I. Jordan εφαρμόζεται σε προβλήματα μηχανικής μάθησης. Τα τελευταία χρόνια χρησιμοποιείται κυρίως για την μοντελοποίηση του θέματος κειμένων (topic modelling) δηλαδή καλείται να αποφανθεί για το θέμα του κειμένου βασιζόμενο στις λέξεις που περιέχει. Το topic modelling είναι ένα πρόβλημα μη επιβλεπόμενης μάθησης για την ταξινόμηση κειμένων. Κατά την εκτέλεση της τεχνικής LDA ορίζεται ο αριθμός k που δηλώνει το πλήθος των topics που πρέπει να χωριστούν τα δεδομένα. Κάθε topic αντιπροσωπεύεται από ένα σύνολο λέξεων. Ο αλγόριθμος LDA υπολογίζει αρχικά σε πιθανοτική μορφή το πλήθος των λέξεων ενός κειμένου εισόδου οι οποίες ανήκουν σε ένα topic και άρα πιθανότατα το κείμενο κατατάσσεται σε αυτό το topic. Επίσης βρίσκει και το πλήθος των κειμένων που έχουν τοποθετηθεί σε μία κατηγορία εξαιτίας κάποιας συγκεκριμένης λέξης που περιέχουν [61].

3.4.4 Guided LDA

Η βιβλιοθήκη GuidedLDA[11] υλοποιεί το μοντέλο Latent Dirichlet allocation (LDA) χρησιμοποιώντας τη γνωστή μέθοδο της στατιστικής collapsed Gibbs sampling[12]. Κατά τον GuidedLDA η διαδικασία εύρεσης και μοντελοποίησης του θέματος του κειμένου (topic modelling) μπορεί να πραγματοποιηθεί καθοδηγούμενη (guided) από κάποιες λέξεις κλειδιά (seed words) που έχουν οριστεί για κάθε topic και σχετίζονται με αυτό. Η δημιουργία αυτού

[10]Πηγή: https://en.wikipedia.org/wiki/Latent_Dirichlet_allocation
[11]Πηγή: https://guidedlda.readthedocs.io/en/latest/
[12]Πηγή: https://en.wikipedia.org/wiki/Gibbs_sampling

του μοντέλου αποσκοπεί στην αντιμετώπιση προβλημάτων που εμφανίζει το απλό LDA μοντέλο στη χρήση του. Αρκετές φορές τα topic που προκύπτουν από την εφαρμογή του LDA δεν είναι κατανοητά ενώ μπορεί να εμφανίζονται και υπερκαλύψεις μεταξύ τους ώστε να μην ικανοποιούν τον ερευνητή. Ο GuidedLDA επιχειρεί να δώσει στις λέξεις μία ώθηση προς την κατεύθυνση, δηλαδή το θέμα, που θέλουμε να συγκλίνουν με βάση την έννοια τους. Προσπαθεί να προσθέσει στο μοντέλο επιπλέον γνώση και πληροφορία μέσω της αρχικοποίησης και της σύνδεσης των topic με λέξεις κλειδιά κι με αυτόν τον τρόπο να αξιοποιήσει την κοινή εννοιολογική σημασία μεταξύ των λέξεων για να πετύχει το κατάλληλο topic modelling. Με άλλα λόγια η τεχνική που εφαρμόζει ο GuidedLDA τείνει να αλλάξει την κατηγοριοποίηση του προβλήματος και την προσέγγιση για τη μελέτη των δεδομένων από unsupervised με τον LDA σε semi-supervised learning.

3.4.5 Μείωση Διαστάσεων

Η μείωση διαστάσεων (dimensionality reduction ή dimension reduction) είναι μία τεχνική που εφαρμόζεται για τον μετασχηματισμό ενός μεγάλου συνόλου δεδομένων σε ένα παρόμοιο αρκετά μικρότερο και υλοποιείται με τη χρήση διάφορων αλγορίθμων. Πιο συγκεκριμένα σε ένα σύνολο δεδομένων οι διαστάσεις αναφέρονται στο πλήθος των features. Σκοπός της μεθόδου είναι να μειώσει το μεγάλο αριθμό των ανεξάρτητων μεταβλητών εισόδου και να δημιουργήσει ένα σύνολο δεδομένων με λίγες διαστάσεις οι οποίες όμως θα αποτελούν ουσιαστική αναπαράσταση των πρωτότυπων δεδομένων. Ιδανικά, η μέθοδος μείωσης διαστάσεων θέλει να εκπληρώσει την γνωστή θεωρία της εγγενούς διάστασης (intrinsic dimension) που εφαρμόζεται στην μηχανική μάθηση και την αναγνώριση προτύπων. Η εγγενής διάσταση για ένα σύνολο δεδομένων μπορεί να θεωρηθεί ως ο αριθμός των μεταβλητών που απαιτούνται σε μια ελάχιστη αναπαράσταση των δεδομένων. Σε πολλά προβλήματα η ύπαρξη πολλών χαρακτηριστικών δεν είναι επιθυμητή για την ανάλυση των δεδομένων καθώς μπορεί να υπάρχει θόρυβος και να οδηγήσει σε λανθασμένα συμπεράσματα. Αυτό συμβαίνει συχνά για παράδειγμα όταν ο πίνακας των δεδομένων εισόδου είναι αραιός (sparse matrix), δηλαδή τα περισσότερα στοιχεία του είναι μηδενικά. Η μείωση διαστάσεων χρησιμοποιείται σε περιπτώσεις όπου υπάρχει μεγάλος αριθμός παρατηρήσεων και μεταβλητών όπως για την αναγνώριση ομιλίας, την ανάλυση σημάτων ή όπως στην δική μας εφαρμογή για την λεξικογραφική και σημασιολογική ανάλυση κειμένου [59].

3.4.6 Αποσύνθεση Μοναδικής Τιμής

Η Αποσύνθεση Μοναδικής Τιμής (Singular Value Decomposition - SVD) είναι μία αρκετά διαδεδομένη μέθοδος για την μείωση διαστάσεων. Τα δεδομένα όπου περιλαμβάνουν μεγάλο αριθμό χαρακτηριστικών ελαττώνονται ως προς αυτά τα χαρακτηριστικά διατηρώντας τα πιο σχετικά για την πραγματοποίηση των προβλέψεων. Όπως προαναφέρθηκε το αποτέλεσμα της μείωσης διαστάσεων είναι ένας νέος πίνακας χαμηλότερου βαθμού. Αυτό επιτυγχάνεται εφαρμόζοντας στα αρχικά δεδομένα τη μέθοδο SVD όπου αναλύει σε ιδιάζουσες τιμές τον πίνακα δεδομένων. Ο τύπος που δίνει την ανάλυση του πίνακα A είναι ο εξής:

$$A = U\,\Sigma\,V^T$$

Όπου U ένας $m \times m$ ορθομοναδιαίος πίνακας, Σ ένας ορθογώνιος πίνακας $m \times n$ με μη αρνητικές τιμές μόνο στη διαγώνιο και V^T ένας ανάστροφος $n \times n$ ορθομοναδιαίος πίνακας.

Ο SVD διαλέγει τις k μεγαλύτερες μοναδικές τιμές του πίνακα Σ για την κατασκευή του νέου πίνακα και μπορεί να εφαρμοστεί σε NLP προβλήματα για πίνακες που δηλώνουν την ύπαρξη και τις εμφανίσεις λέξεων σε ένα κείμενο.

Ο Truncated SVD είναι μία παραλλαγή του SVD κατά την οποία δίνεται έμφαση στα στοιχεία των πινάκων U και V^T που αντιστοιχούν στο στοιχείο του διαγώνιου πίνακα Σ με τη μεγαλύτερη ιδιάζουσα τιμή. Χρησιμοποιεί μόνο τα πρώτα k μεγαλύτερα στοιχεία του Σ θέτοντας τα τυπόλοιπα ίσα με μηδέν και έτσι εκμεταλλεύεται k στήλες των U και V^T, όπου αυτό το k δηλώνει τον αριθμό των διαστάσεων που επιθυμεί να έχει ο προγραμματιστής για τα δεδομένα μετά την εφαρμογή της μεθόδου. Η επιλογή κατάλληλης τιμής για το k είναι ένα σημαντικό ζήτημα και αμφιταλαντεύεται μεταξύ ακρίβειας και χρόνου. Σε περιπτώσεις που το k είναι μεγάλο η προσέγγιση του αρχικού πίνακα είναι σαφώς καλύτερη επιβαρύνοντας ωστόσο το χρόνο υπολογισμού ενώ η επιλογή μικρής τιμής για το k περιορίζει την ακρίβεια αλλά μειώνει σημαντικά το χρόνο εκτέλεσης. Ο Truncated SVD υλοποιεί την μείωση των διαστάσεων χωρίς να συγκεντρώνει τα δεδομένα πριν κάνει τους απαραίτητους υπολογισμούς και συνεπώς λειτουργεί πολύ αποδοτικά για αραιούς πίνακες. Κυρίως χρησιμοποιείται σε πίνακες που μετρούν την εμφάνιση λέξεων ή σε tf-idf πίνακες που προέρχονται από την εφαρμογή vectorizers και για αυτό το λόγο εφαρμόστηκε και στην παρούσα λύση [60].

3.5 Αξιολόγηση Αλγορίθμων

Τα μοντέλα επιβλεπόμενης μάθησης εμφανίζουν αρκετές διαφορές στην απόδοση τους ανάλογα με τον αλγόριθμο που έχει εφαρμοστεί κατά την υλοποίηση τους, τα χαρακτηριστικά (features) των δεδομένων εισόδου, τις επιθυμητές εξόδους καθώς και το είδος του προβλήματος που πρέπει να λύσουν. Αυτό έχει σαν αποτέλεσμα την αξιολόγηση των αλγορίθμων και των μοντέλων με βάση την καταλληλότητά τους, δηλαδή σύμφωνα με το πόσο ικανοποιητικά και ακριβή αποτελέσματα δίνει ο κάθε αλγόριθμος ανάλογα τις συνθήκες, ώστε να βρεθεί η βέλτιστη λύση. Για τον προσδιορισμό της απόδοσης ενός μοντέλου χρησιμοποιούνται διάφορες μέθοδοι και μετρικές. Ορισμένες μετρικές αφορούν το δείκτη απόδοσης που αξιολογεί την ποσοτικοποίηση της επιτυχίας ενός μοντέλου, ενώ άλλες μέθοδοι σχετίζονται με τη διαδικασία με την οποία εξάγονται κάποιες μετρικές. Ωστόσο, όσον αφορά τη μέτρηση της ακρίβειας των αλγορίθμων παλινδρόμησης δεν υπάρχει ένα συγκεκριμένο μέτρο αξιολόγησής τους. Από την άλλη υπάρχουν αρκετές μετρικές για την καταγραφή και αξιολόγηση των σφαλμάτων μεταξύ πραγματικών και προβλεπόμενων τιμών όπως το μέσο απόλυτο σφάλμα και το μέσο τετραγωνικό σφάλμα. Στη συνέχεια παρουσιάζονται οι μετρικές και οι μέθοδοι που χρησιμοποιούνται συχνά για την αξιολόγηση των αλγορίθμων παλινδρόμησης-ταξινόμησης και εξηγούμε τους λόγους για τους οποίους επιλέξαμε την καθεμία σε αυτήν τη μελέτη.

3.5.1 Μέθοδοι Αξιολόγησης

Κατά την εκτέλεση των αλγορίθμων επιβλεπόμενης μάθησης συναντώνται αρκετές και διαφορετικές μέθοδοι σχετικά με τον τρόπο που πραγματοποιείται η εκπαίδευση και η αξιολόγηση του συστήματος. Μία από τις πιο γνωστές μεθόδους είναι το cross-validation κατά την οποία δεσμεύεται ένα μέρος των δεδομένων ώστε να χρησιμοποιηθεί για την αξιολόγηση του μοντέλου. Μία απλή προσέγγιση είναι να χωριστούν τα δεδομένα σε training και test σε ποσοστά 70% - 30% ή 80% - 20% και να γίνει η αξιολόγηση. Υπάρχουν αρκετές τεχνικές

cross-validation και η πιο διαδεδομένη που έχει και αρκετές παραλλαγές αποτελεί η k-fold cross-validation. Επίσης η αξιολόγηση των χαρακτηριστικών των δεδομένων εισόδου (feature importance) κρίνεται εξίσου χρήσιμη για την βελτίωση της αποδοτικότητας του μοντέλου και γίνεται μέσω εκτέλεσης κατάλληλων αλγορίθμων όπως είναι για παράδειγμα ο ταξινομητής των τυχαίων δέντρων (Extremely Randomized Trees Classifier ή αλλιώς Extra Trees Classifier). Τέλος, σημαντικό ρόλο στην επιλογή και την αξιολόγηση του αλγορίθμου παίζει και η επιλογή των κατάλληλων και βέλτιστων παραμέτρων που ορίζονται κατά την εκτέλεση του. Η διαδικασία επιλογής αυτών των παραμέτρων ονομάζεται βελτίωση υπερπαραμέτρων (hyperparameter tuning) και γίνεται μέσω μεθόδων όπως η αναζήτηση πλέγματος (grid search) ή η τυχαία αναζήτηση (randomized search). Όλες οι προαναφερθείσες μέθοδοι αναλύονται σε αυτή την ενότητα.

3.5.1.1 K-fold cross validation

Η διαδικασία του k-fold cross-validation χωρίζει τα δεδομένα σε k υποσύνολα (folds) ίσου μεγέθους. Η μέθοδος αυτή έχει k γύρους και στον καθένα τα k-1 υποσύνολα χρησιμοποιούνται ως training set για τον αλγόριθμο ενώ το υποσύνολο που απομένει χρησιμοποιείται ως test set με βάση το οποίο εξάγονται οι επιθυμητές μετρικές [9], [21]. Οι πιο συνηθισμένες τιμές που χρησιμοποιούνται για το k το 5 ή το 10. Όταν ολοκληρωθούν όλες οι επαναλήψεις της διαδικασίας γίνεται υπολογισμός του μέσου όρου κάθε μετρικής και οι προκύπτοντες αριθμοί αποτελούν τον δείκτη απόδοσης του εκάστοτε αλγορίθμου. Το σφάλμα για τον k-fold validation υπολογίζεται από τον παρακάτω τύπο:

$$cv_{(k)} = \frac{1}{k}\sum_{i=1}^{k} MSE_i$$

Μία παραλλαγή της μεθόδου αυτής είναι η Leave One Out strategy cross validation όπου ισχύει k=n, δηλαδή ο αριθμός των υποσυνόλων ισούται με τον αριθμό των δειγμάτων των δεδομένων εισόδου. Πρόκειται για μία εξαντλητική μέθοδο κατά την οποία σε κάθε επανάληψη χρησιμοποιείται μόνο ένα δείγμα δεδομένων ως test set και όλα τα υπόλοιπα αποτελούν το training set. Στο τέλος της διαδικασίας υπολογίζονται οι μέσοι όροι των μετρικών που εξήχθησαν σε κάθε γύρο όπως και στον κανονικό k-fold.

Η μέθοδος Stratified k-fold cross validation είναι μία έκδοση της κανονικής k-fold μεθόδου στην οποία τα υποσύνολα που δημιουργούνται περιλαμβάνουν περίπου ίδιο ποσοστό δειγμάτων από κάθε ομάδα όπως και στο σύνολο των δεδομένων. Με άλλα λόγια, ρυθμίζει κάθε υποσύνολο να περιέχει τουλάχιστον m δείγματα από κάθε κλάση. Αυτή η προσέγγιση διασφαλίζει ότι μία κλάση δεν εμφανίζεται υπερβολικά πολλές ή λίγες φορές ή ακόμα και καθόλου στις τιμές εξόδου και εφαρμόζεται ιδιαίτερα σε περιπτώσεις που οι τιμές εξόδου των δεδομένων παρουσιάζουν ανισορροπία.

Τα πλεονεκτήματα της μεθόδου k-fold cross-validation είναι αρκετά. Αρχικά, εμφανίζεται μικρή μεροληψία (bias) κατά τη διαδικασία εκπαίδευσης του αλγορίθμου, καθώς όλα τα δεδομένα αποτελούν μέρος του test set ακριβώς μία φορά ενώ χρησιμοποιούνται στο training set k-1 φορές. Επιπρόσθετα, ένα αρκετά σημαντικό πλεονέκτημα αφορά τη διακύμανση (variance), η οποία μειώνεται καθώς το k αυξάνεται και όλα τα δεδομένα έχουν εμφανιστεί στο test set. Αυτό έχει σαν αποτέλεσμα η μέθοδος του k-fold cross-validation μέσω της χρήσης

των διαφορετικών υποσυνόλων να προσφέρει όσο το δυνατόν αντικειμενικά αποτελέσματα για την επίδοση ενός μοντέλου αποφεύγοντας το underfitting[13] και το overfitting[14] του αλγορίθμου στα δεδομένα. Επίσης, βοηθά στο να εξαχθούν ασφαλή συμπεράσματα για το πόσο καλά γενικεύει το μοντέλο και πόσο αποδοτικό μπορεί να είναι αν εφαρμοστεί σε καινούρια δεδομένα. Όσον αφορά, την stratified εκδοχή της μεθόδου παρατηρείται ότι αποδίδει καλύτερα στη μεροληψία και στη διακύμανση σε σύγκριση με το κανονική k-fold cross validation μέθοδο [25].

Στην παρούσα διπλωματική εργασία για την προσέγγιση που επιδιώκει την πρόβλεψη της ηλικίας ομάδας του χρήστη επιλέχθηκε η stratified 5-fold cross-validation μέθοδος μέσω της βελτιστοποίησης των υπερπαραμέτρων, ώστε να βρεθεί το μοντέλο μηχανικής μάθησης που προσφέρει το βέλτιστο αποτέλεσμα. Ο λόγος που επιλέχθηκε αυτή η μέθοδος με k=5 είναι ότι το δείγμα των δεδομένων παρουσιάζει ανισορροπία στις τιμές εξόδου και ταυτόχρονα προσφέρει ένα καλό συνδυασμό κατανάλωσης υπολογιστικών πόρων και αξιόπιστων αποτελεσμάτων.

3.5.1.2 Σπουδαιότητα Χαρακτηριστικών

Η μελέτη της σπουδαιότητας των χαρακτηριστικών (feature importance) και η επιλογή τους (feature selection) παίζουν σημαντικό ρόλο στην αξιολόγηση ενός μοντέλου μηχανικής μάθησης. Αρχικά, βρίσκοντας τη σπουδαιότητα των πεδίων του δείγματος των δεδομένων επιτυγχάνεται μία καλύτερη νόηση της λογικής του μοντέλου που εφαρμόζεται. Το γεγονός αυτό βοηθά στη βελτίωση του μοντέλου καθώς και στην επαλήθευση ότι λειτουργεί αποτελεσματικά. Επιπλέον, με αυτή την επαυξημένη πληροφορία για τα χαρακτηριστικά εισόδου η μελέτη του μοντέλου μπορεί να επικεντρωθεί μόνο σε αυτά που κρίνονται ως τα πιο “σημαντικά” και επιδρούν σε μεγαλύτερο βαθμό στη απόδοση του αλγορίθμου αφαιρώντας τα υπόλοιπα [23]. Η διαδικασία αυτή πολλές φορές βοηθά στην απομάκρυνση του θορύβου που μπορεί να υπάρχει στα δεδομένα και οδηγεί σε παρόμοια ή και καλύτερα αποτελέσματα, ενώ ταυτόχρονα συμβάλλει στη μείωση του χρόνου εκπαίδευση του μοντέλου αφού μειώνονται οι μεταβλητές που λαμβάνονται υπόψη.

Στην παρούσα εργασία χρησιμοποιήθηκε ο αλγόριθμος ταξινόμησης τυχαίων δέντρων (Extremely Randomized Trees Classifier ή Extra Trees Classifier) για τον υπολογισμό της σπουδαιότητας των χαρακτηριστικών. Πρόκειται, για έναν αλγόριθμο που συλλέγει σε ένα δάσος (forest) τα αποτελέσματα πολλαπλών μη συσχετισμένων δέντρων απόφασης (decision trees). Κάθε δέντρο περιέχει έναν αριθμό χαρακτηριστικών από τα οποία διαλέγει το καλύτερο με βάση τη διασπορά τους και το τοποθετεί στο δάσος. Τέλος, παρουσιάζει σε φθίνουσα σειρά από άποψη σπουδαιότητας τα χαρακτηριστικά χρησιμοποιώντας για τον υπολογισμό της τη Mean Decrease in Impurity (MDI) μετρική μέθοδο[27].

[13] Underfitting: Ονομάζεται το φαινόμενο κατά το οποίο ένας αλγόριθμος μηχανικής μάθησης αποτυγχάνει να προσαρμοστεί σε ένα σύνολο δεδομένων με αποτέλεσμα να μην μπορεί να αποδώσει ικανοποιητικά όταν εφαρμοστεί σε νέα δεδομένα (στην περίπτωση ενός ταξινομητή να μπορεί να προβλέψει σωστά την κλάση των καινούριων δεδομένων).

[14] Overfitting: Ονομάζεται το φαινόμενο κατά το οποίο ένας αλγόριθμος μηχανικής μάθησης έχει προσαρμοστεί πολύ καλά σε ένα σύνολο δεδομένων με αποτέλεσμα να αποτυγχάνει να αποδώσει ικανοποιητικά όταν εφαρμοστεί σε καινούρια δεδομένα (στην περίπτωση ενός ταξινομητή να μπορεί να προβλέψει σωστά την κλάση των καινούριων δεδομένων).

3.5.1.3 Βελτίωση Υπερπαραμέτρων

Η βελτίωση των υπερπαραμέτρων (Hyperparameter Tuning-Optimization) είναι μία μέθοδος που έχει στόχο να διαλέξει τις καλύτερες δυνατές υπερπαραμέτρους για την εκτέλεση ενός αλγορίθμου μηχανικής μάθησης. Αυτές οι υπερπαράμετροι είναι στην ουσία παράμετροι οι οποίες ορίζονται ακριβώς πριν ξεκινήσει η διαδικασία της εκμάθησης και δηλώνονται στον κατασκευαστή (constructor) του μοντέλου [24]. Για παράδειγμα, αν χρησιμοποιούμε έναν αλγόριθμο με τυχαία δάση (random forest) οι υπερπαράμετροι αφορούν τις τιμές που μπορεί να πάρει το πλήθος των εκτιμητών (estimators) ή το μέγιστο βάθος (max depth). Αυτή η ανάγκη για την εξεύρεση των βέλτιστων παραμέτρων, οφείλεται στο γεγονός ότι κάθε μοντέλο μπορεί να απαιτεί διαφορετικά βάρη ή περιορισμούς καθώς και άλλο ρυθμό εκμάθησης ώστε να μπορεί να γενικευθεί σε διαφορετικά δεδομένα εισόδου. Οι πιο διαδεδομένες τεχνικές για την πραγματοποίηση του hyperparameter tuning είναι η αναζήτηση πλέγματος (Grid Search) ή η τυχαία αναζήτηση (Randomized Search), ενώ ταυτόχρονα για την επίτευξη της βέλτιστης αξιολόγησης χρησιμοποιείται συχνά η μέθοδος του cross-validation.

Η αναζήτηση πλέγματος (Grid Search) είναι ένας αρκετά συνηθισμένος τρόπος για να υλοποιηθεί η βελτιστοποίηση των υπερπαραμέτρων. Κατά τη μέθοδο αυτή, μελετώνται και αξιολογούνται μία προς μία, μέσω εξαντλητικής αναζήτησης, όλες οι πιθανές υπερπαράμετροι που έχουν οριστεί. Έτσι επιτυγχάνεται η διερεύνηση πολλών μοντέλων όπου το καθένα είναι μοναδικό και έχει ξεχωριστό συνδυασμό υπερπαραμέτρων καλύπτοντας κάθε υποπερίπτωση, οι οποίες τείνουν να θυμίζουν τα στοιχεία ενός πίνακα (grid). Εφαρμόζεται μαζί με μεθόδους cross-validation, όπως για παράδειγμα μέσω του GridSearchCV[15] αλγορίθμου, ώστε να χρησιμοποιηθούν τα δεδομένα εισόδου με τον πιο αποτελεσματικό τρόπο χωριζόμενα σε train και test δεδομένα καθώς και να γίνει η εκπαίδευση του μοντέλου με τον καλύτερο συνδυασμό υπερπαραμέτρων. Μέσω αυτής της υλοποίησης, μπορούν να κατασκευαστούν γρήγορα και εύκολα όλοι οι πιθανοί συνδυασμοί train-test δεδομένων και υπερπαραμέτρων που διαφορετικά θα αποτελούσαν μία αρκετά χρονοβόρα διαδικασία. Το βασικό πλεονέκτημα της μεθόδου grid search είναι ότι εγγυάται τον βέλτιστο συνδυασμό παραμέτρων που δίνονται ως είσοδοι αφού εξετάζει κάθε πιθανό συνδυασμό τους. Ωστόσο, πρόκειται για μία τεχνική που ιδιαίτερα αργή που απαιτεί πολλούς υπολογιστικούς πόρους ώστε να εφαρμοστεί, κάτι που επιβαρύνεται ιδιαίτερα σε περιπτώσεις όπου είναι μεγάλος ο όγκος των δεδομένων εισόδου και το πλήθος των χαρακτηριστικών. Γενικά συνίσταται η χρήση της για εφαρμογές με λίγα δεδομένα.

Η τυχαία αναζήτηση (Randomized Search) είναι εξίσου μία αρκετά διαδεδομένη μέθοδος για να πραγματοποιηθεί το hyperparameter tuning. Διαφέρει σε σχέση με την μέθοδο grid search στον τρόπο που πραγματοποιεί την αναζήτηση για την εύρεση των βέλτιστων συνδυασμών των υπερπαραμέτρων. Όπως στην αναζήτηση πλέγματος έτσι και σ αυτή την περίπτωση ως πρώτο βήμα είναι η δημιουργία ενός πίνακα υπερπαραμέτρων. Ωστόσο, στη συγκεκριμένη τεχνική γίνεται τυχαία αναζήτηση μεταξύ των πιθανών συνδυασμών των υπερπαραμέτρων που έχουν ορισθεί και όχι εξαντλητική προσπέλαση. Επίσης χρησιμοποιείται μαζί με cross-validation μεθόδους παρόμοια με όσα αναφέρθηκαν για την grid search. Στην περίπτωση της τυχαίας αναζήτησης προσφέρεται ο RandomizedSearchCV[16] αλγόριθμος, ο οποίος επιτρέπει στον

[15] Πηγή: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
[16] Πηγή: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html

ερευνητή να διαχειρίζεται καλύτερα το πλήθος των συνδυασμών που θα εκτελεστούν ορίζοντας τον αριθμό των επαναλήψεων. Ταυτόχρονα το πλήθος των επαναλήψεων που θα εκτελεστεί βασίζεται και στους διαθέσιμους υπολογιστικούς πόρους. Ακόμη, αξίζει να τονιστεί πως κρίνεται ιδιαίτερα σημαντική η επιλογή του πλήθους των επαναλήψεων που θα πραγματοποιήσει ο αλγόριθμος, καθώς ένας μικρός αριθμός μειώνει την πιθανότητα εύρεσης της βέλτιστης λύσης, ενώ αντίθετα πολλές επαναλήψεις οδηγούν σε μεγάλη αύξηση του χρόνου εκτέλεσης. Το κυριότερο πλεονέκτημα της randomized search αφορά τον αρκετά μειωμένο χρόνο επεξεργασίας που προσφέρει, ενώ παράλληλα αποδίδει πολύ καλά. Βέβαια δεν εγγυάται πάντα την εύρεση των βέλτιστων υπερπαραμέτρων όπως αναφέρθηκε.

Στην παρούσα διπλωματική εργασία επιλέχθηκε η μέθοδος Randomized Search έναντι της μεθόδου Grid Search, διότι είναι μια αρκετά απλή και πιο εύκολα παραλληλοποιήσιμη διαδικασία. Ταυτόχρονα σημειώνει εξαιρετική απόδοση, παρά το γεγονός ότι εξετάζει αρκετά λιγότερους πιθανούς συνδυασμούς από την αναζήτηση πλέγματος και εκτελείται πολύ πιο σύντομα [28]. Επίσης, αρκετές φορές η τυχαία αναζήτηση αποδίδει καλύτερα από την αναζήτηση πλέγματος εφόσον έχουν δοθεί οι ίδιοι πόροι. Η εφαρμογή της τεχνικής έγινε μέσω του αλγορίθμου RandomizedSearchCV. Για να βρεθεί ο συνδυασμός που πετυχαίνει την υψηλότερη ακρίβεια αφού εκτελεστεί όλη η διαδικασία αναζήτησης, πραγματοποιήθηκε κλήσης της μεθόδου best_estimator_ η οποία παρέχεται από τον εκάστοτε αλγόριθμο αναζήτησης και στην περίπτωση αυτή από τον RandomizedSearchCV.

3.5.2 Μετρικές Αξιολόγησης Παλινδρόμησης

Οι μετρικές χρησιμοποιούνται συνήθως για την αξιολόγηση της απόδοσης ενός αλγορίθμου με σκοπό την εξεύρεση του καλύτερου μοντέλου με τα πιο ικανοποιητικά αποτελέσματα. Στη συνέχεια αυτής της ενότητας παρουσιάζονται οι μετρικές που εφαρμόστηκαν στην παρούσα εργασία για τους αλγορίθμους παλινδρόμησης.

3.5.2.1 Μέσο απόλυτο σφάλμα (MAE)

Στην παλινδρόμηση το μέσο απόλυτο σφάλμα (Mean Absolute Error - MAE) αποτελεί ένα μέτρο για την ακρίβεια της πρόβλεψης σε σχέση με τις πραγματικές τιμές. Είναι το άθροισμα της απόλυτης τιμής των διαφορών μεταξύ πραγματικής και προβλεπόμενης τιμής διά το πλήθος τους. Το MAE εκφράζεται ως πραγματικός αριθμός με τιμές μεγαλύτερες ή ίσες του μηδενός. Άρα διαπιστώνεται πως όσο μικρότερη είναι η τιμή του τόσο καλύτερη είναι και η ακρίβεια της μεθόδου που εφαρμόστηκε. Βέβαια θεωρείται σημαντικό να τονιστεί πως η τιμή του εξαρτάται και από την κλίμακα των τιμών που συγκρίνονται [43]. Υπολογίζεται με τη χρήση του παρακάτω τύπου:

$$MAE = \frac{1}{M}\sum_{i=1}^{M}|y_i - \hat{y}_i|$$

3.5.2.2 Μέσο απόλυτο ποσοστιαίο σφάλμα (MAPE)

Αρκετά συχνά στα προβλήματα παλινδρόμησης κρίνεται σημαντικό να εκτελεστεί ο υπολογισμός των σφαλμάτων πρόβλεψης σε ποσοστιαία μορφή με στόχο την αξιολόγηση της ακρίβειας του μοντέλου. Το μέσο απόλυτο ποσοστιαίο σφάλμα (Mean Absolute Percentage

Error) είναι ο στατιστικός δείκτης που χρησιμοποιείται για αυτό το σκοπό. Πρόκειται, για μία αρκετά χρήσιμη μετρική ιδιαίτερα σε περιπτώσεις όπου οι πραγματικές τιμές και άρα και οι τιμές που πρέπει να προβλεφθούν είναι αρκετά υψηλές. Το μέσο απόλυτο ποσοστιαίο σφάλμα εκφράζεται επί τις εκατό και λαμβάνει τιμές μεγαλύτερες ή ίσες του μηδενός. Συνεπώς, για όλο και μικρότερες τιμές του MAPE ο αλγόριθμος σημειώνει όλο και καλύτερη απόδοση [37]. Το μέσο απόλυτο ποσοστιαίο σφάλμα υπολογίζεται με τη χρήση του παρακάτω τύπου:

$$MAPE = \frac{100\%}{M} \sum_{i=1}^{M} \frac{|y_i - \hat{y}_i|}{y_i}$$

3.5.2.3 Μέσο τετραγωνικό σφάλμα (MSE)

Το μέσο τετραγωνικό σφάλμα (Mean Squared Error - MSE) είναι μία ακόμη μετρική που εφαρμόζεται για την αξιολόγηση της ακρίβειας της πρόβλεψης. Αυτή η μέθοδος δίνει πολύ μεγάλο βάρος στα μεγάλα σφάλματα και μικρότερη βαρύτητα στα μικρά σφάλματα αφού τετραγωνίζει τις τιμές που προκύπτουν από την αφαίρεση πραγματικής και προβλεπόμενης τιμής. Λαμβάνει θετικές πραγματικές τιμές και όσο μικρότερη είναι η τιμή του MSE τόσο καλύτερες είναι οι προβλέψεις του μοντέλου [44]. Υπολογίζεται από τον παρακάτω τύπο ως εξής:

$$MSE = \frac{1}{M} \sum_{i=1}^{M} (y_i - \hat{y}_i)^2$$

3.5.2.4 Ρίζα μέσης τετραγωνικής απόκλισης (RMSE)

Η ρίζα μέσης τετραγωνικής απόκλισης (Root Mean Square Error - RMSE) είναι μία μέθοδος που χρησιμοποιείται συχνά για την μέτρηση των αποκλίσεων μεταξύ των πραγματικών και προβλεπόμενων τιμών στα προβλήματα παλινδρόμησης. Πρόκειται, για την τετραγωνική ρίζα του μέσου όρου των σφαλμάτων, δηλαδή της διαφοράς των τιμών εξόδου με τις πραγματικές. Λαμβάνει πάντα θετικές πραγματικές τιμές και όσο μικρότερη είναι η τιμή της τόσο καλύτερη είναι και η ακρίβεια. Βέβαια πρέπει σε αυτό να ληφθεί υπόψη και η κλίμακα των τιμών των δεδομένων [45]. Υπολογίζεται ως εξής από τον παρακάτω τύπο:

$$RMSE = \sqrt{\frac{1}{M} \sum_{i=1}^{M} (y_i - \hat{y}_i)^2}$$

3.5.2.5 Συντελεστής προσδιορισμού

Ο συντελεστής προσδιορισμού R^2 (coefficient of determination) ή R2 score ή R-squared αποτελεί μία μετρική αξιολόγησης που χρησιμοποιείται σε προβλήματα ταξινόμησης. Ο σκοπός του είναι να εξετάσει κατά πόσο η ευθεία παλινδρόμησης ερμηνεύει τα δεδομένα του δείγματος. Πιο συγκεκριμένα μετρά το ποσοστό της διακύμανσης της εξαρτημένης μεταβλητής y που εξηγείται από ένα σύνολο ανεξάρτητων μεταβλητών X. Πρόκειται για μία τεχνική που δεν έχει μονάδα μέτρησης. Το εύρος τιμών που λαμβάνει είναι μεταξύ του 0, δηλαδή καθόλου

εφαρμογή, και του 1 που σημαίνει τέλεια εφαρμογή αλλά αυτές είναι ακραίες περιπτώσεις και συνήθως ισχύει $0 < R^2 < 1$. Ο συντελεστής R^2 ορίζεται ως το λόγο του αθροίσματος των τετραγώνων εξαιτίας της παλινδρόμησης προς το συνολικό άθροισμα τετραγώνων και παρακάτω φαίνεται ο τύπος που τον εκφράζει:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

όπου

$$SS_{tot} = \sum_i (y_i - \bar{y})^2$$

και

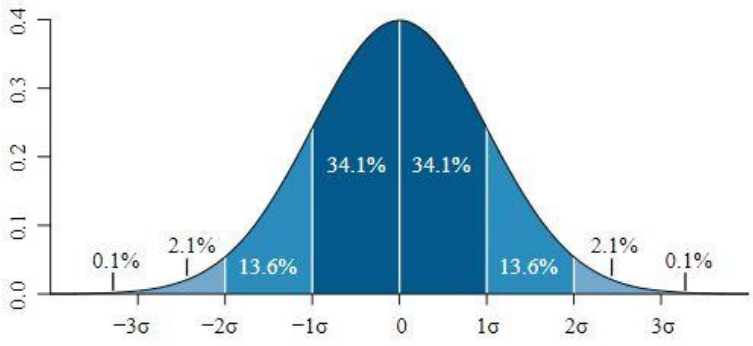
$$SS_{res} = \sum_i (y_i - f(x_i))^2 = \sum_i {e_i}^2$$

3.5.2.6 Τυπική απόκλιση

Η τυπική απόκλιση (Standard Deviation - STD) είναι μία μετρική που χρησιμοποιείται για τον υπολογισμό της μεταβολής ή αλλιώς της διασποράς ενός συνόλου δεδομένων. Υπολογίζεται ως η τετραγωνική ρίζα της διακύμανσης, δηλαδή της απόστασης της τυχαίας μεταβλητής x από τη μέση τιμή $\bar{x}$. Ο τύπος που την εκφράζει είναι ο εξής:

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

Για δεδομένα που παρουσιάζουν χαμηλή τυπική απόκλιση εξάγεται το συμπέρασμα ότι τείνουν κοντά στον μέσο όρο ενώ αντίθετα μία υψηλή τιμή για την τυπική απόκλιση δηλώνει ότι τα στοιχεία λαμβάνουν τιμές σε ένα μεγάλο φάσμα. Η κανονική κατανομή ή αλλιώς κατανομή Gauss περιγράφει την περίπτωση όπου η τυπική απόκλιση είναι μικρή και απεικονίζεται σε γραφική παράσταση με τη μορφή που παρατηρείται στην Εικόνα 3.3[17] παρακάτω[56].



Εικόνα 3.3: Κανονική κατανομή

[17] Πηγή:https://en.wikipedia.org/wiki/Standard_deviation#/media/File:Standard_deviation_diagram.svg

3.5.3 Μετρικές Αξιολόγησης Ταξινόμησης

Η αξιολόγηση της απόδοσης ενός αλγορίθμου γίνεται εφαρμόζοντας διάφορες μετρικές ώστε να βρεθεί το καλύτερο μοντέλο που παρουσιάζει τα πιο ικανοποιητικά αποτελέσματα. Στη συνέχεια αυτής της ενότητας παρουσιάζονται οι μετρικές που εφαρμόστηκαν στην παρούσα εργασία για τους αλγορίθμους ταξινόμησης.

3.5.3.1 Accuracy

Η ακρίβεια ή αλλιώς accuracy αποτελεί μία μετρική που εφαρμόζεται συχνά για την αξιολόγηση μοντέλων ταξινόμησης. Συγκεκριμένα είναι ο λόγος του πλήθους των σωστών προβλέψεων που έκανε το μοντέλο προς το πλήθος όλων των προβλέψεων, όμως στην περίπτωση ταξινομητή ορίζεται αναλυτικότερα από τον τύπο:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Όπου TP (true positive) είναι οι σωστές θετικές προβλέψεις, TN (true negative) οι σωστές αρνητικές προβλέψεις, FP (false positive) οι λανθασμένες θετικές προβλέψεις και FN (false negative) οι λανθασμένες αρνητικές προβλέψεις.

Η μετρική accuracy μετράται ποσοστιαία και μπορεί να δώσει αρκετά αξιόπιστη πληροφορία για την απόδοση ενός ταξινομητή. Ένα καλό μοντέλο πρέπει να σημειώνει όσο το δυνατόν μεγαλύτερο ποσοστό για να είναι αποδοτικό. Ωστόσο η μέθοδος αυτή δεν μπορεί να λειτουργήσει μόνη της και να δώσει ικανοποιητικό και αξιόπιστο συμπέρασμα για περιπτώσεις όπου τα δεδομένα παρουσιάζουν ανομοιομορφία και σημειώνεται μεγάλη διαφορά μεταξύ των θετικών και των αρνητικών ετικετών. Αυτό μπορεί να διαπιστωθεί εύκολα από έναν πίνακα σύγχυσης ή πίνακα σφάλματος (confusion matrix), όπως δείχνει ο Πίνακας 3.1, όπου οπτικοποιεί την απόδοση του αλγορίθμου απεικονίζοντας τα TP, TN, FP, FN και παρουσιάζεται η. Έτσι κρίνεται απαραίτητη η αντιμετώπιση αυτής της ανισορροπίας για την εξαγωγή ασφαλέστερων αξιολογήσεων που επιτυγχάνεται με τη χρήση των μετρικών precision, recall και f1-score που αναλύονται στις ακόλουθες ενότητες.

Πίνακας 3.1: Confusion matrix

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

3.5.3.2 Precision

Μία μετρική που χρησιμοποιείται συχνά για την αξιολόγηση ταξινομητών είναι η μέθοδος precision. Σκοπεύει να δείξει κατά πόσο είναι ακριβής η πρόβλεψη του μοντέλου για τις θετικές τιμές ως προς το πλήθος των τιμών που είναι όντως θετικές. Είναι αρκετά χρήσιμη μετρική ώστε να αξιολογηθεί αποτελεσματικότερα το μοντέλο σε περιπτώσεις όπου τα δεδομένα

παρουσιάζουν ανομοιομορφία και οι κλάσεις δεν είναι ισορροπημένες [58]. Δίνεται από τον παρακάτω τύπο:

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{Total\ Predicted\ Positive}$$

3.5.3.3 Recall

Η μέθοδος recall αποτελεί μία ακόμα μετρική που χρησιμοποιείται συχνά για την αξιολόγηση μοντέλων ταξινόμησης μαζί με τη μέθοδο precision. Σκοπός της είναι να υπολογίζει πόσες από τις τιμές που είναι πραγματικά θετικές εκτιμήθηκαν ως θετικές από το μοντέλο. Όπως και η μετρική precision, έτσι και η μετρική recall θεωρείται αρκετά βοηθητική στην αξιολόγηση της απόδοσης μοντέλων ταξινόμησης όπου τα δεδομένα εισόδου δεν είναι ομοιόμορφα χωρισμένα [58]. Ο τύπος που την υπολογίζει είναι:

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{Total\ Actual\ Positive}$$

3.5.3.4 F1-score

Η μετρική F1-score ή αλλιώς F-measure χρησιμοποιείται συνδυαστικά με τις μετρικές precision και recall για την αξιολόγηση της απόδοσης σε μοντέλα ταξινόμησης. Πρόκειται για μία τεχνική που επιδιώκει να εξισορροπήσει μεταξύ των precision και recall και σε περιπτώσεις που υπάρχει άνιση κατανομή των δεδομένων μεταξύ των κλάσεων και υπάρχει μεγάλο πλήθος πραγματικών αρνητικών τιμών [57]. Αποτελεί συνδυασμό των precision και recall, ενώ μπορεί να ερμηνευθεί και ως ο σταθμισμένος μέσος όρος τους. Δίνεται από τον εξής τύπο:

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Κεφάλαιο 4

4 Τεχνικό υπόβαθρο

Για την πραγματοποίηση της παρούσας διπλωματικής εργασίας χρειάστηκε να γραφτεί κώδικας στην γλώσσα προγραμματισμού Python ώστε να γίνει η επεξεργασία, η ανάλυση των δεδομένων, καθώς και η εφαρμογή των κατάλληλων αλγορίθμων μηχανικής μάθησης. Επίσης, χρειάστηκε να αποθηκευτούν τα δεδομένα σε αρχεία με σκοπό να είναι εύκολα και γρήγορα προσπελάσιμα. Για τις ανάγκες τις διπλωματικής επιλέχθηκε η μορφή CSV για τα αρχεία δεδομένων. Για την ανάπτυξη του κώδικα της εργασίας χρειάστηκε η εγκατάσταση ενός προγραμματιστικού περιβάλλοντος και κάποιον εργαλείων, όπου επιλέχθηκε το JupyterLab το οποίο προσφέρεται μέσω του Anaconda. Τα παραπάνω στοιχεία περιγράφονται αναλυτικότερα στις επόμενες ενότητες.

4.1 Βιβλιοθήκες της γλώσσας προγραμματισμού Python για επεξεργασία δεδομένων

Η Python είναι μία γλώσσα προγραμματισμού ιδιαίτερα διαδεδομένη που δημιουργήθηκε από τον Ολλανδό προγραμματιστή Γκίντο Βαν Ρόσσουμ (Guido Van Rossum) το 1990 με την πρώτη έκδοσή της να έρχεται το 1991. Για τη δημιουργία της νέας γλώσσας ο Van Rossum βασίστηκε στην γνώση του πάνω στην ABC, μία γλώσσα εκπαιδευτικού σκοπού, στο ολλανδικό πανεπιστήμιο CWI (Centrum Wiskunde & Informatica) [29]. Η Python παρουσιάζει σημαντικές διαφορές με άλλες προγραμματιστικές γλώσσες. Ορισμένες και αρκετά αξιοσημείωτες είναι ότι χρησιμοποιεί διερμηνευτή (interpreter) αντί για μεταγλωττιστή (compiler), ενώ ακόμη εστιάζει στην αναγνωσιμότητα του κώδικα. Ο βασικός της στόχος είναι να προσφέρει στον προγραμματιστή τη δυνατότητα να αναπτύξει τις εφαρμογές του σε αρκετά λιγότερες γραμμές κώδικα σε σχέση με άλλες γνωστές γλώσσες προγραμματισμού, όπως η Java ή η C++, κάτι που πετυχαίνει μέσω της ευκολίας στη χρήση και στο συντακτικό της. Οι πιο κοινές εκδόσεις της Python είναι η 2.x και η 3.x, και στην παρούσα εργασία χρησιμοποιήθηκε η 3.7 έκδοση της.

Η Python αποτελεί μία γλώσσα γενικού σκοπού που υποστηρίζει πολλαπλές τεχνικές προγραμματισμού. Πιο συγκεκριμένα υποστηρίζει πλήρως τον συναρτησιακό (functional), τον αντικειμενοστραφή (object-oriented) και τον δομημένο (structured) προγραμματισμό. Αυτό το πλεονέκτημα της την καθιστά κυρίαρχη στην παραγωγή κώδικα, ο οποίος μπορεί να εφαρμοστεί και να προσαρμοστεί εύκολα σε πληθώρα περιπτώσεων και σε έργα μικρής αλλά και μεγάλης κλίμακας. Επίσης, χρησιμοποιεί δυναμικούς τύπους δεδομένων και διαθέτει τεχνική συλλογής σκουπιδιών (garbage collector).

Ένα σπουδαίο στοιχείο της Python που τη διακρίνει μεταξύ άλλων γλωσσών αφορά τον μεγάλο αριθμό βιβλιοθηκών που διαθέτει [10]. Αυτές οι βιβλιοθήκες συμβάλλουν στην απλότητα και στη γρήγορη εκμάθηση της γλώσσας αφού υλοποιούν αρκετά συνηθισμένες και χρήσιμες διεργασίες. Επιπλέον, παρέχουν στους προγραμματιστές χρήσιμα εργαλεία που διευκολύνουν την ανάπτυξη ολοκληρωμένων προγραμμάτων σε αρκετά λιγότερο χρόνο. Στις ενότητες που ακολουθούν αναλύονται οι βιβλιοθήκες που χρειάστηκαν για την υλοποίηση της παρούσας διπλωματικής εργασίας.

4.1.1 SciPy

Η SciPy[18] είναι μια βιβλιοθήκη ελεύθερου λογισμικού (open source) που χρησιμοποιείται σε εφαρμογές στα μαθηματικά και στην επιστήμη των υπολογιστών. Περιέχει μεθόδους για αρκετές σημαντικές εργασίες της επιστήμης και βοηθά σημαντικά σε περιπτώσεις που ο προγραμματιστής θέλει να επεξεργαστεί αριθμούς ή να προβάλλει τα αποτελέσματα της επεξεργασίας τους μέσω ενός υπολογιστή. Περιλαμβάνει τις εξαιρετικά χρήσιμες βιβλιοθήκες NumPy, Pandas, scikit-learn, matplotlib.

4.1.2 NumPy

Η NumPy[19] (Numerical Python) είναι κι αυτή μια βιβλιοθήκη ελεύθερου λογισμικού που δημιουργήθηκε από τον Travis Oliphant το 2005. Σκοπός της είναι να προσφέρει στον προγραμματιστή εύκολη και γρήγορη διαχείριση σε περιπτώσεις που επεξεργάζεται μεγάλους και πολυδιάστατους πίνακες. Επίσης παρέχει αρκετές χρήσιμες μαθηματικές συναρτήσεις υψηλού επιπέδου που αποδίδουν βέλτιστα κατά την εφαρμογή τους στους πίνακες. Ιδιαίτερα σημαντικό να αναφερθεί είναι και το γεγονός πως η βιβλιοθήκη SciPy σχεδιάστηκε ώστε να μπορεί να ενσωματωθεί με πίνακες NumPy.

4.1.3 Pandas

Το Pandas[20] αποτελεί επίσης μια βιβλιοθήκη open-source, η οποία αρχικά είχε αναπτυχθεί στις γλώσσες C και Python το 2008 από τον Wes McKinney. Πρόκειται, για ένα αρκετά σημαντικό εργαλείο που χρησιμοποιείται ευρέως από τους προγραμματιστές για τη διαχείριση και την ανάλυση των δεδομένων κυρίως σε εφαρμογές που έχουν αναπτυχθεί με τη γλώσσα Python. Το βασικό αντικείμενο της βιβλιοθήκης είναι ότι προσφέρει γρήγορες και ευέλικτες δομές δεδομένων με αρκετές εφαρμογές. Η κύρια δομή της όπου αποθηκεύει τα δεδομένα είναι το Data Frame, το οποίο μοιάζει με πίνακα και διευκολύνει σε μεγάλο βαθμό διεργασίες σχετικές με αυτά. Αυτό συμβαίνει διότι διαθέτει έτοιμες συναρτήσεις για την ανάκτηση των δεδομένων από αρκετά είδη πηγών (source), όπως τα αρχεία CSV ή οι βάσεις δεδομένων καθώς και για την αποθήκευσή τους σε αντίστοιχα συστήματα (sink).

4.1.4 Dask

Το Dask[21] αποτελεί ένα ακόμη πακέτο ελεύθερου λογισμικού της Python (διανέμεται με BSD license) που πρωτοεμφανίστηκε το 2018 και είναι μια υλοποίηση των βιβλιοθηκών Pandas και NumPy αφού περιλαμβάνει όλα τα χρήσιμα εργαλεία τους που αναφέρθηκαν στις ενότητες 4.1.3 και 4.1.2 αντίστοιχα, αλλά ταυτόχρονα τις επεκτείνει σε επίπεδο μεγάλου όγκου δεδομένων (big data). Χρησιμοποιείται ώστε να βοηθήσει στην κλιμακωσιμότητα (scalability) των ροών επεξεργασίας της NumPy (NumPy workflows) με τεχνικές παράλληλες επεξεργασίας είτε σε cluster είτε σε ένα μόνο υπολογιστή, ώστε να βοηθήσει στην πολυδιάστατη και γρήγορη ανάλυση των δεδομένων. Τέλος προσφέρει στον προγραμματιστή

[18] Πηγή: https://www.scipy.org/about.html
[19] Πηγή: https://numpy.org/
[20] Πηγή: https://pandas.pydata.org/
[21] Πηγή: https://docs.dask.org/en/latest/

τη δυνατότητα να επεξεργαστεί και να αποθηκεύσει δεδομένα αρκετά μεγαλύτερα από τη μνήμη RAM του υπολογιστικού συστήματος που διαθέτει.

4.1.5 Scikit-learn

Το Scikit-learn[22] αποτελεί μια βιβλιοθήκη ελεύθερου λογισμικού της Python (διανέμεται με BSD license) η οποία παρέχει υλοποιήσεις για αλγορίθμους επιβλεπόμενης και μη επιβλεπόμενης μάθησης καθώς και αρκετά εργαλεία για μετρικές και μεθόδους αξιολόγησης, όπως αναφέρθηκαν στην ενότητα 3, για την αποδοτικότητα αυτών των αλγορίθμων. Έχει βασιστεί στις βιβλιοθήκες NumPy, SciPy, και matplotlib και διαθέτει εύκολα και αποδοτικά εργαλεία για την εξόρυξη, την προεπεξεργασία και την ανάλυση των δεδομένων. Έτσι είναι ένα σημαντικό εργαλείο για την επίλυση προβλημάτων ταξινόμησης (classification), ομαδοποίησης (clustering) και παλινδρόμησης (regression) αλλά και για την επιλογή του βέλτιστου μοντέλου.

4.1.6 Matplotlib

Η Matplotlib[23] είναι επίσης μια open-source βιβλιοθήκη που δημιουργήθηκε το 2003 από τον John Hunter. Πρόκειται για μια βιβλιοθήκη που χρησιμοποιείται για την απεικόνιση μεγάλου όγκου δεδομένων σε γραφήματα, όπως ιστογράμματα (histogram) ή διαγράμματα "πίτες". Σκοπός είναι η εύκολη και γρήγορη σχεδίαση διαγραμμάτων με λίγες γραμμές κώδικα. Λειτουργεί κυρίως με πίνακες NumPy αλλά και με το συνολικό πακέτο της SciPy.

4.1.7 Seaborn

Η Seaborn[24] αποτελεί μια βιβλιοθήκη ανοικτού κώδικα της Python που η υλοποίηση της έχει βασιστεί στην Matplotlib και ενσωματώνει εύκολα τις δομές δεδομένων της Pandas. Παρέχει χρήσιμα εργαλεία για τον όμορφο σχεδιασμό και την κατανοητή απεικόνιση των στατιστικών σε αρκετές μορφές όπως πίνακες (matrix) ή πλέγματα (grid) και γραφικές παραστάσεις (plot).

4.1.8 Beautiful Soup

Η Beautiful Soup[25] είναι μία βιβλιοθήκη της Python που δημιουργήθηκε αρχικά το 2004 από τον Leonard Richardson. Πρόκειται, για ένα πακέτο που πραγματοποιεί προσπέλαση (parsing) σε αρχεία HTML και XML δημιουργώντας ένα δέντρο προσπέλασης (parse tree) με σκοπό την επεξεργασία τους και την εξαγωγή δεδομένων. Είναι αρκετά χρήσιμη για τη μέθοδο του web scraping, δηλαδή την εξόρυξη δεδομένων από ιστοσελίδες στο διαδίκτυο.

4.1.9 Guided LDA

Η βιβλιοθήκη GuidedLDA[26] είναι μία open-source βιβλιοθήκη της Python και υλοποιεί τον αλγόριθμο GuidedLDA που αναλύθηκε στην ενότητα 3.4.4 και χρησιμοποιήθηκε για την εύρεση του θέματος κειμένου (topic definition) των tweets.

---

[22] Πηγή: https://scikit-learn.org/stable/
[23] Πηγή: https://matplotlib.org/
[24] Πηγή: https://seaborn.pydata.org/
[25] Πηγή: https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[26] Πηγή: https://guidedlda.readthedocs.io/en/latest/

4.1.10 SpaCy

Η SpaCy[27] αποτελεί επίσης μία ανοιχτού λογισμικού βιβλιοθήκη της Python που αρχικά δημιουργήθηκε από τον Matthew Honnibal και κυκλοφόρησε για πρώτη φορά τον Φεβρουάριο του 2015. Είναι ένα εξελιγμένο λογισμικό που έχει αναπτυχθεί στην Python και στην Cython για καλύτερη διαχείριση της μνήμης του συστήματος και αποτελεί τον πιο γρήγορο αναλυτή συντακτικού και λέξεων. Χρησιμοποιείται κυρίως για σκοπούς φυσικής επεξεργασίας γλώσσας (NLP) ενώ ακόμα υποστηρίζει ροές εργασιών για deep learning δίνοντας τη δυνατότητα σύνδεσης με αρκετές βιβλιοθήκες όπως η TensorFlow και η PyTorch. Διαθέτει όλα τα απαραίτητα εργαλεία που υλοποιούν τις διεργασίες ενός NLP προβλήματος αφού εσωτερικά της έχει στήσει ένα νευρωνικό δίκτυο που περιλαμβάνει μοντέλα για tokenization, stemming, lemmatization, POS tagging και NER. Συγχρόνως διαθέτει μοντέλα και μπορεί να εφαρμοστεί για πολλές γλώσσες μεταξύ των οποίων είναι τα Αγγλικά, τα Γαλλικά αλλά και τα Ελληνικά, γεγονός που αυξάνει συνεχώς την προτίμηση της έναντι του πακέτου NLTK από τους προγραμματιστές τα τελευταία χρόνια. Τέλος πρόκειται για μία βιβλιοθήκη που αποδίδει πολύ καλά και σε περιπτώσεις εξαγωγής πληροφοριών πάνω σε μεγάλο όγκο δεδομένων και σε εφαρμογές μεγάλης κλίμακας.

4.1.11 NLTK (Natural Language Toolkit)

Το NLTK[28] είναι ένα open-source εργαλείο (toolkit) φυσικής γλώσσας (natural language) όπου η αρχική του έκδοση κυκλοφόρησε το 2001 και δημιουργήθηκε από τους Steven Bird, Edward Loper και Ewan Klein στο τμήμα της επιστήμης υπολογιστών και πληροφορικής του πανεπιστημίου της Πενσυλβάνια. Περιλαμβάνει ένα σύνολο βιβλιοθηκών και προγραμμάτων για την συντακτική και στατιστική ανάλυση NLP. Προσφέρει στον προγραμματιστή εργαλεία για επεξεργασία κειμένου που υλοποιούν τα στάδια εργασιών του NLP όπως tokenization, stemming, tagging, parsing, καθώς και classification (ομαδοποίηση) και σημασιολογική συλλογιστική (semantic reasoning) σε πολλές γλώσσες [30], [31]. Το πακέτο NLTK είναι ένα από τα πιο διαδεδομένα για επεξεργασία φυσικής γλώσσας αφού διαθέτει πάνω από 50 λεξικά είναι εύκολο στη χρήση. Επιπλέον, διαθέτει εργαλεία γραφικής απεικόνισης των δεδομένων αλλά και δείγματα δεδομένων (sample data).

4.1.12 Gensim

Η Gensim[29] αποτελεί επίσης μία NLP βιβλιοθήκη ελεύθερου λογισμικού που δημιουργήθηκε από τον Radim Rehurek με την πρώτη κυκλοφορία της να γίνεται το 2009. Χρησιμοποιείται για τη δημιουργία διανυσμάτων που αναπαριστούν λέξεις και για την αναζήτηση σε συλλογές λέξεων (corpus). Επιπρόσθετα, προσφέρει υλοποιήσεις για αρκετούς αποτελεσματικούς αλγορίθμους NLP που πραγματοποιούν μοντελοποίηση θέματος (topic modeling), ευρετήρια αρχείων (document indexing) και εύρεση ομοιότητας (similarity retrieval) για μεγάλα κείμενα. Ορισμένοι από τους αλγορίθμους που περιλαμβάνει είναι οι Latent Semantic Analysis (LSA/LSI/SVD), Latent Dirichlet Allocation (LDA), Random Projections (RP), Hierarchical Dirichlet Process (HDP) ή ο word2vec. Στην παρούσα εργασία χρησιμοποιήθηκε ο LDA από αυτούς που αναφέρθηκαν.

[27] Πηγή: https://spacy.io/
[28] Πηγή: https://www.nltk.org/
[29] Πηγή: https://pypi.org/project/gensim/

4.1.13 Joblib

Η Joblib[30] είναι ένα πακέτο εργαλείων της Python που διανέμεται με BSD license. Παρέχει στον προγραμματιστή τη δυνατότητα να διαχειριστεί με ταχύτητα μεγάλο όγκο δεδομένων και διαθέτει αρκετές βελτιστοποιήσεις για πίνακες της NumPy. Ο βασικός σκοπός της βιβλιοθήκης είναι να πετύχει με εύκολο τρόπο καλύτερη απόδοση σε περιπτώσεις διεργασιών που απαιτούν πολύ χρόνο. Έτσι εστιάζει στην πραγματοποίηση παράλληλων υπολογισμών καθώς και στην αποφυγή εκτέλεσης χρονοβόρων διαδικασιών που επαναλαμβάνονται συχνά εφαρμόζοντας τεχνικές disk-caching για ταχύτερη ανάγνωση και αποθήκευση των δεδομένων. Επίσης, χρησιμοποιεί συναρτήσεις ώστε να αποθηκεύει Python αντικείμενα (objects) σε αρχεία ώστε να διαχειρίζεται καλύτερα τα δεδομένα αλλά και να διατηρεί την τελευταία λειτουργική κατάσταση του συστήματος σε περιπτώσεις σφάλματος. Στην παρούσα εργασία χρησιμοποιήθηκε η συνάρτηση dump για την αποθήκευση των μοντέλων που διερευνήθηκαν για το topic modelling των tweets καθώς και η αντίστοιχη συνάρτηση load για την φόρτωση τους.

4.1.14 XGBoost

Η XGBoost[31] είναι μια βιβλιοθήκη ελεύθερου λογισμικού που δημιουργήθηκε το Μάρτιο του 2014 και διατίθεται στις γλώσσες Python, Java, C++, R και Julia. Παρέχει την δυνατότητα εφαρμογής του αλγορίθμου μηχανικής μάθησης extreme gradient boosting (XgBoost) με υλοποιήσεις τόσο για προβλήματα ταξινόμησης όσο και παλινδρόμησης, ο οποίος αυξάνει ραγδαία τη δημοτικότητα του στις μέρες μας λόγω των σπουδαίων αποτελεσμάτων και της μεγάλης αποδοτικότητας που παρουσιάζει.

4.2 Εργαλεία και Περιβάλλοντα Ανάπτυξης (Frameworks)

4.2.1 Anaconda

Το Anaconda[32] είναι ένα πακέτο εργαλείων ελεύθερου λογισμικού που διατίθεται για τις γλώσσες Python και R με σκοπό να βοηθήσει την έρευνα και την ανάπτυξη εφαρμογών στους τομείς της μηχανικής μάθησης, του data science και στην επεξεργασία μεγάλου όγκου δεδομένων. Δημιουργήθηκε από τους Peter Wang και Travis Oliphant τον Ιούλιο του 2012 και έκτοτε η ομάδα της εταιρίας Anaconda, που βρίσκεται στο Όστιν των Η.Π.Α., το αναπτύσσει και το διαχειρίζεται. Το πακέτο του Anaconda διατίθεται για όλα τα διάσημα λειτουργικά συστήματα όπως τα Windows, Linux, και macOS ενώ περιλαμβάνει πάνω από 7.500 open source πακέτα και βιβλιοθήκες.

Η εγκατάσταση και η διαχείριση των διαφόρων βιβλιοθηκών και περιβαλλόντων εντός του Anaconda γίνονται μέσω του διαχειριστή πακέτων conda. Ο προγραμματιστής μπορεί να χρησιμοποιήσει το Anaconda είτε μέσω της γραμμής εντολών (cmd στα Windows, terminal στα Linux και στα macOS), είτε μέσω εφαρμογής γραφικού περιβάλλοντος (GUI) μέσω του

[30] Πηγή: https://joblib.readthedocs.io/en/latest/
[31] Πηγή: https://xgboost.readthedocs.io/en/latest/
[32] Πηγή: https://docs.anaconda.com/anaconda/

Anaconda Navigator[33]. Μπορεί να εναλλάσσει τη χρήση τους όμως ανάλογα με την προτίμησή του ή με τι ταιριάζει καλύτερα σε κάθε περίπτωση. Ο Navigator περιλαμβάνει πολλές χρήσιμες εφαρμογές και εργαλεία για την ανάπτυξη προγραμμάτων, με συνεχόμενες ενημερώσεις για κάθε μία, από τις οποίες ο χρήστης μπορεί να εγκαταστήσει όποιες επιθυμεί. Μεταξύ αυτών συναντώνται τα JupyterLab, Jupyter Notebook, Spyder, PyCharm, VSCode, RStudio και άλλα. Στην παρούσα εργασία χρησιμοποιήθηκε το JupyterLab.

4.2.2 Jupyter - JupyterLab

Το Jupyter[34] είναι ένα open-source εργαλείο της Python που δημιουργήθηκε το 2015 από τον Fernando Perez. Χρησιμοποιείται για την εύκολη και γρήγορη ανάπτυξη προγραμμάτων στον τομέα του data science και της επιστήμης των υπολογιστών. Το Jupyter προσφέρει ένα περιβάλλον σε μορφή ιστοσελίδας (web-based environment) στο οποίο ο ερευνητής εργάζεται με notebooks όπου αναπτύσσει και τεστάρει των κώδικά του. Στα notebooks εκτός από κώδικα μπορεί να γίνει απεικόνιση των δεδομένων καθώς και να γίνει συγγραφή κειμένου αφού υποστηρίζονται πολλές μορφές αρχείων όπως τα CSV ή τα απλά αρχεία κειμένου. Τα Jupyter notebooks αποτελούν το πιο συνηθισμένο και διαδεδομένο εργαλείο εργασίας για τους περισσότερους Python data scientists.

Το JupyterLab[35] αποτελεί μία νέα web-based υλοποίηση του Jupyter και κυκλοφόρησε τον Φεβρουάριο του 2018. Το JupyterLab είναι κι αυτό ένα περιβάλλον που υποστηρίζει την ανάπτυξη εφαρμογών σχετικών με το data science και τη μηχανική μάθηση και δίνει πολλές δυνατότητες παραμετροποίησης του user interface. Επίσης, ο χρήστης μπορεί να δουλεύει παράλληλα, ευέλικτα και εύκολα πάνω σε κείμενα και εφαρμογές μέσω των Jupyter notebooks, των επεξεργαστών κειμένου ή των terminals που διαθέτει. Πρόκειται, για ένα κλιμακώσιμο εργαλείο που μπορεί να εκτελεί ταυτόχρονα αρκετά notebooks ανάλογα και με την υπολογιστική δύναμη του μηχανήματος.

Στην παρούσα διπλωματική εργασία προτιμήθηκε η χρήση του JupyterLab, διότι υπήρχε μεγαλύτερη εξοικείωση με το συγκεκριμένο εργαλείο καθώς προσφέρει ένα πιο φιλικό και εύχρηστο γραφικό περιβάλλον (User Interface-UI).

4.3 Μορφοποίηση δεδομένων

4.3.1 CSV

Το CSV[36] (Comma Separated Values) είναι ένα αρχείο κειμένου που χρησιμοποιείται ευρέως από τους προγραμματιστές για την απεικόνιση μεγάλων στηλών δεδομένων οι οποίες χωρίζονται μεταξύ τους με κόμμα. Κάθε γραμμή του αρχείου αποτελεί μία εγγραφή δεδομένων. Είναι αρκετά εύκολο να γίνει η προσπέλαση και η επεξεργασία των δεδομένων που περιέχει με τη χρήση των αντίστοιχων βιβλιοθηκών που προσφέρονται από την Python, όπως τα Pandas που περιγράφονται στην ενότητα 4.1.3 παραπάνω.

[33] Πηγή: https://docs.anaconda.com/anaconda/navigator/
[34] Πηγή: https://jupyter.org/
[35] Πηγή: https://jupyterlab.readthedocs.io/en/stable/index.html
[36] Πηγή: https://en.wikipedia.org/wiki/Comma-separated_values

4.3.2 JSON

Το JSON[37] (JavaScript Object Notation) είναι ένας τύπος αρχείου που χρησιμοποιείται ευρέως τα τελευταία χρόνια για την ανταλλαγή δεδομένων. Είναι ιδιαίτερα εύχρηστο και χρήσιμο για πολλές περιπτώσεις αφού έχουν αναπτυχθεί πολλές μέθοδοι για την ανάγνωση του και την δημιουργία του σε αρκετές γλώσσες προγραμματισμού. Σημαντικό πλεονέκτημά του είναι το γεγονός πως είναι ανεξάρτητο από τη γλώσσα προγραμματισμού που το χρησιμοποιεί. Η μορφή του JSON αρχείου είναι απλή και ευανάγνωστη για τους ανθρώπους.

[37] Πηγή: https://www.json.org/json-en.html

Κεφάλαιο 5

5 Twitter

5.1 Δομή Προφίλ χρηστών Twitter

Στην παρούσα μελέτη το πρώτο βασικό αντικείμενο που χρησιμοποιείται για ανάλυση είναι τα στοιχεία του προφίλ του χρήστη στην πλατφόρμα του Twitter. Η δημιουργία ενός νέου προφίλ στο Twitter είναι αρκετά απλή και απαιτεί από τον χρήστη τη συμπλήρωση μόνο δύο υποχρεωτικών πεδίων, του ονόματος χρήστη (username) που είναι μοναδικό για τον κάθε χρήστη και του ονόματος που αναγράφεται στην οθόνη (display name) τα οποία φαίνονται και δημόσια στο προφίλ του χρήστη. Οι υπόλοιπες πληροφορίες του προφίλ είναι προαιρετικές αλλά εφόσον συμπληρωθούν αναγράφονται κι αυτές δημόσια. Ο χρήστης συμπληρώνοντας και τα προαιρετικά πεδία που παρέχει το Twitter έχει τη δυνατότητα να διαμορφώσει το προφίλ του δίνοντας ορισμένες επιπλέον πληροφορίες για τα προσωπικά του στοιχεία όπως την ημερομηνία γέννησής του και την τοποθεσία του, καθώς και μπορεί να ανεβάσει φωτογραφίες προφίλ (profile photo) και κεφαλίδας (header photo). Επίσης ο χρήστης μπορεί να διαλέξει ένα χρώμα θέματος (theme color) που του αρέσει για να κοσμεί το προφίλ του.

Τα παρακάτω πεδία είναι αυτά που φαίνονται στο προφίλ ενός χρήστη και μπορούν να υποστούν επεξεργασία[38] όποτε ο ίδιος το επιθυμεί:

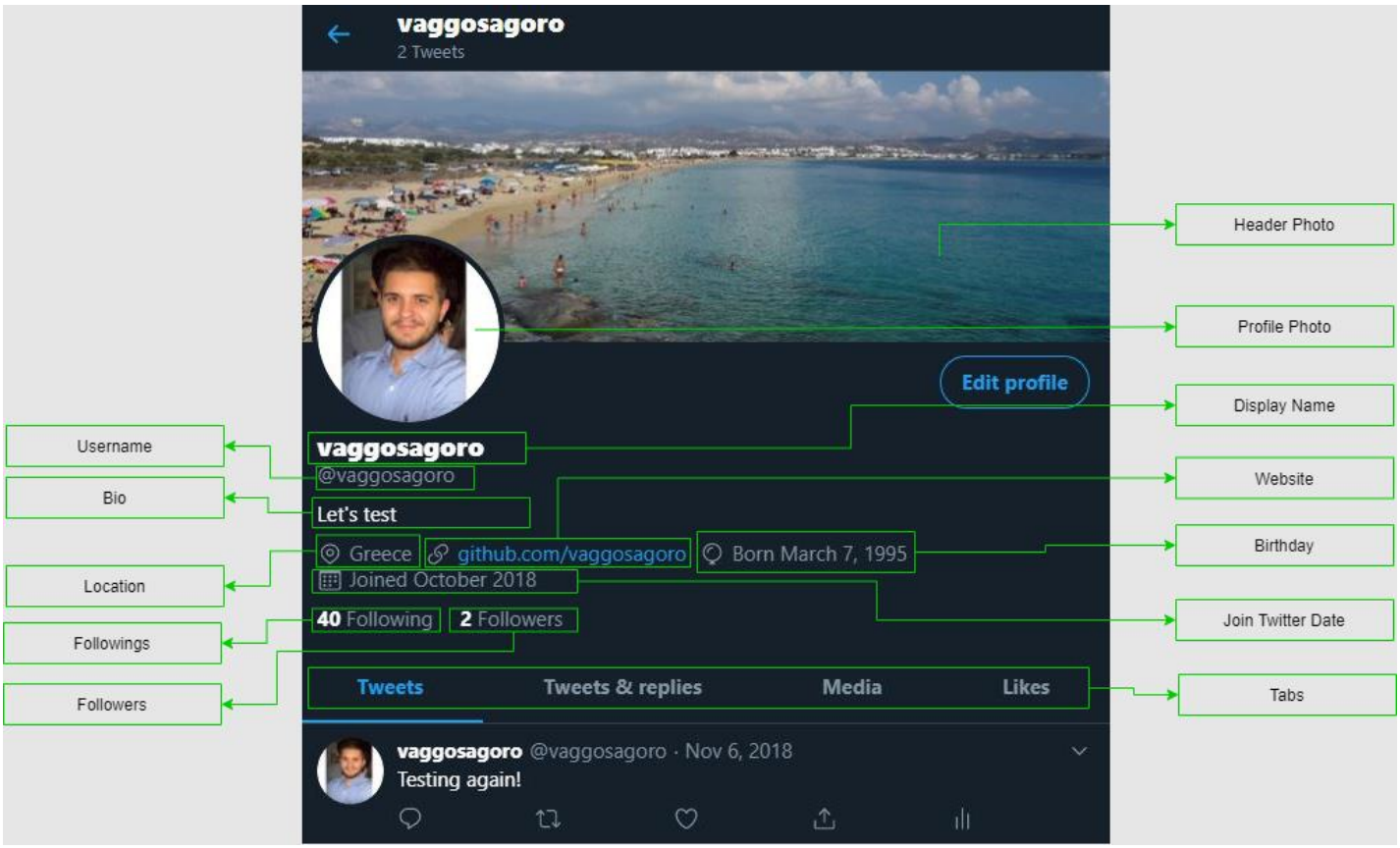
- Φωτογραφία κεφαλίδας - Header photo
- Φωτογραφία προφίλ - Profile photo
- Όνομα στην οθόνη - Display name
- Όνομα χρήστη – Username
- Βιογραφικό - Bio (μέγιστο επιτρεπόμενο όριο 160 χαρακτήρες)
- Τοποθεσία - Location
- Ιστοσελίδα - Website
- Γενέθλια - Birthday

Στην Εικόνα 5.1 παρουσιάζεται το προφίλ ενός χρήστη όπως φαίνεται στην ιστοσελίδα του Twitter. Αρχικά, βλέπουμε σαν κεφαλίδα το όνομα χρήστη (username) και των αριθμό των tweets που έχει δημοσιεύσει. Στη συνέχεια, φαίνονται τα υποχρεωτικά και προαιρετικά πεδία που περιγράψαμε παραπάνω. Ταυτόχρονα, παρατηρούμε ορισμένα επιπλέον στοιχεία του προφίλ του χρήστη και πρόκειται για την ημερομηνία δημιουργίας του προφίλ του στην πλατφόρμα του Twitter, για τον αριθμό των ατόμων που ακολουθεί (Following) και τον αριθμό των ατόμων που τον ακολουθούν (Followers).

Στο κάτω μέρος της εικόνας εμφανίζονται τέσσερις καρτέλες που παρέχουν και εμφανίζουν στην οθόνη ξεχωριστές πληροφορίες ανάλογα με την επιλογή τους. Στην πρώτη καρτέλα (Tweets) βρίσκονται όλα τα tweets που έχει αναρτήσει ο χρήστης. Στην επόμενη καρτέλα (Tweets & replies) υπάρχουν τα tweets του χρήστη μαζί με τις αντίστοιχες απαντήσεις από

[38] Πηγή: https://help.twitter.com/en/managing-your-account/how-to-customize-your-profile

άλλους χρήστες που αναφέρονται σε αυτά. Η τρίτη καρτέλα (Media) περιέχει τις φωτογραφίες και τα βίντεο που έχει ανεβάσει ο χρήστης, ενώ στην τελευταία καρτέλα (Likes) συναντάμε τις δημοσιεύσεις στις οποίες ο χρήστης έχει κάνει like. Τέλος, στην Εικόνα 5.2 φαίνονται οι επιλογές που δίνονται στον χρήστη για το χρώμα θέματος του προφίλ του.



Εικόνα 5.1: Δομή του προφίλ ενός χρήστη στο Twitter

Εικόνα 5.2: Επιλογή χρώματος θέματος στο Twitter

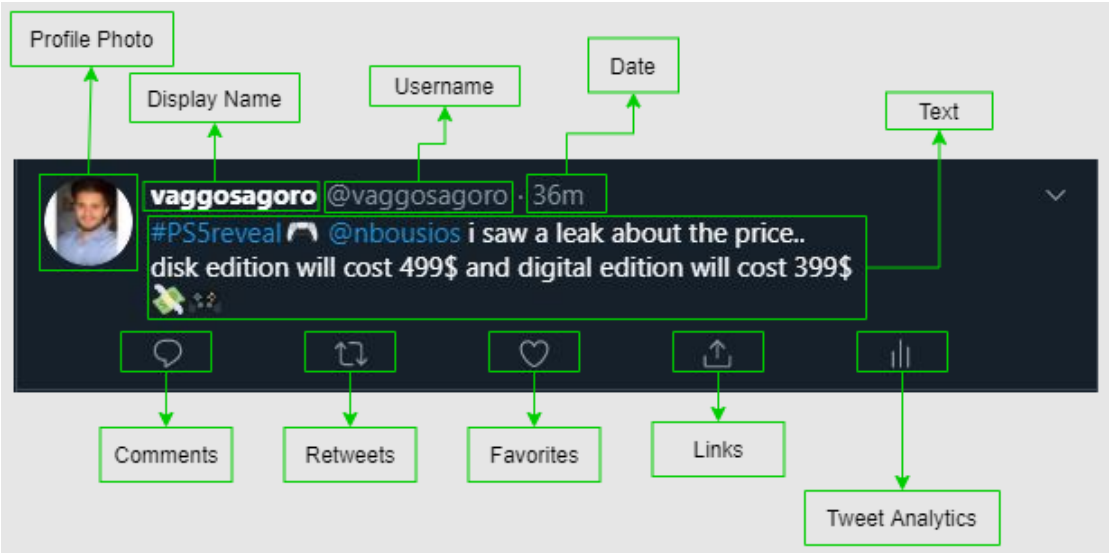
Κρίνεται σημαντικό να επισημανθεί σε αυτό το σημείο ότι χρησιμοποιώντας το Twitter API μέσω του εργαλείου Tweepy, το οποίο περιγράφεται στην Ενότητα 5.3, διαπιστώθηκε πως στα εξαγόμενα δεδομένα για κάθε χρήστη υπάρχουν διαφορές στην ονομασία ορισμένων πεδίων του προφίλ που προαναφέραμε. Έτσι το πεδίο που χαρακτηρίζει το Twitter ως display name αναφέρεται ως "username", ενώ το πεδίο που το Twitter χαρακτηρίζει ως username, αναφέρεται ως "screen name". Για την αποφυγή παρανοήσεων, στην παρούσα εργασία συμπεριλαμβανομένης και της αναφοράς σε παλαιότερες μελέτες, τα πεδία θα αναφέρονται ως display name και username ώστε να υπάρχει συνάφεια με την επίσημη ονομασία που δίνει το Twitter καθώς και τις αναφορές και την Εικόνα 5.1 που αναλύθηκαν παραπάνω.

5.2 Δομή Tweet

Στην παρούσα εργασία το δεύτερο βασικό αντικείμενο που χρησιμοποιείται για την εξαγωγή συμπερασμάτων είναι τα tweets που δημοσιεύει ένας χρήστης στην πλατφόρμα του Twitter. Τα στοιχεία που απαρτίζουν καθένα από αυτά τα tweets όπως εμφανίζονται στην οθόνη είναι τα εξής:

- Μικρογραφία της εικόνας χρήστη
- Όνομα οθόνης - display name
- Όνομα χρήστη - username
- Ημερομηνία ανάρτησης
- Το περιεχόμενο του tweet (text) ως κείμενο, σύνδεσμος ή/και εικόνα με μέγιστο αριθμό 280 χαρακτήρων
- Επισήμανση "Σχόλιο" (comments)
- Επισήμανση "Αναδημοσιεύσεων" (retweets) από άλλους χρήστες
- Επισήμανση "Μου αρέσει" (favorites)
- Επισήμανση "Σύνδεσμοι" (links)
- Επισήμανση "Tweet Analytics"
- Επισήμανση "Retweeted" σε περίπτωση που ο χρήστης αναδημοσίευσε κάποιο άλλο tweet.

Στην παρουσιάζεται ένα tweet με τα στοιχεία που αναφέρθηκαν.



Εικόνα 5.3: Δομή του Tweet

Χρησιμοποιώντας το Twitter API μέσω του εργαλείου Tweepy, το οποίο περιγράφεται στην Ενότητα 5.3, διαπιστώθηκε ότι καθώς γίνεται η εξαγωγή των δεδομένων από τα tweets περιλαμβάνονται τα στοιχεία που αναφέρθηκαν προηγουμένως αλλά και αρκετές επιπλέον πληροφορίες. Από αυτές οι πιο σημαντικές είναι τα πεδία του προφίλ του χρήστη, το permalink όπου βρίσκεται και είναι διαθέσιμο το tweet και η γλώσσα που είναι γραμμένο το tweet. Τα χαρακτηριστικά του περιεχομένου ενός του tweet, όπως το θέμα του κειμένου, οι αναφορές (mentions) σε άλλους χρήστες, τα hashtags για κατηγοριοποίηση του tweet ή η ύπαρξη emoji σε συνδυασμό με τα ανωτέρω στοιχεία έχουν ιδιαίτερη σημασία για τις αναλύσεις της παρούσας έρευνας.
Τέλος κρίνεται σημαντικό να γίνει μία αναφορά σε μία αλλαγή ορόσημο που έγινε τα τελευταία χρόνια στα tweets των χρηστών του Twitter και αφορά το μέγιστο μέγεθος τους, το οποίο από 140 χαρακτήρες στο παρελθόν πλέον ανέρχεται σε 280. Είναι αρκετά προφανές να συμπεράνει

κανείς ότι αυτή η αλλαγή επηρεάζει άμεσα τις μελέτες για την ανίχνευση της ηλικίας των χρηστών του Twitter που βασίζονται στην ανάλυση των tweets που αναρτούν, αφού πλέον οι ερευνητές θα έχουν περισσότερη πληροφορία για τα λεξικογραφικά χαρακτηριστικά τους.

5.3 Twitter API – Tweepy

Το Twitter, ακολουθώντας πλέον τον ίδιο δρόμο με τα περισσότερα μέσα κοινωνικής δικτύωσης, προσφέρει μια διεπαφή (API) μέσω της οποίας κάθε ενδιαφερόμενος προγραμματιστής ή ερευνητής μπορεί έχει πρόσβαση και να αξιοποιήσει μέρος των υπηρεσιών της πλατφόρμας καθώς και τα δεδομένα των χρηστών που είναι δημόσια και διαθέσιμα για ανάκτηση και χρήση. Από τα διαθέσιμα δεδομένα τα πιο αξιοσημείωτα είναι οι πληροφορίες του προφίλ και τα tweets του χρήστη όπως περιγράφηκαν στην Ενότητα 5.1 και στην Ενότητα 5.2 αντίστοιχα. Για τους σκοπούς της παρούσας έρευνας η επικοινωνία με το Twitter API[39] πραγματοποιήθηκε χρησιμοποιώντας το Tweepy. Το Tweepy[40] είναι μια open-source βιβλιοθήκη της Python που λειτουργεί σαν wrapper για το Twitter API και δίνει στον προγραμματιστή εύκολη πρόσβαση στις υπηρεσίες του. Στη συνέχεια περιγράφονται τα βήματα που ακολουθήσαμε για την πρόσβαση στα δεδομένα των χρηστών του Twitter.

Αρχικά, το Twitter API προϋποθέτει OAuth Authentication[41], ώστε να μπορεί κάποιος να πραγματοποιήσει κάποιο request στην πλατφόρμα. Έτσι αποκτώντας λογαριασμό πιστοποιημένου προγραμματιστή δημιουργήσαμε ένα Twitter application[42] και μας δόθηκε πρόσβαση σε ένα API Key και ένα API Secret καθώς και σε ένα access token. Αυτά χρησιμοποιήθηκαν για τη δημιουργία ενός authentication handler με χρήση του Tweepy[43]. Το επόμενο βήμα ήταν να δημιουργηθεί ένα instance του Twitter API μέσω της μεθόδου tweepy.api έχοντας ως παραμέτρους τον authentication handler που αναφέραμε και την wait_on_rate_limit = True όπως φαίνεται στο παρακάτω τμήμα κώδικα.

```
import tweepy
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_key, access_secret)
api = tweepy.API(auth, wait_on_rate_limit=True)
```

Η τελευταία παράμετρος σχετίζεται με το rate-limit που επιβάλλει το Twitter API για το ύψος των αιτήσεων (requests) που μπορούν να πραγματοποιηθούν. Πιο συγκεκριμένα, τα rate limits διαιρούνται ανά χρήστη ή ανά εφαρμογή σε διαστήματα 15 λεπτών, όπου ανάλογα το είδος του request ο αριθμός μπορεί να διαφέρει σε 15, 180, 900 ή 1500 GET requests ανά διάστημα[44]. Σε περίπτωση που ο αριθμός των requests υπερβεί το rate limit, το API επιστρέφει ως απάντηση το response code HTTP 429 “Too Many Requests” και απαιτείται αναμονή πριν την αποστολή νέου request.
Με τη δημιουργία του instance του Twitter API και μέσω του Tweepy καλέσαμε τη μέθοδο api.get_user περνώντας ως παράμετρο το user ID με σκοπό τη λήψη του screen name κάθε χρήστη του dataset και την ταυτοποίηση τους καθώς όσοι χρήστες είχαν μη προσβάσιμα προφίλ αφαιρέθηκαν από το dataset της έρευνας. Αποκτώντας αυτή την επιπλέον πληροφορία και

[39] Πηγή: https://developer.twitter.com/en/docs
[40] Πηγή: https://github.com/tweepy/tweepy
[41] Πηγή: https://oauth.net/articles/authentication/
[42] Πηγή: https://developer.twitter.com/en/apps
[43] Πηγή: http://docs.tweepy.org/en/v3.5.0/index.html
[44] Πηγή: https://developer.twitter.com/en/docs/twitter-api/v1/rate-limits

χρησιμοποιώντας τα screen names των χρηστών αυτή τη φορά ως παράμετρο έγινε κλήση της μεθόδου api.user_timeline που επιστρέφει τη ζητούμενη πληροφορία σε μορφή JSON[45] αρχείων. Η μέθοδος αυτή επιστρέφει τα tweets του χρήστη με το συγκεκριμένο screen name για προκαθορισμένο χρονικό διάστημα ή πλήθος, το οποίο εισέρχεται ως παράμετρος στη συνάρτηση, καθώς και όλα τα δεδομένα που σχετίζονται με τα πεδία του προφίλ του. Το Twitter ονομάζει Tweet Object[46] το αρχείο που περιλαμβάνει την πληροφορία σχετικά με το tweet και User Object[47] το αρχείο που περιλαμβάνει την πληροφορία αναφορικά με το προφίλ του χρήστη. Στην παρούσα μελέτη η παράμετρος του χρόνου ορίστηκε ως 'oldest', δηλαδή από την δημιουργία του λογαριασμού, και έτσι λήφθηκαν όλα τα tweets που είχε αναρτήσει ο κάθε χρήστης. Τα δεδομένα των χρηστών αποθηκεύτηκαν σε ξεχωριστά JSON αρχεία για κάθε χρήστη με όνομα το αντίστοιχο screen name του, ενώ υπήρξε και έλεγχος με κατάλληλο μήνυμα για τους χρήστες που δεν είχαν αναρτήσει κάποιο tweet ώστε να αφαιρεθούν από τη μελέτη στα επόμενα στάδια της προεπεξεργασίας. Ο σχετικός κώδικας με τις δύο συναρτήσεις που δημιουργήθηκαν για την απόκτηση των δεδομένων ενός χρήστη παρατίθεται παρακάτω:

```
def get_user_screen_name(x):
    try:
        return api.get_user(x).screen_name
    except:
        return "NoneValue"

def get_all_tweets(fname, screen_name):
    alltweets = []
    try:
        new_tweets = api.user_timeline(screen_name = screen_name)
        alltweets.extend(new_tweets)
        if not alltweets:
            print ("No posts yet for %s" % screen_name)
        else:
            oldest = alltweets[-1].id - 1
            while len(new_tweets) > 0:
                new_tweets = api.user_timeline(screen_name = screen_name, max_id=oldest)
                alltweets.extend(new_tweets)
                oldest = alltweets[-1].id - 1
            #write tweet objects to JSON
            file = open(fname, 'w')
            for status in alltweets:
                file.write(json.dumps(status._json))
                file.write('\n')
            file.close()
            return alltweets
    except Exception as e:
        print("%s for user %s" % ( e, screen_name))
        return "No tweets fetched"
```

Για λόγους πληρότητας, αν και δεν χρησιμοποιήθηκε στην παρούσα εργασία, αναφέρουμε ότι το Twitter προσφέρει κι ένα Streaming API μέσω του οποίου ο προγραμματιστής μπορεί να αποκτήσει πρόσβαση στα tweets που αναρτώνται σε πραγματικό χρόνο. Το Streaming API

[45] Πηγή: https://www.json.org/json-en.html
[46] Πηγή: https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/overview/tweet-object
[47] Πηγή: https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/overview/user-object

είναι χρήσιμο για την ανάκτηση μεγάλου αριθμού tweets και οι περισσότεροι το χρησιμοποιούν για να κατασκευάσουν τις βάσεις δεδομένων τους, φιλτράροντας τα tweets ανάλογα με κάποιες προδιαγραφές. Έτσι για παράδειγμα, μπορούν να επιλέξουν να κρατήσουν μόνο τους χρήστες που έχουν ένα συγκεκριμένο αριθμό από tweets ή followers ή followings ή έχουν δημιουργήσει το προφίλ τους μετά από κάποια συγκεκριμένη ημερομηνία. Επίσης το streaming εφαρμόζεται και σε tweets που περιέχουν συγκεκριμένες λέξεις (keywords) ή hashtags για πιο εξειδικευμένες αναλύσεις των χρηστών του Twitter. Το Tweepy παρέχει στον προγραμματιστή τις κλάσεις StreamListener και Stream οι οποίες πραγματοποιούν το φιλτράρισμα των εισερχόμενων tweets και έπειτα την ανάλυση τους ώστε να κρατηθούν μόνο τα επιθυμητά στοιχεία.

Κεφάλαιο 6

6 Μελετώμενο πρόβλημα και προτεινόμενη προσέγγιση

6.1 Πρόκληση εύρεσης κατάλληλων συνόλων δεδομένων

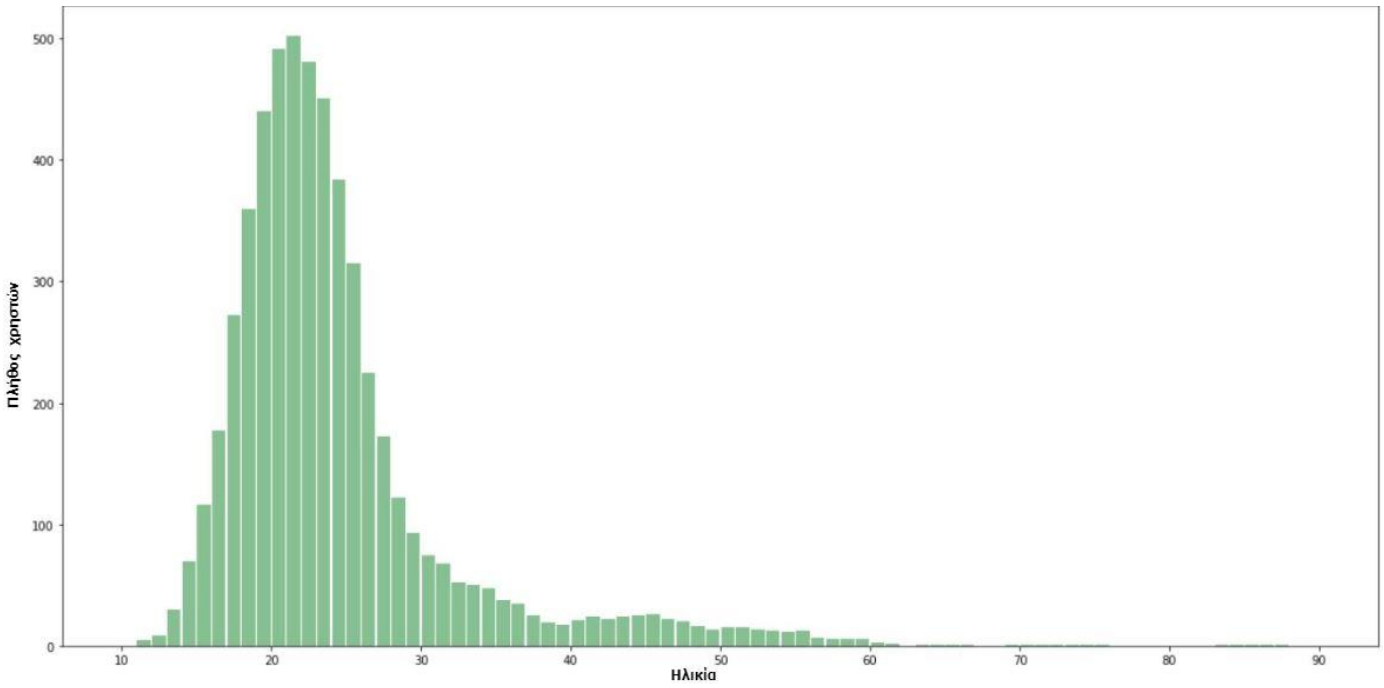
Το πρόβλημα της ανίχνευσης της ηλικίας των χρηστών στα μέσα κοινωνικής δικτύωσης και κυρίως στο Twitter κρίνεται μία αρκετά δύσκολη και απαιτητική πρόκληση, όπως διαπιστώθηκε και τονίστηκε από τις μελέτες της συναφούς βιβλιογραφίας που αναφέρθηκαν στην ενότητα 2 της παρούσας διπλωματικής εργασίας. Αρχικά, στην ήδη μεγάλη δυσκολία που εμφανίζει το πρόβλημα του προσδιορισμού της ηλικίας έρχεται να προστεθεί, ειδικά για το Twitter, και η απουσία επαρκών δεδομένων. Αυτό οφείλεται στο γεγονός ότι στο Twitter, σε αντίθεση με άλλα κοινωνικά δίκτυα όπως το Facebook, δεν υπάρχουν γενικά δεδομένα ηλικίας για τους χρήστες. Έτσι, ενώ παρέχει μεγάλο όγκο πληροφορίας σχετικά με τη δραστηριότητα των χρηστών που είναι εύκολα προσβάσιμος μέσω του API που διαθέτει, είναι αρκετά δύσκολο να βρεθούν σύνολα δεδομένων με ταυτοποιημένο το πεδίο της ηλικίας για την επεξεργασία και τη διεξαγωγή σχετικών ερευνών. Επίσης, κρίνεται δύσκολη η εξόρυξη της ηλικίας μέσω των ελεύθερων διαθέσιμων στοιχείων. Αυτό το εμπόδιο εμφανίστηκε νωρίς και στην παρούσα διπλωματική εργασία κατά τη διαδικασία αναζήτησης έγκυρων δεδομένων.

Επιπλέον, όσον αφορά την πλατφόρμα του Twitter, η αδυναμία εύρεσης ηλικιακών δεδομένων προκύπτει και από το γεγονός πως η ιστοσελίδα δεν απαιτεί τη συμπλήρωση στοιχείων σχετικών με την ηλικία για τη δημιουργία νέου λογαριασμού από έναν χρήστη. Το πεδίο “birthday” του προφίλ, όπως περιγράφηκε στην ενότητα 5.1, είναι προαιρετικό με αποτέλεσμα ελάχιστοι χρήστες να παρέχουν αυτή την πληροφορία. Στα πλαίσια της παρούσας εργασίας βρέθηκε ένα σύνολο δεδομένων με ταυτοποιημένες τις ηλικίες των χρηστών από το άρθρο [75]. Αυτό αποτέλεσε την πηγή της πληροφορίας που επεξεργαστήκαμε και δόθηκε ως είσοδος στους αλγορίθμους μηχανικής μάθησης που δοκιμάστηκαν.

Το σύνολο δεδομένων που μελετήθηκε περιλάμβανε συνολικά 1471 χρήστες με την πληροφορία για την ηλικία τους. Ωστόσο, πρόκειται για ένα μικρό δείγμα το οποίο συρρικνώθηκε ακόμα περισσότερο σε 1275 χρήστες, κατά τη διαδικασία λήψης των δεδομένων του προφίλ τους, λόγω των ιδιωτικών λογαριασμών που δεν επέτρεπαν πρόσβαση στις πληροφορίες τους. Επίσης, αφαιρέθηκε ένας μικρός αριθμός χρηστών, επειδή δεν ήταν δυνατό να υποστηριχθούν πολύγλωσσοι χρήστες. Η επεξεργασία των δεδομένων τους απαιτούσε τη χρήση μεταφραστή (google translator), ο οποίος όμως επέτρεπε περιορισμένο αριθμό αιτημάτων. Τελικά, για αυτό το λόγο διατηρήθηκαν πληροφορίες μόνο από αγγλόφωνα άτομα. Συνεπώς, γίνεται εύκολα αντιληπτό πως το υπάρχον σύνολο δεδομένων δεν μπορεί να περιέχει αντιπροσωπευτικό δείγμα χρηστών ώστε να είναι ικανό μετά την επεξεργασία που θα υποστεί να παράγει ασφαλή αποτελέσματα στις προβλέψεις για κάθε ηλικιακή ομάδα. Ιδιαίτερη προσοχή έπρεπε να δοθεί και στους μη ενεργούς χρήστες, διότι το δείγμα ήταν από το 2014 και ίσως περιλάμβανε απαρχαιωμένα δεδομένα σε συνδυασμό με την ιδιότητα της ηλικίας ως μεταβλητό μέγεθος από χρόνο σε χρόνο. Οι ανωτέρω λόγοι οδήγησαν σε μία τεχνική επέκτασης του συνόλου δεδομένων για την διεξαγωγή της μελέτης η οποία περιγράφεται στην επόμενη ενότητα.

6.2 Προτεινόμενη προσέγγιση

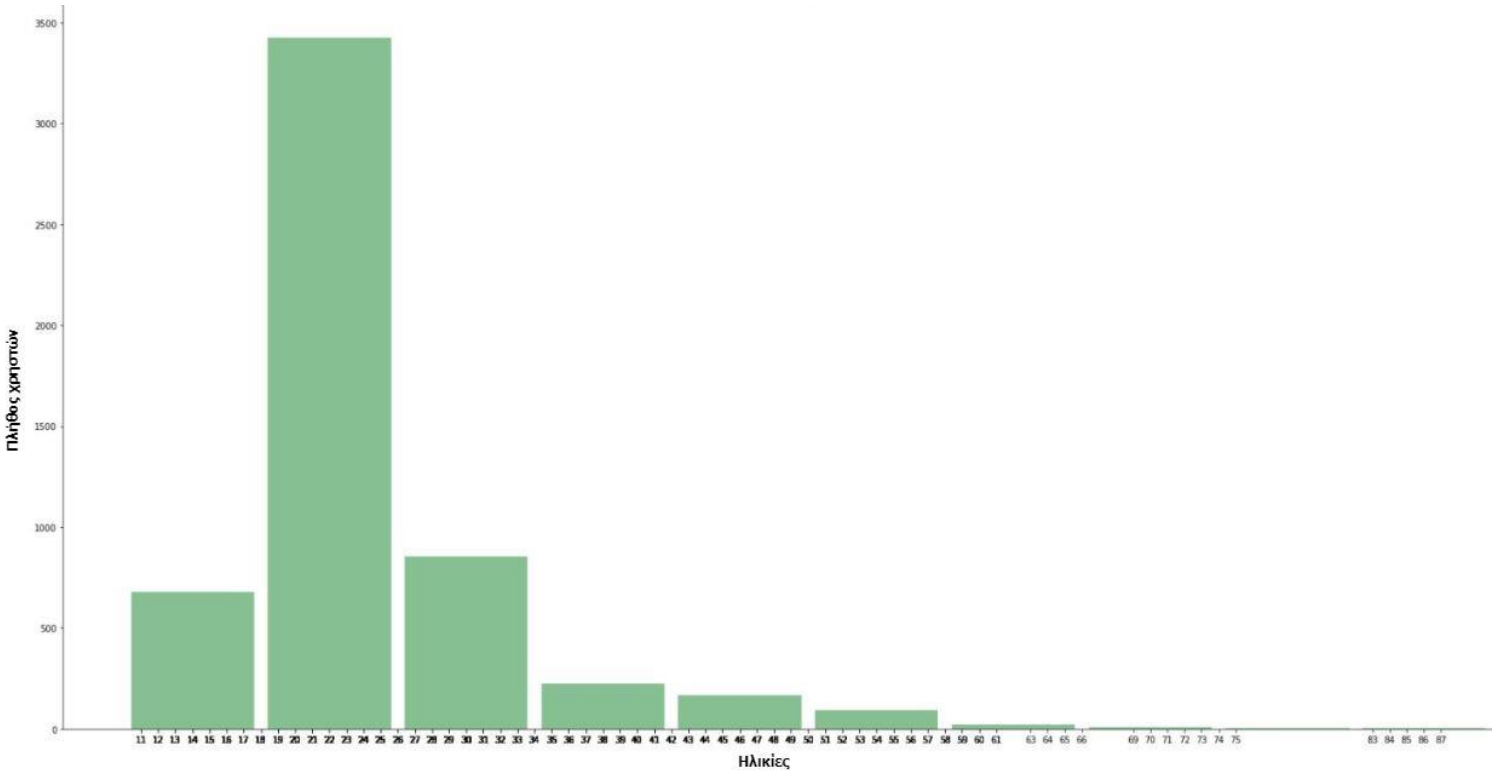
Ο σχεδιασμός της επίλυσης του προβλήματος επηρεάστηκε από τις δυσκολίες που ανέκυψαν και η προσέγγιση της παρούσας μελέτης βασίστηκε και καθοδηγήθηκε από άλλες παρόμοιες έρευνες. Στην προηγούμενη ενότητα επισημάνθηκε η απουσία επαρκών δεδομένων και η εύρεση μικρού συνόλου δεδομένων που δεν είναι αντιπροσωπευτικό για κάθε ηλικιακή κλάση. Αυτό το γεγονός οδήγησε σε ένα τέχνασμα που διεύρυνε το υπάρχουν δείγμα σε περίπου 5477 χρήστες. Η αύξηση του dataset έγινε μελετώντας κάθε χρήστη ανά έτος παρουσίας του στην πλατφόρμα του Twitter δίνοντας του μία αντίστοιχη ετικέτα. Με την λήψη των δεδομένων από το API προσδιορίστηκε ο χρόνος που ο χρήστης είχε λογαριασμό στην ιστοσελίδα και η ηλικία του για κάθε χρόνο. Έτσι αν ένας χρήστης είχε ενεργό προφίλ για 5 χρόνια, μελετήθηκε ως ξεχωριστός και μοναδικός χρήστης για κάθε μία από τις 5 αυτές χρονιές με την αντίστοιχη ηλικία. Για παράδειγμα αν ο χρήστης “George” έχει τρέχουσα ηλικία 25 έτη και διατηρούσε λογαριασμό από το 2015 μέχρι το 2017, μετατράπηκε σε τρεις καινούριους διακριτούς χρήστες, τον “George_25” με ηλικία 25 ετών για το 2017, τον “George_24” με ηλικία 24 ετών για το 2016 και τον “George_23” με ηλικία 23 ετών για το 2015. Μέσω αυτής τη τεχνικής αυξήθηκαν οι χρήστες για κάθε ηλικιακή ομάδα και ταυτόχρονα αντιμετωπίστηκε η μεταβολή της ηλικίας με την πάροδο των χρόνων. Η κατανομή των χρηστών στο dataset που προέκυψε παρατηρείται στην Εικόνα 6.1 παρακάτω. Από την κατανομή διαπιστώνεται πως η κατώτερη ηλικία για κάποιον χρήστη είναι τα 11 έτη και η ανώτερη τα 87 έτη.



Εικόνα 6.1: Κατανομή ηλικιών στο σύνολο δεδομένων

Το νέο σύνολο δεδομένων που δημιουργήθηκε οδήγησε στην σύλληψη της ιδέας να αντιμετωπιστεί το πρόβλημα της ανίχνευσης της ηλικίας από την παρούσα εργασία ως ένα πρόβλημα παλινδρόμησης (regression). Δηλαδή, να γίνονται προβλέψεις που θα δίνουν την ακριβή τιμή της ηλικίας του χρήστη, μία προσπάθεια που δεν έχει πραγματοποιηθεί ξανά στο παρελθόν από άλλες μελέτες. Πρόκειται, για μία πρωτότυπη και καινοτόμα προσέγγιση που σκοπεύει να παρέχει μεμονωμένες τιμές για την ηλικία και όχι ένα εύρος τιμών που μπορεί να ανήκει στοχεύοντας σε καλύτερο και πιο ακριβή προσδιορισμό. Συγχρόνως όμως, ώστε να

ακολουθηθεί η γενικότερη τάση της επιστημονικής κοινότητας πάνω στο πρόβλημα της ηλικίας και να είναι συγκρίσιμη η παρούσα μελέτη με τις ήδη υπάρχουσες θα γίνει και μία υλοποίηση μέσω ταξινόμησης (classification). Η ομαδοποίηση των χρηστών έγινε με βάση το διαχωρισμό που προέκυψε από το ιστόγραμμα του συνόλου δεδομένων. Το ιστόγραμμα υπολογίζει την βέλτιστη κατανομή των χρηστών σε ομάδες ανάλογα με τον αριθμό των ομάδων (bins) που θα του ζητηθούν στην είσοδο. Στην Εικόνα 6.2 παρουσιάζεται το ιστόγραμμα του συνόλου δεδομένων που χρησιμοποιήθηκε στην παρούσα εργασία για τιμή της παραμέτρου εισόδου ίση με 10.



Εικόνα 6.2: Ιστόγραμμα χρηστών για 10 ομάδες

Παρατηρείται από το ιστόγραμμα πως δεν υπάρχει ομοιόμορφη κατανομή των χρηστών σε όλες τις ηλικιακές ομάδες και ιδιαίτερα για τιμές άνω των 60 ετών, όπου σε αυτήν την περίπτωση ορισμένες ομάδες συνενώθηκαν. Τελικά, αυτό οδήγησε στην επιλογή 8 κατηγοριών για την ταξινόμηση των χρηστών, ώστε να υπάρχει όσο το δυνατόν πιο αντιπροσωπευτικό δείγμα για κάθε ηλικιακό εύρος. Οι ηλικιακές ομάδες φαίνονται στον πίνακα 6.1 και είναι:

Πίνακας 6.1: Ηλικιακές κατηγορίες

Δείκτης	**Ηλικιακές ομάδες**	**Χρήστες**
1	11-17	679
2	18-25	3424
3	26-33	856
4	34-41	226
5	42-49	168
6	50-57	91
7	58-65	20
8	66+	13

Μελετώντας τον πίνακα 2.1 παρατηρείται πως οι ηλικιακές ομάδες των μεγάλων έχουν πολύ λιγότερα δείγματα σε σχέση με αυτές των νέων. Αυτό το γεγονός θα δυσκολέψει σημαντικά την εκπαίδευση των αλγορίθμων στο να προβλέψουν αποτελεσματικά τιμές για όλες τις κλάσεις.

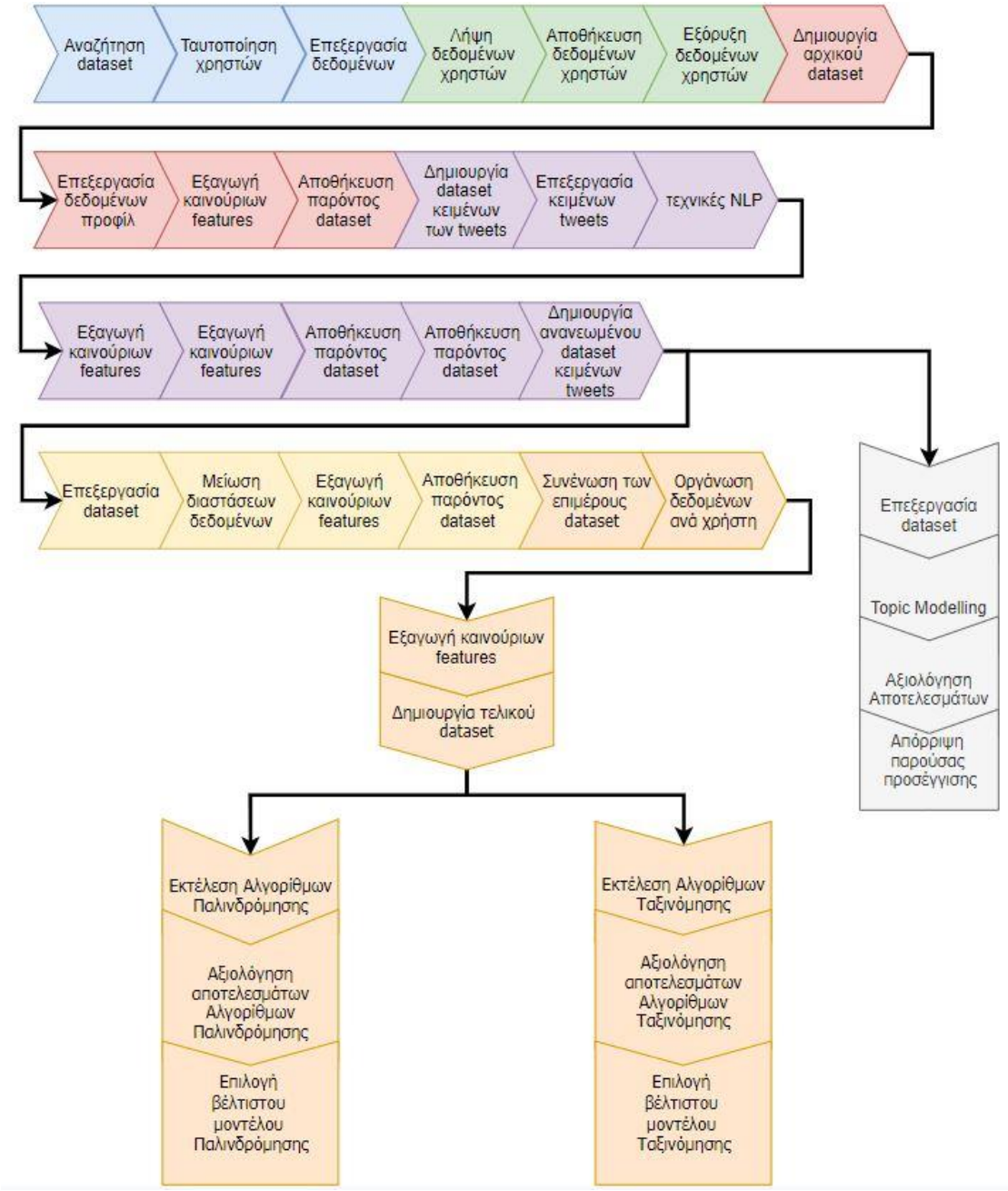
Από τις σχετικές μελέτες που αναφέρθηκαν στο κεφάλαιο 2 διαπιστώθηκε πως στις περισσότερες από αυτές αξιοποιούνται και αναλύονται τα λεξικογραφικά δεδομένα που προκύπτουν από τις δημοσιεύσεις των χρηστών. Αυτό συμβαίνει διότι προσφέρουν αρκετά χρήσιμη πληροφορία για τα ενδιαφέροντα των χρηστών και έτσι βοηθούν στην ανίχνευση της ηλικίας. Επίσης, κρίνονται ιδιαίτερα σημαντικά για τις προβλέψεις και τα στοιχεία του προφίλ των χρηστών μαζί με κάποια επιπλέον χαρακτηριστικά που προκύπτουν από την προεπεξεργασία τους. Ορισμένες προσεγγίσεις εκμεταλλεύονται μόνο τα γλωσσολογικά δεδομένα ή μόνο τα χαρακτηριστικά του προφίλ ή και τα δύο συνδυαστικά για την επίλυση του προβλήματος.

Η παρούσα έρευνα πραγματοποιήθηκε συνδυάζοντας τα λεξικογραφικά χαρακτηριστικά με αυτά του προφίλ των χρηστών, με στόχο την συμβολή τους για ένα καλύτερο αποτέλεσμα. Συνεπώς, το πρόβλημα απαιτεί και την επεξεργασίας φυσικής γλώσσας (NLP) για την επίλυσή του. Ταυτόχρονα υλοποιήθηκε και ένας topic modelling μηχανισμός για τον προσδιορισμό του θέματος των tweets και κατ' επέκταση των προτιμήσεων του κάθε χρήστη. Η συναισθηματική ανάλυση (sentiment analysis) των κειμένων, αν και υπήρχε σε ορισμένες παρόμοιες μελέτες, δεν εξετάστηκε ώστε να μην επιβαρύνει τη διαδικασία και την πολυπλοκότητα της λύσης. Η εφαρμογή της μεθόδου NLP δημιούργησε κάποια χαρακτηριστικά όπως το πλήθος των hashtags ή των tags-mentions που έχει χρησιμοποιήσει στα κείμενα του ο χρήστης.

Σχετικά με τα στοιχεία του προφίλ των χρηστών έγινε διαλογή για το ποια θα λάβουν μέρος στα πειράματα. Ο Πίνακας 2.1 που παρουσιάζει τα features παρόμοιων ερευνητικών έργων βοήθησε στην επιλογή αυτών που θα αποτελέσουν είσοδο σε αυτή την προσέγγιση. Έτσι αξιοποιήθηκαν ορισμένα από αυτά, όπως ο αριθμός των followers και των followings. Ωστόσο κάποια features που χρησιμοποιήθηκαν συχνά από άλλες σχετικές μελέτες εξαιρέθηκαν και στη θέση τους τοποθετήθηκαν άλλα λιγότερο δημοφιλή, όπως ο αριθμός των likes ή ηλικία του λογαριασμού. Σκοπός αυτής της τακτικής ήταν να εξεταστεί η αξία της πληροφορίας που μπορεί να παρέχουν και άλλα όχι εξίσου συχνά δοκιμασμένα features. Επίσης, ένα από τα στοιχεία που δεν αποτέλεσαν μέρος της ανάλυσης ήταν το βιογραφικό των χρηστών διότι η τεχνική που εφαρμόστηκε για την επέκταση του δείγματος προφανώς δεν επέτρεπε την αξιοποίηση του.

Το σύνολο των χαρακτηριστικών εισόδου που χρησιμοποιήθηκαν και προέρχονται είτε από το προφίλ του χρήστη είτε από την γλωσσολογική επεξεργασία των tweets του περιγράφονται αναλυτικά στην ενότητα 7.7.

Το διάγραμμα που φαίνεται στην Εικόνα 6.3 παρουσιάζει σχηματικά τη σύνοψη των βημάτων που ακολουθήθηκαν για την επίλυση του προβλήματος.



Εικόνα 6.3: Βήματα επίλυσης προβλήματος

6.3 Αυστηρή περιγραφή προβλήματος

Το πρόβλημα που επιλύει η παρούσα διπλωματική εργασία μπορεί να περιγραφεί από τα εξής βήματα:

1. Βρίσκεται σύνολο δεδομένων N χρηστών u_i για i από 1 έως N.
2. Γίνεται ταυτοποίηση των χρηστών και προκύπτει νέος αριθμός επιβεβαιωμένων χρηστών U_i μικρότερος του u_i.
3. Λαμβάνονται K δεδομένα για κάθε χρήστη U_i που περιλαμβάνονται στα T_j tweets τους, για τα οποία ισχύει $j > i$.

4. Γίνεται μαθηματική επεξεργασία των K δεδομένων για κάθε ένα T_j tweet και έπειτα από 4 στάδια προκύπτουν M δεδομένα για κάθε T_j tweet, όπου $M > K$.
5. Ο κάθε χρήστης U_i γίνεται X_n για κάθε εμφάνιση του για διαφορετική τιμή της ηλικίας του στο σύνολο των δεδομένων N.
6. Το σύνολο δεδομένων ομαδοποιείται με βάση κάθε χρήστη X_n.
7. Επί των X_n χρηστών δίνονται τα C_i χαρακτηριστικά για i από 1 έως M που αφορούν τα δεδομένα.
8. Με δεδομένα τα C_i, που πρόκειται για τα χαρακτηριστικά που καταγράφονται στον πίνακα 7.6 ζητείται να προσδιοριστεί:
    i. Η ακριβής τιμή της ηλικίας για κάθε χρήστη X_n.
    ii. Η ηλικιακή ομάδα που ανήκει ο κάθε χρήστης X_n.
9. Αξιολογούνται οι εκτιμήσεις που παράγονται στο βήμα 8.

Κεφάλαιο 7

7 Συλλογή, Προετοιμασία και Επεξεργασία Δεδομένων

Σε αυτό το κεφάλαιο περιγράφεται η διαδικασία συλλογής, αποθήκευσης και προεπεξεργασίας δεδομένων. Επίσης, αναλύεται και η διαδικασία εξαγωγής χαρακτηριστικών (features) που αποτελούν την είσοδο των αλγορίθμων μηχανικής μάθησης. Το σύνολο των εφαρμογών της διπλωματικής εργασίας αναπτύχθηκε σε γλώσσα προγραμματισμού Python.

7.1 Περιγραφή Αρχικού Συνόλου Δεδομένων

Το σύνολο δεδομένων που χρησιμοποιήθηκε για τη διεξαγωγή των πειραμάτων στην παρούσα μελέτη είχε δημιουργηθεί το 2014 και προήλθε από το άρθρο [75] που το διέθετε δημόσια στο διαδίκτυο[48]. Περιλάμβανε ταυτοποιημένες ηλικιακές πληροφορίες για 1471 χρήστες του Twitter. Αποτελούνταν από δύο στήλες, όπου η πρώτη περιείχε τους μοναδικούς κωδικούς ταυτοποίησης (user_id) των χρηστών και η δεύτερη την αντίστοιχη ηλικία τους. Στην Εικόνα 7.1 παρακάτω παρουσιάζεται ένα στιγμιότυπο των αρχικών δεδομένων.

	A	B
1	ID	Age
2	2326023704	13
3	917234222	13
4	1032892134	13
5	1304864995	13
6	2289726986	13
7	952514378	13
8	399079215	13
9	473287546	13
10	842638981	13

Εικόνα 7.1: Στιγμιότυπο αρχικών δεδομένων

7.2 Συλλογή και Αποθήκευση Πληροφοριών

Για κάθε χρήστη υπολογίστηκε το έτος γέννησης του αφαιρώντας από το 2014 την αντίστοιχη ηλικία του που αναγραφόταν στο σύνολο δεδομένων. Όπως αναφέρθηκε τα δεδομένα προέρχονταν από το 2014 και κρίθηκε απαραίτητο να προσαρμοστούν χρονολογικά με το έτος 2019 όπου ξεκίνησε η παρούσα εργασία. Για το λόγο αυτό αρχικά προστέθηκαν 5 χρόνια στην ηλικία κάθε χρήστη. Επίσης, έγινε λήψη του username του κάθε χρήστη μέσω της μεθόδου get_user(x).screen_name που παρέχει το Tweepy και αφαιρέθηκαν από τα δεδομένα οι χρήστες

[48] Πηγή: https://figshare.com/articles/dataset/_Who_Tweets_Deriving_the_Demographic_Characteristics_of_Age_Occupation_and_Social_Class_from_Twitter_User_Meta_Data_/1321154?file=1928555

που είχαν ιδιωτικά τα στοιχεία τους χωρίς να επιτρέπουν πρόσβαση σε αυτά. Μετά το πέρας αυτής της διαδικασίας στο dataset παρέμειναν 1275 χρήστες για τους οποίους λήφθηκαν μέσω του Tweepy τα δεδομένα του προφίλ τους και το χρονολόγιο (timeline) που περιλάμβανε όλη τη δραστηριότητα τους και τα tweets που είχαν δημοσιεύσει από τη δημιουργία του λογαριασμού τους στην πλατφόρμα. Το Tweepy καθώς και η διαδικασία συλλογής των πληροφοριών των χρηστών περιγράφηκαν στην ενότητα 5.3 παραθέτοντας και τον αντίστοιχο κώδικα που την υλοποιεί. Τέλος όπως επισημάνθηκε τα δεδομένα κάθε χρήστη αποθηκεύτηκαν σε αρχεία τύπου JSON με όνομα το username του, ώστε να είναι εύκολο να προσπελαστούν, ακολουθώντας τη δομή του Tweet Object το οποίο περικλείει και το User Object. Η συνάρτηση saveTweetsToFile() που υλοποιεί αυτή τη λειτουργία χρησιμοποιεί το username για την την αναζήτηση και τη συνάρτηση get_all_tweets() που παρουσιάστηκε στην ενότητα 5.3 για τη συλλογή των δεδομένων. Ακόμη, παρέχει και κατάλληλη πληροφορία για τους χρήστες που δεν έχουν δημοσιεύσει κάποιο tweet. Ο κώδικας φαίνεται παρακάτω:

```
def saveTweetsToFile(username):
    try:
        file_name = username +'.json'
        fname = os.path.join(store_files_dir, file_name)
        file = open(fname,'w')
        return get_all_tweets(fname, username)
    except Exception as e:
        print("%s for user %s" % ( e, screen_name))
        return "No tweets"
```

Επιπλέον εφαρμόστηκε η τεχνική του web scrapping μέσω της βιβλιοθήκης Beautiful Soup, που περιγράφεται στην ενότητα 4.1.8, με σκοπό την λήψη του βιογραφικού (description) και της ημερομηνίας γέννησης (birthday) από το προφίλ του χρήστη ώστε να γίνει επαλήθευση με το έτος γέννησης που είχε υπολογιστεί. Η αναζήτηση των πληροφοριών έγινε με κλειδί το username. Ωστόσο, από τη διαδικασία αυτή διατηρήθηκαν μόνο τα δεδομένα του βιογραφικού, διότι ελάχιστοι χρήστες είχαν συμπληρώσει την ηλικία τους. Έτσι δημιουργήθηκε το αρχικό dataset με στήλες το username, το user_id, το βιογραφικό, την ηλικία και το έτος γέννησης. Η συνάρτηση που υλοποιεί την αναζήτηση των ανωτέρω στοιχείων είναι ο εξής:

```
def getUserDescriptionAndBirthday(username):
    URL = "https://twitter.com/" + username
    r = requests.get(URL)
    soup = BeautifulSoup(r.content, 'html5lib')
    description = soup.find('p',{'class':"ProfileHeaderCard-bio"}).text
    birthday = soup.find('span',{'class':"ProfileHeaderCard-birthdateText"}).text.strip()
    return description, birthday
```

7.3 Προετοιμασία Δεδομένων

Η κατασκευή του dataset και η εξαγωγή των χαρακτηριστικών άρχισε από την επεξεργασία των δεδομένων των JSON αρχείων όπου αποθηκεύτηκαν τα tweets κάθε χρήστη. Τα αρχεία αυτά περιλαμβάνουν πληροφορία σχετική με τα tweets, ενώ ταυτόχρονα ενσωματώνουν πληροφορίες αναφορικά με το προφίλ του χρήστη. Τα δεδομένα χωρίστηκαν σε έξι υποσύνολα τα οποία μελετήθηκαν χωριστά, με σκοπό να μην υπάρξουν προβλήματα μνήμης στο υπολογιστικό σύστημα που εκτέλεσε την λειτουργία. Η προσπέλαση των αρχείων κάθε χρήστη

πραγματοποιήθηκε με βάση τα πεδία του Tweet Object για να εξαχθούν πληροφορίες για τα tweets, αλλά και με βάση το User Object για να ληφθούν οι πληροφορίες του προφίλ. Για αυτό το λόγο δημιουργήθηκαν δύο οντότητες, η tweet_attributes και η user_attributes. Επειδή κάποια πεδία των δύο οντοτήτων είχαν κοινή ονομασία τοποθετήθηκε πρόθεμα ώστε να διαφοροποιηθούν. Έτσι για τα στοιχεία της οντότητας user_attributes δημιουργήθηκαν νέες ετικέτες όπου προστέθηκε το πρόθεμα "user". Τα πεδία που εξορύχθηκαν για κάθε περίπτωση καθώς και η περιγραφή τους φαίνονται στους πίνακες 7.1 και 7.2 παρακάτω:

Πίνακας 7.1: Χαρακτηριστικά της οντότητας Tweet

Tweet Attributes	
Πεδίο	**Περιγραφή**
text	Το κείμενο που περιέχει το tweet σε μορφή UTF-8
user	Ο χρήστης που δημοσίευσε το tweet (περιλαμβάνει το User Object)
lang	Ο δείκτης που δηλώνει σε ποια γλώσσα έχει γραφτεί το tweet • **en**: για τα tweet που έχουν γραφτεί στην αγγλική γλώσσα • **und**: για τα tweet όπου η γλώσσα συγγραφής δεν μπορεί να ανιχνευθεί
retweet_count	Το πλήθος των φορών που το συγκεκριμένο tweet αναδημοσιεύθηκε
retweeted_status	Υποδεικνύει αν το συγκεκριμένο tweet αποτελεί αναδημοσίευση κάποιου άλλου tweet
retweeted	Υποδεικνύει αν το συγκεκριμένο tweet έχει αναδημοσιευθεί από κάποιον άλλον πιστοποιημένο χρήστη
favorite_count	Το πλήθος των likes που έλαβε το tweet
created_at	Η ώρα που δημιουργήθηκε το tweet εκφρασμένη στην συντονισμένη παγκόσμια ώρα (UTC)
is_quote_status	Η μεταβλητή αληθείας (true ή false) που δείχνει αν το tweet είναι quoted, δηλαδή αν πρόκειται για ένα retweet που περιέχει το προσωπικό σχόλιο του χρήστη που το δημοσίευσε

Πίνακας 7.2: Χαρακτηριστικά της οντότητας User

User Attributes	
Πεδίο	**Περιγραφή**
user_id	Ο κωδικός που ταυτοποιεί μοναδικά τον χρήστη
user_name	Το όνομα του χρήστη όπως συμπληρώθηκε κατά τη δημιουργία του λογαριασμού του
screen_name	Το όνομα (username) που εμφανίζεται στο προφίλ του χρήστη και είναι μοναδικό
user_followers_count	Το πλήθος των followers που έχει το προφίλ
user_friends_count	Το πλήθος των followings, δηλαδή των λογαριασμών που ο χρήστης ακολουθεί
user_listed_count	Το πλήθος των δημόσιων ομάδων όπου ο χρήστης είναι μέλος
user_favourites_count	Το πλήθος των tweets που ο χρήστης έχει δηλώσει ότι του αρέσουν
user_statuses_count	Το πλήθος των tweets που έχει δημοσιεύσει ο χρήστης συμπεριλαμβανόμενων των retweets
user_created_at	Η ώρα που δημιουργήθηκε ο λογαριασμός εκφρασμένη στην συντονισμένη παγκόσμια ώρα (UTC)

Η διαδικασία εξόρυξης των δεδομένων και εισαγωγής τους σε ένα data frame επαναλαμβάνεται για κάθε ένα από τα έξι υποσύνολα. Τα παραγόμενα datasets αποθηκεύονται σε CSV αρχεία και περιλαμβάνουν σε κάθε σειρά ένα tweet και σε κάθε στήλη τα αντίστοιχα δεδομένα του. Οι στήλες των datasets είναι τα στοιχεία των οντοτήτων για τα tweets και τους χρήστες με τις ετικέτες όπως περιγράφηκαν στους πίνακες 7.1 και 7.2. Ο σχετικός κώδικας δίνεται παρακάτω:

```
tweet_attributes = ['text', 'user', 'lang', 'retweet_count', 'retweeted_status',
                    'retweeted', 'favorite_count', 'created_at', 'is_quote_status']
user_attributes = ['id', 'name', 'screen_name', 'followers_count', 'friends_count',
                   'listed_count', 'favourites_count', 'statuses_count', 'created_at']

def createDFperFile(fname,file):
    l = []
    with open(os.path.join(data_dir, file)) as f:
        for line in f.readlines():
            l.append(json.loads(line))
    tmp = {}
    for i in range(len(l)):
        for j in l[i]:
            if j in tweet_attributes:
                if j =='user':
                    for k in l[i][j]:
                        if k in user_attributes:
                            if 'user_' + str(k) in tmp.keys():
                                tmp['user_' + str(k)].append(l[i][j][k])
                            else:
                                tmp['user_' + str(k)] = [l[i][j][k]]
                    continue
                if j in tmp.keys():
                    tmp[j].append(l[i][j])
                else:
                    tmp[j] = [l[i][j]]
    df_cur = pd.DataFrame(dict([(k,Series(v)) for k,v in tmp.items()]))
    return df_cur

def readTweetsFromFile():
    try:
        df = pd.DataFrame(columns=['text', 'user', 'lang', 'retweet_count',
                                   'retweeted_status', 'retweeted', 'favorite_count',
                                   'created_at', 'is_quote_status','user_id', 'user_name',
                                   'user_screen_name', 'user_followers_count',
                                   'user_friends_count', 'user_listed_count',
                                   'user_favourites_count', 'user_statuses_count',
                                   'user_created_at'])
        for file_name in os.listdir(data_dir):
            if file_name.endswith(".json"):
                fname = os.path.join(data_dir, file_name)
                df = pd.concat([df, createDFperFile(fname,file_name)], sort=True)
        return df
    except Exception as e:
        print("%s for user" % e)
        return "No tweets fetched"
```

Τα σύνολα δεδομένων που δημιουργήθηκαν συνενώθηκαν κατάλληλα με το αρχικό που περιγράφηκε στην ενότητα 7.2 και περιλάμβανε τις πληροφορίες user_id, ηλικία, username, βιογραφικού και το έτος γέννησης. Τα συνολικά δεδομένα αποθηκεύτηκαν σε αρχεία τύπου CSV για να υποστούν περαιτέρω επεξεργασία.

Το επόμενο βήμα ήταν να υπολογιστούν ορισμένα χρονολογικά δεδομένα. Έτσι βρέθηκε η χρονολογία που γράφτηκε το κάθε tweet από το πεδίο "created_at", μέσω της οποίας υπολογίστηκε το έτος που ο χρήστης δημοσίευσε κάποιο tweet για τελευταία φορά. Αυτά τα στοιχεία οδήγησαν στη δημιουργία του χαρακτηριστικού "account_age", που σχετίζεται με το χρονικό διάστημα που ο χρήστης διαθέτει λογαριασμό στην πλατφόρμα.

Επιπλέον εξήχθη πληροφορία σχετικά με τη ζώνη ώρας μέσα στην ημέρα όπου αναρτήθηκε το tweet. Επιλέχθηκαν τρεις διαφορετικές ζώνες, η πρωινή ζώνη με την ετικέτα "morning_tweets" για τις ώρες από 8:00 έως 15:59, η απογευματινή ζώνη με την ετικέτα "evening_tweets" για τις ώρες 16:00 μέχρι 23:59 και η νυχτερινή ζώνη με την ετικέτα "night_tweets" που αφορούσε τις ώρες 00:00 έως 07:59. Η εξαγωγή αυτών των δεδομένων έγινε αξιοποιώντας το πεδίο "created_at". Η συνάρτηση που υπολογίζει τη ζώνη είναι η εξής:

```
def findTweetZone(tweet_time):
    morning_s = datetime.time(8, 0, 0)
    morning = (morning_s.hour * 60 + morning_s.minute) * 60 + morning_s.second
    evening_s = datetime.time(16, 0, 0)
    evening = (evening_s.hour * 60 + evening_s.minute) * 60 + evening_s.second
    night_s = datetime.time(23, 59, 59)
    night = (night_s.hour * 60 + night_s.minute) * 60 + night_s.second
    tweet_time = (i.hour * 60 + i.minute) * 60 + i.second
    if (t >= morning and t < evening):
        return "morning"
    elif (t >= evening and t <= night):
        return "evening"
    else:
        return "night"
```

Το πεδίο "account_age" και οι τρεις νέες ετικέτες "morning_tweets", "evening_tweets" και "night_tweets" προστέθηκαν στις στήλες του συνολικού dataset ως τέσσερα νέα features.

Ακόμη υπήρξε η δημιουργία δύο νέων πεδιών, των "aged_user_name" και "aged_user_id". Τα νέα στοιχεία ερμηνεύονται ως το όνομα και το used_id του χρήστη συνοδευόμενα με την ηλικία του όταν δημοσίευσε το αντίστοιχο tweet. Αυτά τα νέα πεδία χρησιμοποιήθηκαν σε μεταγενέστερο στάδιο για την τεχνική επαύξησης των χρηστών του dataset, λόγω της οποίας δεν μελετήθηκε το βιογραφικό του κάθε χρήστη και αφαιρέθηκε από τα δεδομένα.

Τα υποσύνολα των δεδομένων συνενώθηκαν σε ένα αρχείο που περιλάμβανε συνολικά 2.620.724 tweets με τα αντίστοιχα δεδομένα τους που υπολογίστηκαν σε αυτό το στάδιο. Πραγματοποιήθηκε έλεγχος μέσω του πεδίου "lang" ώστε να διατηρηθούν μόνο τα tweets που είναι γραμμένα στην αγγλική γλώσσα και τα υπόλοιπα να εξαιρεθούν από την διαδικασία του NLP. Αυτό το φιλτράρισμα οδήγησε σε ένα dataset που περιλάμβανε 2.402.792 tweets. Τα δεδομένα διαχωρίστηκαν σε στατιστικά και κείμενο και αποθηκεύτηκαν σε δύο διαφορετικά αρχεία με κλειδί το νέο πεδίο "aged_user_id", ώστε να είναι πιο διαχειρίσιμα.

7.4 Επεξεργασία φυσικής γλώσσας στα tweets

Το επόμενο στάδιο επεξεργασίας ήταν η εφαρμογή αρκετών από τις τεχνικές της επεξεργασίας φυσικής γλώσσας (NLP), με σκοπό να εξαχθεί επιπλέον χρήσιμη πληροφορία για κάθε tweet και έγινε στο δεύτερο αρχείο που περιλάμβανε μόνο τα κειμένα και το πεδίο "aged_user_id". Τα λεξικογραφικά δεδομένα μπορούν να αποτελέσουν σημαντική πηγή δεδομένων και να βοηθήσουν σε μεγάλο βαθμό στην διεξαγωγή των προβλέψεων της ηλικίας των χρηστών. Όπως παρατηρήθηκε από τη μελέτη παρόμοιων ερευνών η ανάλυση των γλωσσολογικών στοιχείων του χρήστη και ο τρόπος που γράφει μπορούν να συσχετιστούν με την ηλικία του. Συνεπώς και στην παρούσα έρευνα θα γίνει προσπάθεια εξόρυξης χαρακτηριστικών από τα tweets.

Η απόφαση επιλογής των σχετικών με τα κείμενα χαρακτηριστικών που θα χρησιμοποιηθούν στην ανάλυση πραγματοποιήθηκε μελετώντας τα στοιχεία του πίνακα 2.1, όπου παρουσιάζονται αυτά που χρησιμοποιήθηκαν σε συναφεή άρθρα. Έτσι προέκυψαν έξι νέα features για το dataset που φαίνονται στον πίνακα 7.3 παρακάτω:

Πίνακας 7.3: Λεξικογραφικά χαρακτηριστικά

Features	Περιγραφή
total_emoji	Το συνολικό πλήθος των emoji (emoticons) που έχει χρησιμοποιήσει ο χρήστης στα tweets του
total_hashtags	Το συνολικό πλήθος των hashtags (#) που έχει χρησιμοποιήσει ο χρήστης στα tweets του
total_len_text	Το συνολικό μήκος σε χαρακτήρες όλων των tweets που έχει δημοσιεύσει ο χρήστης
total_tags	Το συνολικό πλήθος των tags και mentions (@) που έχει κάνει ο χρήστης στα tweets του
total_URLs	Το συνολικό πλήθος των URLs (σύνδεσμοι, links) που έχει δημοσιεύσει ο χρήστης μέσω των tweets του
total_count_retweet	Το συνολικό πλήθος των tweets που έχει δημοσιεύσει ο χρήστης και είναι retweets (@RT)

Αρχικά η διαδικασία εξόρυξης των στοιχείων του πίνακα 7.3 έγινε μεμονωμένα για κάθε tweet και το συνολικό πλήθος υπολογίστηκε στην φάση επεξεργασίας κατά την οποία τα tweets του συνόλου δεδομένων ομαδοποιήθηκαν ανά χρήστη. Αναπτύχθηκαν συναρτήσεις οι οποίες ανίχνευσαν τα συγκεκριμένα στοιχεία μέσα στο κείμενο των tweets. Επίσης, υλοποιήθηκαν συναρτήσεις που αφαιρούσαν τα στοιχεία αυτά από το κείμενο καθώς και αντικαθιστούσαν ειδικούς χαρακτήρες HTML με το αντίστοιχο σύμβολο, όπως "&lt" με το "<". Αυτό έγινε με σκοπό να καθαριστεί το κείμενο από σύμβολα και μη αλφαριθμητικούς χαρακτήρες ώστε να είναι μπορεί να πραγματοποιηθεί περαιτέρω λεξικογραφική επεξεργασία στο περιεχόμενο του. Σχετικά με την εύρεση των emoticons μέσα στο κείμενο των tweets χρησιμοποιήθηκαν δύο συναρτήσεις. Η πρώτη που εφαρμόστηκε ήταν η emoji_count(), η οποία διατίθεται μέσω της βιβλιοθήκης emoji[49] της python και υπολογίζει το πλήθος των emoji στο κείμενο εισόδου. Ωστόσο, παρατηρείται σύνηθες το φαινόμενο οι χρήστες να εκφράζουν στα tweets τους τα emoji μέσω συμβόλων, σημείων στίξης και ειδικών χαρακτήρων όπως για παράδειγμα το :-) αντί του γνωστού emoji 😊 για να δηλώσουν τη χαρά τους. Αυτό το γεγονός οδήγησε στην χρήση μίας δεύτερης συνάρτησης, η οποία μέσω τεχνικής NLP ανιχνεύει αν στο κείμενο των tweets εμφανίζεται μία σειρά ειδικών χαρακτήρων ή σημείων στίξης διαφορετικών μεταξύ τους που συνθέτουν ένα emoji. Για την διαδικασία NLP εγκαταστάθηκε και χρησιμοποιήθηκε το μοντέλο "en_core_web_sm"[50] για την αγγλική γλώσσα που παρέχεται μέσω του πακέτου

[49] Πηγή: https://pypi.org/project/emoji/
[50] Πηγή: https://spacy.io/usage/models

SpaCy. Ο κώδικας παρακάτω παραθέτει τις συναρτήσεις που υλοποιήθηκαν σε αυτό το στάδιο της επεξεργασίας.

```
# Find Hashtags
def findHashtags(x):
    return re.findall(r'#\w+', x)

# Remove Hashtags
def removeHashtags(x):
    return ' '.join(re.sub(r'#\w+', ' ',x).split())

# Find Tags
def findTags(x):
    return re.findall(r'@\w+', x)

# Remove tags
def removeTags(x):
    return ' '.join(re.sub(r'@\w+', ' ',x).split())

# Find retweet
def findRetweet(x):
    return re.findall(r'RT @\w+', x)

# Remove retweets
def removeRetweets(x):
    return ' '.join(re.sub(r'RT :+', ' ',x).split())

# Find URLs
def findURLs(x):
    return re.findall('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\
                      (\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+', x)

# Remove URLs
def removeURLs(x):
    return re.sub('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+]|[!*\(\),
                  ]|(?:%[0-9a-fA-F][0-9a-fA-F]))+', ' ',x)

# Find Emoji written with Punctuation
def findEmojiWithPunct(x):
    emoji_count = 0
    for token in nlp(x):
        if token.pos_ is 'PUNCT':
            if len(token.text) >1:
                c1 = token.text[0]
                c2 = token.text[1]
                if c1 != c2 :
                    emoji_count = emoji_count +1
    return emoji_count

# Remove Emoji from text
def strip_emoji(text):
    return re.sub(emoji.get_emoji_regexp(), r' ', text)
```

```
# Find length of the text
def len_text (text):
    return len(text)

# Replace HTML special characters
def replaceCharsHTML(x):
    x = x.replace("&lt;", "<")
    x = x.replace("&gt;", ">")
    x = x.replace("&amp;", "&")
    return s
```

Συνολικά η διαδικασία επεξεργασίας των tweets με την εφαρμογή των ανωτέρω συναρτήσεων διήρκησε 4 ώρες και 36 λεπτά περίπου. Πρόκειται για μία αρκετά χρονοβόρα διαδικασία, ιδιαίτερα απαιτητική ως προς τη μνήμη και την επεξεργαστική ισχύ που καταναλώνει. Μετά το πέρας αυτού του βήματος, τα νέα δεδομένα προστέθηκαν στο υπάρχον dataset κειμένου, ενώ ακόμη προστέθηκε μία στήλη με την ετικέτα "clean_text" η οποία περιλάμβανε το καθαρό κείμενο χωρίς τα emoji και τα υπόλοιπα σύμβολα. Το νέο dataset αποθηκεύτηκε σε αρχείο CSV, έυκολα προσπελάσιμο. Επίσης, μέσω αυτού δημιουργήθηκε ένα νέο αρχείο CSV που περιλάμβανε τα πεδία "aged_user_id", "clean_text" και "age", ώστε να απομονωθεί το καθαρό κείμενο και να μειωθεί το μέγεθος του με στόχο να επιτευχθεί καλύτερη απόδοση κατά την επεξεργασία του.

Στη συνέχεια, το επόμενο βήμα της επεξεργασίας περιλάμβανε την NLP ανάλυση του καθαρού κειμένου των tweets για την εξαγωγή πληροφορίας σχετικής με τα ενδιαφέροντα των χρηστών. Στόχος αυτής της φάσης της μελέτης ήταν να γίνει αναγνώριση της εννοιολογικής σημασίας των λέξεων που χρησιμοποιούνται στα tweets και ο συνολικός υπολογισμός εμφανίσεων τους. Η λειτουργία αυτή υλοποιήθηκε με τη χρήση του μοντέλου "en_core_web_sm"[50] που αναφέρθηκε νωρίτερα και με τη χρήση του NER[51] (named entity recognition) μηχανισμού που παρέχει το πακέτο SpaCy κατά τη ροή των διεργασίων (pipeline) NLP. Η μέθοδος αυτή προσδίδει ετικέτα οντότητας (entity) σε κάθε γνωστή κατηγοριοποιημένη λέξη που ανιχνεύει στην πρόταση που επεξεργάζεται, ενώ αφήνει χωρίς ετικέτα όσες λέξεις δεν γνωρίζει το μοντέλο της. Το μοντέλο περιλαμβάνει ένα μεγάλο πλήθος οντοτήτων όπως για αριθμούς (CARDINAL), για ανθρώπους (PERSON), για ζώα (ANIMAL), για γεωπολιτικές περιοχές (GPE) όπως χώρες ή πόλεις, για οργανισμούς ή εταιρίες (ORG) και άλλα. Το pipeline εργασιών που διαθέτει το πακέτο SpaCy είναι ευέλικτο και προσαρμόσιμο δίνοντας τη δυνατότητα στον προγραμματιστή να επέμβει και να κάνει αλλαγές ανάλογα με τις ανάγκες του[53]. Το ίδιο συμβαίνει και με τη διαδικασία NER η οποία μπορεί να εμπλουτιστεί με νέες οντότητες ώστε να αναγνωρίζει επιπλέον λέξεις.

Στην παρούσα εργασία αναπτύχθηκαν μοντέλα για αναγνώριση ορισμένων νέων οντοτήτων που λειτουργούν και παρεμβάλλονται στη ροή πριν την NER διεργασία και την εξαγωγή του πλήθους εμφάνισης τους στα tweets. Έτσι με βάση τα πιο δημοφιλή θέματα της καθημερινότητας και τα ενδιαφέροντα των ανθρώπων επιλέχθηκαν και ορίστηκαν εννέα νέες οντότητες στην αγγλική γλώσσα, ώστε να ανιχνεύσουν την εννοιολογική σημασία και κατ' επέκταση το θέμα των tweets που δημοσιεύουν οι χρήστες του Twitter. Στις οντότητες αυτές συμπεριλήφθηκε μία που φανέρωνε τη χρήση λέξεων γλώσσας (slang) η οποία χρησιμοποιείται ευρέως από τους νεούς. Ο ορισμός τους πραγματοποιήθηκε με τη σύνθεση αρχείων μορφής txt που περιείχαν λέξεις κλειδιά οι οποίες ήταν ενδεικτικές και χαρακτήριζαν κάθε οντότητα. Για κάθε οντότητα χρησιμοποιήθηκαν πάνω από 50 λέξεις κλειδία οι οποίες βρέθηκαν μέσω του Google Trends[52]. Για παράδειγμα η λέξη "football" ήταν χαρακτηριστική της νέας οντότητας "SPORTS", η λέξη "exams" υποδείκνυε την νέα οντότητα "SCHOOL", ενώ η λέξη "bro" την νεά οντότητα "SLANG". Η λειτουργία αυτή υλοποιήθηκε μέσω ενός entity matcher που έλαβε

[51] Πηγή: https://spacy.io/usage/linguistic-features#named-entities
[52] Πηγή: https://trends.google.com/trends/?geo=GB

ως είσοδο τα αρχεία που περιείχαν τις λέξεις κλειδιά, τις οποιές προσπέλασε μία μία και τους έδωσε την αντίστοιχη ετικέτα.

Όπως αναφέρθηκε, η διεργασία για την αναγνώριση των νέων οντοτήτων προηγείται στη ροή της υπάρχουσας NER του πακέτου SpaCy. Αυτό συμβαίνει ώστε να δίνεται προτεραιότητα στις νέες οντότητες για τον χαρακτηρισμό των λέξεων του κειμένου και έχει οριστεί στον κώδικα που υλοποιεί τη συγκεκριμένη λειτουργικότητα. Έτσι, σε περιπτώσεις όπου κάποιες λέξεις είναι κοινές για τις νέες και τις ήδη υπάρχουσες οντότητες του μοντέλου NER, θα διατηρείται μόνο η ετικέτα της νέας οντότητας. Για παράδειγμα, η λέξη "apple" λαμβάνει την ετικέτα της νέας οντότητας "TECHNOLOGY" και όχι την ετικέτα "ORG" με την οποία την χαρακτήριζε το αρχικό μοντέλο NER. Ο Πίνακας 7.4 παρουσιάζει τις εννέα νέες οντότητες μαζί με ορισμένες ενδεικτικές λέξεις-κλειδιά, διότι δε γίνεται να καταγραφούν όλες στο παρόν κείμενο, που τις ταυτοποιούν.

Πίνακας 7.4: Πίνακας νέων οντοτήτων για την διεργασία NER

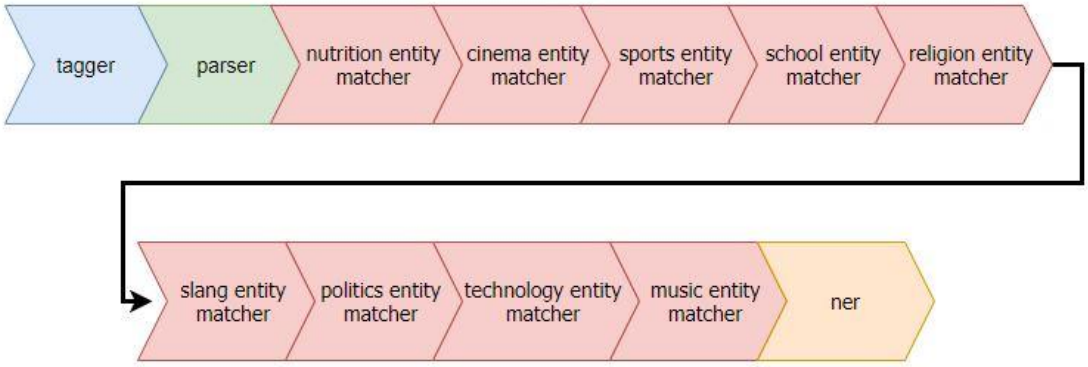
Οντότητα	**Λέξεις-κλειδιά**
CINEMA	cinema, movie, oscars, episode, film, comedy etc.
MUSIC	musician, pop, rock, jazz, disco, piano, guitar etc.
NUTRITION	cook, cheese, meat, fish, chicken, sugar, protein etc.
POLITICS	law, government, elections, parliament, economy etc.
RELIGION	christians, orthodox, islam, buddhism, church etc.
SCHOOL	exams, classroom, degree, teacher, college, student etc.
SPORTS	football, basketball, volleyball, team, mundial, fifa etc.
TECHNOLOGY	apple, microsoft, hardware, software, pc, smartphone etc.
SLANG	bro, omg, yolo, damn, cool, chill, yup, af, bae etc.

Ο κώδικας που υλοποιεί τον entity matcher και τις λειτουργίες που περιγράφηκαν είναι ο εξής:

```
# Define Entity Matcher
class EntityMatcher(object):

    def __init__(self, name, nlp, terms, label):
        self.name = name + "_entity_matcher"
        patterns = [nlp.make_doc(text) for text in terms]
        self.matcher = PhraseMatcher(nlp.vocab)
        self.matcher.add(label, None, *patterns)

    def __call__(self, doc):
        matches = self.matcher(doc)
        seen_tokens = set()
        new_entities = []
        entities = doc.ents
        for match_id, start, end in matches:
            if start not in seen_tokens and end - 1 not in seen_tokens:
                new_entities.append(Span(doc, start, end, label=match_id))
                entities = [e for e in entities if not (e.start < end and e.end > start)]
                seen_tokens.update(range(start, end))
        doc.ents = tuple(entities) + tuple(new_entities)
        return doc
```

```
# Find Entities
def getEntities(x):
    return [(ent.text, ent.label_) for ent in x.ents]

# Load from file keywords
def load_keywords():
    data_dir = '/path/keywords'
    terms = list()
    with open(os.path.join(data_dir, ent +'_keywords.txt')) as fp:
        for line in fp:
            terms.append(line.strip().lower() ,)
    return terms
```

Όπως επισημάνθηκε, με την εφαρμογή του entity matcher και την αλλαγή της ροής εργασιών[53] κατά την διαδικασία NLP, προέκυψε νέο pipeline[54] για την παρούσα μελέτη το οποίο φαίνεται στην Εικόνα 7.2 ως εξής:



Εικόνα 7.2: Ροή εργασιών NLP για την εννοιολογική ανίχνευση των tweets

Η ανάλυση της εννοιολογικής σημασίας και η κατηγοριοποιήση σε θεματικές ομάδες των λέξεων που χρησιμοποιούν οι χρήστες στα tweets είχε σκοπό την καταγραφή των ενδιαφερόντων τους και το πόσο συχνά αναφέρονται σε κάποιο θέμα. Για την μελέτη του κειμένου εφαρμόστηκαν οι τεχνικές tokenization, stop-words, lemmatization και POS-tagging που περιγράφηκαν στην ενότητα 3.4. Για την εξαγωγή των tokens από το κείμενο δημιουργήθηκε μία μέθοδος που ενσωμάτωνε από το πακέτο SpaCy τις τεχνικές stop-words και lemmatization, ενώ έλεγχε μέσω του διαθέσιμου POS-tagging μηχανισμού αν η λέξη είναι σημείο στίξης ώστε να μην το λάβει υπόψη στην ανάλυση. Επίσης, αναπτύχθηκαν συναρτήσεις όπου υπολόγιζαν τη συχνότητα εμφάνισης κάθε οντότητας στο περιεχόμενο των tweets, λαμβάνονας ως είσοδο τα tokens που δημιουργήθηκαν για κάθε κείμενο. Έτσι προστέθηκαν εννέα νέα features ένα για κάθε οντότητα για να εκφράσουν τη συχνότητα εμφάνισής τους και φαίνονται στον πίνακα 7.5 παρακάτω.

Συνολικά η διαδικασία επεξεργασίας των tweets με την εφαρμογή των παρακάτω συναρτήσεων διήρκησε 4 ώρες και 35 λεπτά περίπου, παρόμοια με αυτή της εξαγωγής των στατιστικών δεδομένων του προηγούμενου βήματος. Παρατηρείται πως κάθε NLP διεργασία που δοκιμάστηκε είναι ιδιαίτερα χρονοβόρα και απαιτεί αρκετή μνήμη και την επεξεργαστική ισχύ. Μετά την ολοκλήρωση αυτού του σταδίου επεξεργασίας, τα νέα δεδομένα προστέθηκαν στο υπάρχον dataset κειμένου το οποίο αποθηκεύτηκε και αυτό σε αρχείο CSV.

[53] Πηγή: https://spacy.io/usage/processing-pipelines#pipelines
[54] Πηγή: https://spacy.io/usage/processing-pipelines#custom-components

Πίνακας 7.5: Χαρακτηριστικά συχνότητας οντοτήτων

Feature	Περιγραφή
total_cinema	Το σύνολο εμφάνισης λέξεων της οντότητας CINEMA στα tweets του χρήστη
total_music	Το σύνολο εμφάνισης λέξεων της οντότητας MUSIC στα tweets του χρήστη
total_nutrition	Το σύνολο εμφάνισης λέξεων της οντότητας NUTRITION στα tweets του χρήστη
total_politics	Το σύνολο εμφάνισης λέξεων της οντότητας POLITICS στα tweets του χρήστη
total_religion	Το σύνολο εμφάνισης λέξεων της οντότητας RELIGION στα tweets του χρήστη
total_school	Το σύνολο εμφάνισης λέξεων της οντότητας SCHOOL στα tweets του χρήστη
total_sports	Το σύνολο εμφάνισης λέξεων της οντότητας SPORTS στα tweets του χρήστη
total_technology	Το σύνολο εμφάνισης λέξεων της οντότητας TECHNOLOGY στα tweets του χρήστη
total_slang	Το σύνολο εμφάνισης λέξεων της οντότητας SLANG στα tweets του χρήστη

Οι συναρτήσεις που υλοποιούν τις λειτουργικότητες που περιγράφηκαν φαίνονται παρακάτω:

```
# Tokenization, Remove Stop-words and Lemmatization
def getTokens(x):
    token_list = []
    for token in x:
        if (not token.is_stop and len(token)>=2 and not token.pos_ is 'PUNCT'):
            token_list.append(token.lemma_)
    if not token_list:
        return 'no_tokens'
    return ','.join(token_list)

def findSlang(x):
    temp = 0
    for entity_name, entity_label in x:
        if entity_label == 'SLANG':
            temp = temp +1
    return temp

def findSports(x):
    temp = 0
    for entity_name, entity_label in x:
        if entity_label == 'SPORTS':
            temp = temp +1
    return temp

def findCinema(x):
    temp = 0
    for entity_name, entity_label in x:
        if entity_label == 'CINEMA':
            temp = temp +1
    return temp
```

```
def findPolitics(x):
    temp = 0
    for entity_name, entity_label in x:
        if entity_label == 'POLITICS':
            temp = temp +1
    return temp

def findMusic(x):
    temp = 0
    for entity_name, entity_label in x:
        if entity_label == 'MUSIC':
            temp = temp +1
    return temp

def findSchool(x):
    temp = 0
    for entity_name, entity_label in x:
        if entity_label == 'SCHOOL':
            temp = temp +1
    return temp

def findReligion(x):
    temp = 0
    for entity_name, entity_label in x:
        if entity_label == 'RELIGION':
            temp = temp +1
    return temp

def findTechnology(x):
    temp = 0
    for entity_name, entity_label in x:
        if entity_label == 'TECHNOLOGY':
            temp = temp +1
    return temp

def findNutrition(x):
    temp = 0
    for entity_name, entity_label in x:
        if entity_label == 'NUTRITION':
            temp = temp +1
    return temp
```

7.5 Εύρεση θέματος κειμένου των tweets

Η επεξεργασία των δεδομένων συνεχίστηκε με τη διαδικασία εξαγωγής του θέματος των κειμένων (topic modelling) των tweets μέσω ανάλυσής τους με κατάλληλους αλγορίθμους. Τέτοιου είδους αλγόριθμοι αποτελούν ο LDA και ο GuidedLDA που αναλύονται στις ενότητες 3.4.3 και 3.4.4 αντίστοιχα. Οι δύο αλγόριθμοι δοκιμάστηκαν στο αρχείο που περιλάμβανε τα δεδομένα κειμένου με δύο διαφορετικές τεχνικές, την μέθοδο Bag-of-Words (BoW) και τη μέθοδο TF-IDF που περιγράφονται στις ενότητες 3.4.1 και 3.4.2 αντίστοιχα. Ο σκοπός αυτής της μελέτης ήταν ένας πιο λεπτομερής και ακριβής προσδιορισμός του θέματος των tweets ώστε να υπάρξει επιπλέον πληροφορία για τα ενδιαφέροντα των χρηστών.

Η διεργασία εφαρμόστηκε στα tokens που είχαν προκύψει από την NLP μέθοδο που είχε πραγματοποιηθεί στα tweets στο προηγούμενο στάδιο. Αρχικά, δημιουργήθηκε μία μήτρα εμφάνισης των λέξεων των tweets μέσω της συνάρτησης CountVectorizer() που μετατρέπει μία συλλογή λέξεων σε ένα πίνακα με τις εμφανίσεις των tokens. Πρόκειται για μία αραιή απεικόνιση των λέξεων σε έναν πίνακα (sparse matrix) που είχε ως γραμμές τα tweets και ως στήλες όλες τις λέξεις που αναγνώρισε η συνάρτηση CountVectorizer(). Ουσιαστικά οι στήλες αποτελούν ένα λεξικό. Στην παρούσα εργασία ο πίνακας που δημιουργήθηκε είχε 2.402.793 γραμμές και 271.924 στήλες, δηλαδή το λεξικό (vocabulary) περιλάμβανε 271.924 λέξεις. Επίσης κατασκευάστηκε ένα ευρετήριο (dictionary) που είχε ως κλειδιά (keys) τις λέξεις και ως τιμές (values) την συχνότητα εμφάνισής τους. Η εκτέλεση των συγκεκριμένων διεργασιών είναι μία σύντομη διαδικασία και διήρκησε συνολικά περίπου 1 λεπτό. Ο κώδικας που φαίνεται παρακάτω δείχνει την υλοποίηση τους:

```
from sklearn.feature_extraction.text import CountVectorizer
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(tokens)
voc = count_vect.vocabulary_
tf_feature_names = count_vect.get_feature_names()
word2id = dict((v, idx) for idx, v in enumerate(tf_feature_names))
```

Η πρώτη δοκιμή αφορούσε τον LDA αλγόριθμο πάνω στον αραιό πίνακα των λέξεων. Επιλέχθηκαν οκτώ θέματα για την κατηγοριοποίηση των δεδομένων, πεδίο που ορίζεται στην παράμετρο “n_components”. Το μοντέλο αποθηκεύτηκε με τη βοήθεια του πακέτου joblib, που περιγράφεται στην ενότητα 4.1.13. Ο κώδικας που την υλοποιεί είναι ο εξής:

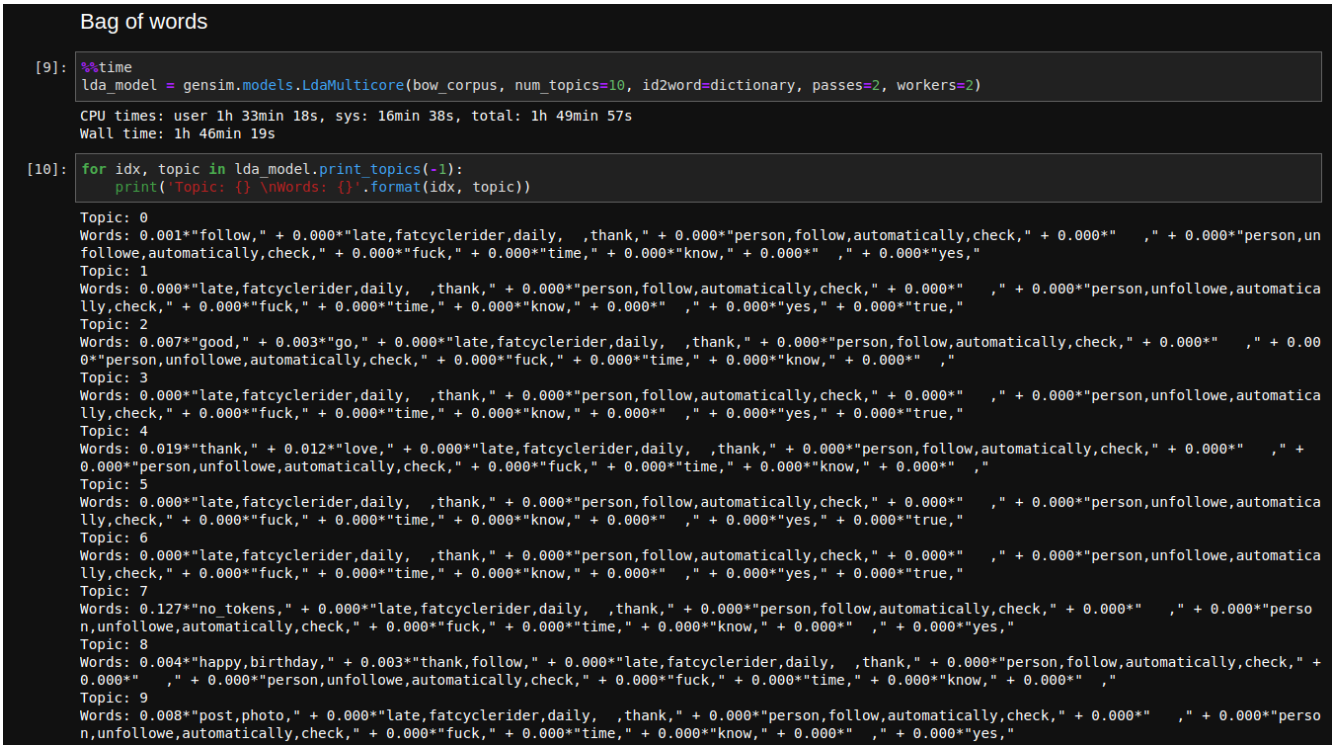
```
from sklearn.decomposition import LatentDirichletAllocation
lda_model = LatentDirichletAllocation(n_components=8,
                                      max_iter=10,
                                      learning_method='online',
                                      random_state=100,
                                      batch_size=128)
lda_model.fit(X_train_counts)
print("Log-Likelihood: ", lda_model.score(X_train_counts))
print("Perplexity: ", lda_model.perplexity(X_train_counts))
joblib.dump(lda_model, "simple_LDA_8_topics.pkl")
```

Η εκμάθηση του μοντέλου ήταν αρκετά απαιτητική σε υπολογιστικούς πόρους και διήρκησε 4 ώρες και 45 λεπτά περίπου. Τα αποτελέσματα δεν ήταν τα επιθυμητά αφού το μοντέλο σημείωσε πολύ μικρή επίδοση για την μετρική log-likelihood με -115.900.711,98 και για την μετρική perplexity με 6.203,13. Συνεπώς δε γίνεται να χρησιμοποιηθεί για το topic modelling των tweets.

Η επεξεργασία συνεχίστηκε με την εφαρμογή του LDA μέσω της προσέγγισης Bag-of-Words. Κατασκευάστηκε μέσω του πακέτου gensim, που περιγράφηκε στην ενότητα 4.1.12, κατάλληλο λεξικό που περιλάμβανε τα tokens. Η μέθοδος αυτή δοκιμάστηκε διαδοχικά για την ομαδοποίηση σε 8, 10 και 25 topics και ορίστηκε μέσω της παραμέτρου “num_topics”. Τα μοντέλα αποθηκεύτηκαν μέσω του πακέτου joblib. Η επιλογή των 8 topics έγινε με βάση το πλήθος των topics (CINEMA, MUSIC, NUTRITION, POLITICS, RELIGION, SCHOOL, SPORTS, TECHNOLOGY) που εξετάστηκαν στο NLP μοντέλο προηγουμένως, ενώ για 10 και 25 ακολουθήθηκαν παρόμοιες προσεγγίσεις. Η διάρκεια εκμάθησης για κάθε περίπτωση ήταν μεγάλη και χρειάστηκε περίπου 1 ώρα και 25 λεπτά για την λύση των 8 topics, 1 ώρα και 46 λεπτά περίπου για τα 10 topics και 2 ώρες και 41 λεπτά για τα 25 topics. Ωστόσο, τα αποτελέσματα για κάθε ομαδοποίηση δεν ήταν τα επιθυμητά. Τα βάρη που έδωσαν τα μοντέλα στις λέξεις-κλειδιά κάθε topic δεν βοηθούσαν στην εξαγωγή κάποιου συμπεράσματος για το

θέμα του. Συνεπώς απορρίφθηκαν οι συγκεκριμένες λύσεις. Ο κώδικας που εφαρμόζει την τεχνική Bag-of-Words φαίνεται παρακάτω, ενώ στην Εικόνα 7.3 παρουσιάζονται ενδεικτικά τα αποτελέσματα του μοντέλου για 10 topics.

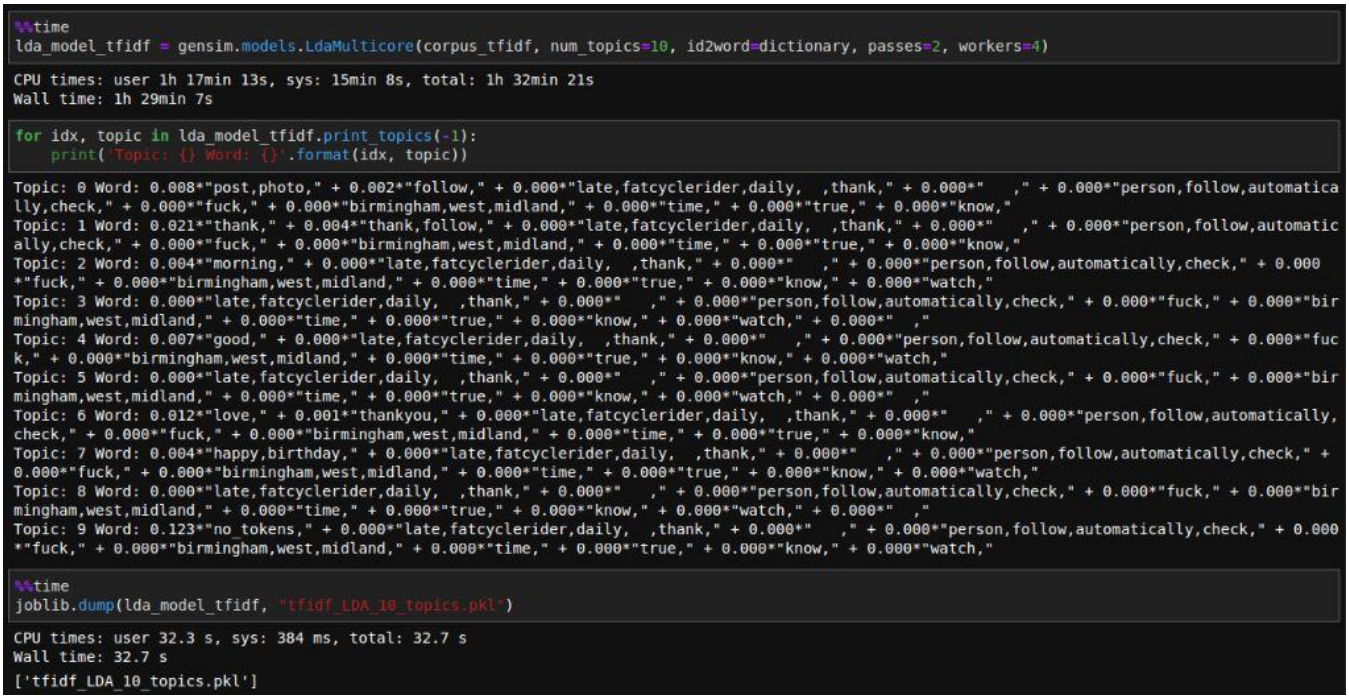
```
import gensim
from gensim import corpora, models
dictionary = gensim.corpora.Dictionary(tokens)
bow_corpus = [dictionary.doc2bow(doc) for doc in tokens]
lda_model = gensim.models.LdaMulticore(bow_corpus, num_topics= n,
                                       id2word=dictionary,
                                       passes=2, workers=2)
for idx, topic in lda_model.print_topics(-1):
    print('Topic: {} \nWords: {}'.format(idx, topic))
joblib.dump(lda_model, "normal_LDA_n_topics.pkl")
```



Εικόνα 7.3: Αποτελέσματα LDA με BoW για 10 Topics

Η επόμενη φάση της επεξεργασίας περιλάμβανε την εξέταση του μοντέλου LDA μέσω της τεχνικής TF-IDF. Αξιοποιήθηκε το λεξικό της μεθόδου BoW το οποίο μοντελοποιήθηκε κατάλληλα ώστε να μελετηθεί με την τεχνική TF-IDF. Όπως και πριν, η μέθοδος δοκιμάστηκε διαδοχικά για την ομαδοποίηση σε 8, 10 και 25 topics και ορίστηκε μέσω της παραμέτρου “num_topics” και τα μοντέλα που προέκυψαν αποθηκεύτηκαν μέσω του πακέτου joblib. Η εκπαίδευση για κάθε περίπτωση διήρκησε αρκετά, αλλά ήταν συντομότερη της μεθόδου BoW. Χρειάστηκε περίπου 1 ώρα και 22 λεπτά για την λύση των 8 topics, 1 ώρα και 29 λεπτά περίπου για τα 10 topics και 1 ώρα και 33 λεπτά για τα 25 topics. Όμως τα αποτελέσματα για κάθε ομαδοποίηση δεν ήταν τα ικανοποιητικά ούτε σε αυτή την προσέγγιση. Τα βάρη που έδωσαν τα νέα μοντέλα στις λέξεις-κλειδιά κάθε topic δεν επέτρεπαν ασφαλή συμπεράσματα για το θέμα του. Συνεπώς απορρίφθηκαν και αυτές οι λύσεις για το topic modelling. Ο κώδικας που χρησιμοποιεί την τεχνική TF-IDF φαίνεται παρακάτω, ενώ στην Εικόνα 7.4 παρουσιάζονται ενδεικτικά τα αποτελέσματα του μοντέλου για 10 topics.

```
import gensim
from gensim import corpora, models
dictionary = gensim.corpora.Dictionary(tokens)
bow_corpus = [dictionary.doc2bow(doc) for doc in tokens]
tfidf = models.TfidfModel(bow_corpus)
corpus_tfidf = tfidf[bow_corpus]
lda_model_tfidf = gensim.models.LdaMulticore(corpus_tfidf,
                                             num_topics=n,
                                             id2word=dictionary,
                                             passes=2, workers=4)
for idx, topic in lda_model_tfidf.print_topics(-1): print('Topic:
{} Word: {}'.format(idx, topic))
joblib.dump(lda_model_tfidf, "tfidf_LDA_n_topics.pkl")
```



Εικόνα 7.4: Αποτελέσματα LDA με BoW για 10 Topics

Η προσπάθεια εξαγωγής του θέματος των tweets ολοκληρώθηκε με τη χρήση του μοντέλου GuidedLDA. Όπως επισημάνθηκε στην ενότητα 3.4.4, πρόκειται για έναν αλγόριθμο που τείνει να μετατρέψει το topic modelling σε πρόβλημα ημι-επιβλεπόμενης μάθησης, σε αντιθεση με τον LDA που είναι μέθοδος μη επιβλεπόμενη μάθησης. Αυτό συμβαίνει επειδή ακολουθεί μία καινοτόμα προσέγγιση και έτσι στον GuidedLDA τα topics ορίζονται ακολουθούμενα από συγκεκριμένες λέξεις-κλειδιά που τα προσδιορίζουν εννοιολογικά.

Η επίλυση του προβλήματος ξεκίνησε από την κατασκευή του λεξικού και του αραιού πίνακα όπως περιγράφηκε στην αρχή. Σε αυτή τη λύση όμως το επόμενο βήμα ήταν η δημιουργία μίας λίστας από λίστες λέξεων χαρακτηριστικές για κάθε topic που επιδιώκει να προβλέψει το μοντέλο. Στην προκειμένη περίπτωση επιλέχθηκαν τα 8 γνωστά topics (CINEMA, MUSIC, NUTRITION, POLITICS, RELIGION, SCHOOL, SPORTS, TECHNOLOGY) που μελετήθηκαν και σε προηγούμενα πειραματικά στάδια. Η λίστα για κάθε topic συμπληρώθηκε μέσω των txt αρχείων που είχαν χρησιμοποιηθεί στην μέθοδο NER προηγουμένως και περιείχαν λέξεις κλειδιά οι οποίες το χαρακτήριζαν. Για την εκμάθηση του αλγορίθμου φορτώθηκε αρχικά ένα instance του μοντέλου με τις πιο ευρέως χρησιμοποιούμενες

υπερπαραμέτρους στη βιβλιογραφία. Έπειτα η εκμάθηση έγινε πάνω στον αραιό πίνακα των δεδομένων και στη λίστα με τα topics και τις λέξεις-κλειδιά επιλέγοντας τιμή 0,15 για την παράμετρο "seed_confidence", που ορίζει την επιπλέον ενίσχυση (extra boost) κάθε ενδεικτικής λέξης. Το μοντέλο που προέκυψε αποθηκεύτηκε μέσω του joblib πακέτου. Η συνολική διάρκεια εκτέλεση ήταν 5 ώρες και 54 λεπτά περίπου. Ο κώδικας που υλοποιεί τη διαδικασία φαίνεται παρακάτω:

```
import guidedlda
# Fill topic list
seed_topic_list = []
seed_topic_list.append(cinema_keywords)
seed_topic_list.append(music_keywords)
seed_topic_list.append(nutrition_keywords)
seed_topic_list.append(politics_keywords)
seed_topic_list.append(religion_keywords)
seed_topic_list.append(school_keywords)
seed_topic_list.append(sports_keywords)
seed_topic_list.append(technology_keywords)
#Instantiate the guidedlda
model = guidedlda.GuidedLDA(n_topics=8, n_iter=100,
                            random_state=7, refresh=10)
#seed_topics is the dictionary {word_id to topic_id}
seed_topics = {}
for t_id, st in enumerate(seed_topic_list):
    for word in st:
        if word.lower() in count_vect.vocabulary_:
            seed_topics[word2id[word]] = t_id
        else:
            continue
# Train model
model.fit(X_train_counts, seed_topics=seed_topics,
          seed_confidence=0.15)
joblib.dump(model, 'Guided_LDA_8_topics.pkl')
```

Με την ολοκλήρωση της εκτέλεσης διαπιστώνεται εύκολα πως το μοντέλο σημείωσε πολύ μικρή επίδοση για την μετρική log-likelihood παρατηρώντας τις τιμές που φαίνονται στην Εικόνα 7.5 για κάθε επανάληψη.

```
INFO:guidedlda:n_topics: 8
INFO:guidedlda:n_iter: 100
INFO:guidedlda:<0> log likelihood: -413250949
INFO:guidedlda:<10> log likelihood: -37944615
INFO:guidedlda:<20> log likelihood: -35248248
INFO:guidedlda:<30> log likelihood: -33334417
INFO:guidedlda:<40> log likelihood: -31868163
INFO:guidedlda:<50> log likelihood: -30773843
INFO:guidedlda:<60> log likelihood: -29817566
INFO:guidedlda:<70> log likelihood: -29152731
INFO:guidedlda:<80> log likelihood: -28524070
INFO:guidedlda:<90> log likelihood: -28031487
INFO:guidedlda:<99> log likelihood: -27605826
```

Εικόνα 7.5: Αποτελέσματα GuidedLDA για 8 topics

Ακόμη, επιλέχθηκαν οι 25 πιο χαρακτηριστικές λέξεις για κάθε topic ώστε να γίνει επαλήθευση και ταυτοποίηση των topics. Ωστόσο, όπως φαίνεται στην Εικόνα 7.6 τα αποτελέσματα δεν ήταν ικανοποιητικά, διότι περιείχαν πολλές ίδιες λέξεις , όπως “like”, “go”, “lool”, “day”, “love” και άλλες, κάνοντας αδύνατη την ταυτοποίηση τους. Μόλις 2 από τα 8 topics μπόρεσαν να αναγνωριστούν και αυτό έγινε για ελάχιστες λέξεις. Μόνο το topic 3, το οποίο αναφέρεται στη θεματική ενότητα SPORTS, ξεχώρισε σημαντικά και ήταν εύκολη η ταυτοποίηση του, αφού περιλάμβανε σχετικές λέξεις όπως “goal”, “win”, “football”, “player” και “league”. Επίσης, το topic 5, σχετικό με το θέμα TECHNOLOGY, μπόρεσε να προσδιοριστεί μέσω των 4 λέξεων “follow”, “tweet”, “post” και “follower”. Ο κώδικας και τα αποτελέσματα για τα topic φαίνονται παρακάτω.

```
n_top_words = 25
topic_word = model.topic_word_
for i, topic_dist in enumerate(topic_word):
    topic_words = np.array(tf_feature_names)[np.argsort(topic_dist)][:-(n_top_words+1):-1]
    print('Topic {}: {}'.format(i, ' '.join(topic_words)))
```

```
Topic 0: like go day get good love work time night need eat think feel want look today know come bed fuck drink sleep haha home amp
Topic 1: good love like day go get time think know look today year come new thank night happy work need people watch want amp tonight wait
Topic 2: go like good day get love know think time today people no_tokens look work need year oh life feel thank fuck want night happy come
Topic 3: win game goal good play fan team league season player go come year time get today score man great like football united city arsenal think
Topic 4: watch good love like day go get time new night look think come today work know tonight year thank amp wait happy video need want
Topic 5: follow new like thank love people amp no_tokens day rt get good today know check go look win follower think tweet time want need post
Topic 6: people like think today get time year vote go good know day amp say want need look come work new man right love great thing
Topic 7: day go get good today like time year work think know come look people new love need amp week want thank great night watch start
```

Εικόνα 7.6: Οι 25 πιο χαρακτηριστικές λέξεις για κάθε topic

Επιπλέον έγινε προσπάθεια να κατηγοριοποιηθούν όλα τα tweets του συνόλου δεδομένων, λαμβάνοντας την αντίστοιχη ετικέτα του topic που ανήκουν. Επιλέχθηκε ως όριο ακρίβειας η τιμή 0.5, δηλαδή ένα tweet για να χαρακτηριστεί ως προς το θέμα του πρέπει να παρουσιάζει πιθανότητα τουλάχιστον 50% να ανήκει στο αντίστοιχο topic. Αυτός ο κανόνας είχε ως αποτέλεσμα το 20% των tweets να μην κατηγοριοποιηθεί σε κάποια θεματική ομάδα. Πρόκειται για μεγάλο ποσοστό αποτυχίας αν θεωρήσουμε ότι επιλέχθηκε ως όριο για την πιθανότητα το 0,5 και έτσι δεν μπορεί να προσφέρει ασφαλή συμπεράσματα για την των tweets. Συνεπώς, ο Guided LDA απέτυχε να βοηθήσει στο topic modelling των tweets και έτσι τα αποτελέσματά του δε θα χρησιμοποιηθούν στα επόμενα στάδια των πειραμάτων για την ανίχνευση της ηλικίας των χρηστών.

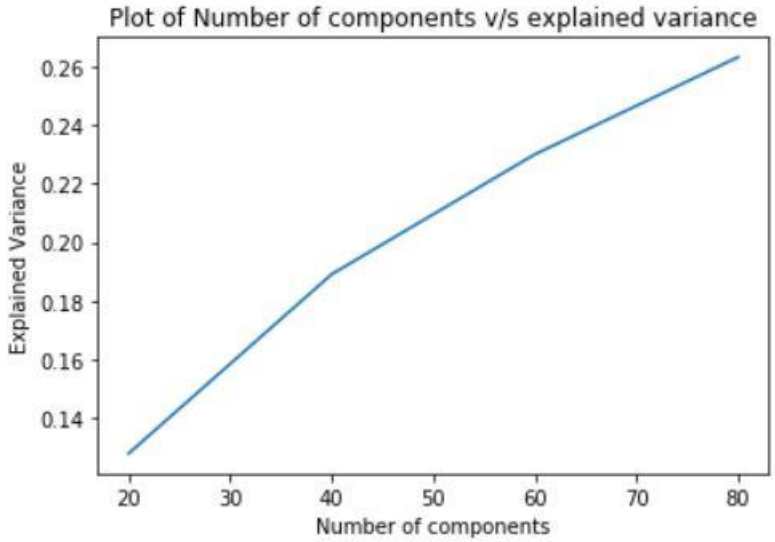
Γενικά, παρατηρήθηκε ότι όλες οι προσπάθειες και προσεγγίσεις που δοκιμάστηκαν για την υλοποίηση του topic modelling των tweets απέτυχαν σε μεγάλο βαθμό και δεν σημείωσαν ικανοποιητική ακρίβεια στις προβλέψεις τους. Για αυτό το λόγο απορρίφθηκαν και δε θα αποτελέσουν πηγή πληροφορίας στη συνέχεια της παρούσας εργασίας για την πρόβλεψη της ηλικίας των χρηστών.

7.6 Εξαγωγή γλωσσολογικών ιδιοτήτων

Η προσπάθεια εξαγωγής χρήσιμων στοιχείων από τα γλωσσολογικά δεδομένα των tweets συνεχίστηκε με εκ νέου επεξεργασία τους. Όπως είναι γνωστό οι αλγόριθμοι μηχανικής μάθησης δεν μπορούν να λάβουν δεδομένα κειμένου ώστε να εκπαιδευτούν αλλά μόνο αριθμούς. Αυτή η ιδιαιτερότητα γέννησε την ιδέα παραγωγής αριθμητικών δεδομένων από τα κείμενα των tweets. Τα tokens αποτέλεσαν τα δεδομένα εισόδου για αυτό το στάδιο. Ο αραιός πίνακας που παράχθηκε μέσω της συνάρτησης CountVectorizer() είχε 2.402.793 γραμμές, μία για κάθε tweet, και 271.924 στήλες, όπως στο προηγούμενο βήμα και ήταν δύσκολα διαχειρίσιμος λόγω των πολλών διαστάσεων του. Σκοπός ήταν να μειωθεί ο μεγάλος αριθμός των στηλών ώστε να δημιουργηθεί ένα σύνολο δεδομένων με λίγες διαστάσεις οι οποίες όμως θα αποτελούν ουσιαστική αναπαράσταση του αρχικού. Αυτό υλοποιήθηκε μέσω της τεχνικής

μείωσης διαστάσεων που περιγράφηκε στην ενότητα 3.4.5 με τη χρήση του αλγορίθμου TruncatedSVD. Οι πιθανές τιμές από τις οποίες θα επιλεγόταν το πλήθος των νέων διαστάσεων ήταν οι 20, 40, 60 και 80 όπου προσδιορίστηκαν με βάση την επεξεργαστική ικανότητα του μηχανήματος. Αποφασίστηκε να προκριθεί αυτή που θα παρουσίαζε τον καλύτερο συντελεστή προσδιορισμού. Ο κώδικας που υλοποιεί αυτή τη διαδικασία καθώς και το σχεδιασμό του διαγράμματος της εικόναςΕικόνα *7.7* φαίνεται παρακάτω.

```
components_list = [20,40,60,80]
svd_list = {}
explained = [] # explained variance ratio for each component of Truncated SVD
for i in components_list:
    print("NEW ITERATION")
    X_counts = count_vect.fit_transform(X)
    print(X_counts.shape)
    svd = TruncatedSVD(n_components=i, n_iter=7, random_state=42).fit(X_counts)
    X_counts = svd.transform(X_counts)
    print(X_counts.shape)
    svd_list[i] = X_counts
    explained.append(svd.explained_variance_ratio_.sum())
    print("Number of components = %r and explained variance = %r"%(i,
                                        svd.explained_variance_ratio_.sum()))
plt.plot(components_list, explained)
plt.xlabel('Number of components')
plt.ylabel("Explained Variance")
plt.title("Plot of Number of components v/s explained variance")
plt.show()
```



Εικόνα 7.7: Διάγραμμα μεταβολής συντελεστή προσδιορισμού ανά αριθμό συνιστωσών

Η εκτέλεση της διεργασίας ήταν σύντομη και διήρκησε περίπου 7 λεπτά. Μελετώντας το διάγραμμα που προέκυψε και φαίνεται στην Εικόνα 7.7 εξάγεται εύκολα το συμπέρασμα πως ο αριθμός 80 για το νέο πλήθος των διαστάσεων αποτελεί την πιο αντιπροσωπευτική λύση. Έτσι επιλέχθηκε για τη συνέχεια της μελέτης ο αντίστοιχος πίνακας για 80 συνιστώσες που παράχθηκε από το μοντέλο και περιλάμβανε τις ζητούμενες αριθμητικές τιμές για τα γλωσσολογικά δεδομένα. Ο πίνακας μετατράπηκε σε ένα νέο data frame με 80 features, όπου ένα στιγμιότυπό του φαίνεται στην Εικόνα 7.8, και συνενώθηκε με τον πίνακα εισόδου με κλειδί το index. Με αυτόν τον τρόπο, κάθε γραμμή του νέου dataset που αντιστοιχούσε σε ένα μεμονωμένο tweet, περιλάμβανε το κείμενο του tweet, τα αριθμητικά του αποτελέσματα καθώς

και το πεδίο “aged_user_id” για την ταυτοποίηση του χρήστη που το δημοσίευσε. Το παρόν σύνολο δεδομένων αποθηκεύτηκε σε νέο αρχείο CSV.

	f0	f1	f2	f3	f4	f5	f6	f7	f8	f9	...
0	0.070176	-0.030015	0.008759	-0.005484	0.011730	-0.004979	0.038772	0.040822	0.060590	0.039388	...
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...
2	0.050356	-0.089195	-0.026249	0.007337	0.031918	0.018509	0.104222	0.038760	0.115670	-0.006744	...
3	0.001650	-0.001629	-0.000088	-0.001141	0.001296	-0.000895	-0.000231	0.000170	0.000176	0.000240	...
4	0.028264	-0.041273	-0.003893	-0.007976	0.017284	0.017460	0.043015	0.026073	0.073937	-0.006138	...
...	...	...	...	...	...	...	...	...	...	...	...

Εικόνα 7.8: Στιγμιότυπο data frame αριθμητικών τιμών για τα tweets

7.7 Το σύνολο δεδομένων (dataset)

Το τελικό σύνολο δεδομένων που αποτέλεσε την είσοδο για τους αλγορίθμους μηχανικής μάθησης που δοκιμάστηκαν κατασκευάστηκε συγκεντρώνοντας όλα τα σύνολα δεδομένων από τα διαφορετικά στάδια επεξεργασίας που περιγράφηκαν στις ανωτέρω ενότητες σε ένα κοινό αρχείο CSV. Κάθε ξεχωριστό αρχείο ομαδοποιήθηκε έχοντας ως κλειδί το πεδίο “aged_user_id” συγκεντρώνοντας έτσι τα στοιχεία για κάθε χρήστη σε μία γραμμή. Άρα το αρχείο από το στάδιο της προετοιμασίας των δεδομένων που περιλάμβανε τα στατιστικά όπως περιγράφηκε στην ενότητα 7.3, τα δύο σύνολα δεδομένων κειμένου από την NLP διεργασία που αναλύθηκε στην ενότητα 7.4 και το αρχείο από την εξαγωγή αριθμητικών δεδομένων για τα tweets, που παρουσιάστηκε στην ενότητα 7.6 τέθηκαν σε αυτή τη διαδικασία. Η ομαδοποίηση των δεδομένων ανά χρήστη μέσω του στοιχείου “aged_user_id” εξυπηρέτησε την τεχνική επαύξησης των χρηστών και πλέον υπήρχαν 5477 χρήστες.

Μέσω της συγκρότησης των δεδομένων ανά χρήστη, κάθε γραμμή περιείχε σε μορφή λίστας τις εξαγόμενες πληροφορίες όλων των tweets του και έτσι χρειάστηκε ο υπολογισμός των συνολικών τιμών για κάθε χαρακτηριστικό εφαρμόζοντας την απλή συνάρτηση sum() για τις λίστες. Αυτό το στάδιο διαφοροποιήθηκε για τις αριθμητικές τιμές των κειμένων με σκοπό να διατηρηθεί η περιγραφικότητα και οι ιδιότητες τους. Έτσι για καθένα από τα 80 features βρέθηκαν δύο νέα. Πιο συγκεκριμένα αυτά ήταν ο μέσος όρος και η διασπορά των τιμών που περιείχαν οι λίστες με τη χρήση των γνωστών συναρτήσεων mean() και stdev(). Τα παλιά χαρακτηριστικά, όπως φαίνονται στην Εικόνα 7.8, αντικαταστάθηκαν και για κάθε ένα f_i από αυτά τοποθετήθηκαν τα avg_f_i και std_f_i συνθέτοντας ένα νέο σύνολο δεδομένων με 160 features που αξιοποιήθηκε στη συνέχεια.

Τελικά, τα ομαδοποιημένα ανά χρήστη σύνολα δεδομένων συγχωνεύθηκαν με κλειδί το πεδίο “aged_user_id” και όλα τα δεδομένα βρέθηκαν σε ένα κοινό τελικό αρχείο CSV, έτοιμο για να αποτελέσει την είσοδο για την εκπαίδευση των αλγορίθμων μηχανικής μάθησης.

Αυτή η συνένωση είχε ως αποτέλεσμα το τελικό αρχείο να περιλαμβάνει 5477 γραμμές, δηλαδή τους χρήστες, και 187 στήλες, δηλαδή τα χαρακτηριστικά τους. Πρόκειται για τα 12 χαρακτηριστικά που προήλθαν από την επεξεργασία των στοιχείων του προφίλ του χρήστη, τα 15 features που δημιουργήθηκαν από την διαδικασία του NLP και τα 160 που προέκυψαν από τα αποτελέσματα της μαθηματικής μελέτης των γλωσσολογικών ιδιοτήτων των χρηστών. Το τελικό dataset περιλαμβάνει τα εξής features:

Πίνακας 7.6: Τα χαρακτηριστικά του dataset

Στήλες	Χαρακτηριστικά	Περιγραφή
1	account_age	Αφορά το χρονικό διάστημα εκφρασμένο σε έτη που ο χρήστης διαθέτει λογαριασμό στο Twitter
2	morning_tweets	Αφορά το πλήθος των tweets που δημοσίευσε ο χρήστης τις πρωινές ώρες από τις 08:00 έως τις 15:59
3	evening_tweets	Αφορά το πλήθος των tweets που δημοσίευσε ο χρήστης τις πρωινές ώρες από τις 16:00 έως τις 23:59
4	night_tweets	Αφορά το πλήθος των tweets που δημοσίευσε ο χρήστης τις πρωινές ώρες από τις 00:00 έως τις 07:59
5	total_quoted_status	Αφορά το πλήθος των tweets του χρήστη που ήταν quoted
6	total_count_retweet	Αφορά το συνολικό πλήθος των retweets (@RT) που έχει δημοσιεύσει ο χρήστης
7	total_emoji	Αφορά το συνολικό πλήθος των emoji (emoticons) που έχει χρησιμοποιήσει ο χρήστης στα tweets του
8	total_favorite_count	Αφορά το πλήθος των likes που έλαβαν συνολικά τα tweets του χρήστη
9	total_hashtags	Αφορά το συνολικό πλήθος των hashtags (#) που έχει χρησιμοποιήσει ο χρήστης στα tweets του
10	total_len_text	Αφορά το συνολικό μήκος σε χαρακτήρες όλων των tweets που έχει δημοσιεύσει ο χρήστης
11	total_music	Αφορά το σύνολο εμφάνισης λέξεων της οντότητας MUSIC στα tweets του χρήστη
12	total_nutrition	Αφορά το σύνολο εμφάνισης λέξεων της οντότητας NUTRITION στα tweets του χρήστη
13	total_politics	Αφορά το σύνολο εμφάνισης λέξεων της οντότητας POLITICS στα tweets του χρήστη
14	total_cinema	Αφορά το σύνολο εμφάνισης λέξεων της οντότητας CINEMA στα tweets του χρήστη
15	total_religion	Αφορά το σύνολο εμφάνισης λέξεων της οντότητας RELIGION στα tweets του χρήστη
16	total_school	Αφορά το σύνολο εμφάνισης λέξεων της οντότητας SCHOOL στα tweets του χρήστη
17	total_slang	Αφορά το σύνολο εμφάνισης λέξεων της οντότητας SLANG στα tweets του χρήστη
18	total_sports	Αφορά το σύνολο εμφάνισης λέξεων της οντότητας SPORTS στα tweets του χρήστη
19	total_technology	Αφορά το σύνολο εμφάνισης λέξεων της οντότητας TECHNOLOGY στα tweets του χρήστη
20	total_retweeted_count	Αφορά το πλήθος των tweets του χρήστη που έχουν γίνει retweet από άλλους πιστοποιημένους χρήστες
21	total_tags	Αφορά το συνολικό πλήθος των tags-mentions (@) που έχει κάνει ο χρήστης στα tweets του
22	total_URLs	Αφορά το συνολικό πλήθος των URLs (σύνδεσμοι, links) που έχει δημοσιεύσει ο χρήστης μέσω των tweets του
23	user_followers_count	Αφορά το πλήθος των followers που έχει ο χρήστης
24	user_friends_count	Αφορά το πλήθος των followings, δηλαδή των λογαριασμών που ακολουθεί ο χρήστης
25	user_favourites_count	Αφορά το πλήθος των tweets που ο χρήστης έχει κάνει like
26	user_listed_count	Αφορά το πλήθος των δημόσιων ομάδων όπου ο χρήστης είναι μέλος
27	user_statuses_count	Αφορά το πλήθος των tweets που έχει δημοσιεύσει ο χρήστης συμπεριλαμβανόμενων των retweets
28 έως 187	avg_f_i και std_f_i	Αφορούν κάθε ένα από τα 160 γλωσσολογικά features

Στην Εικόνα 7.9 παρακάτω απεικονίζεται ένα στιγμιότυπο του dataset:

account_age	age	aged_user_id	evening_tweets	morning_tweets	night_tweets	total_cinema	total_count_retweet	total_emoji	total_favorite_count	total_hashtags	total_len_text	total_music	total_nutrition	total_politics	total_quoted_status	total_religion	total_retweeted_count	total_school	total_slang	total_sports	total_tags	total_technology	user_favourites_count	user_followers_count	user_friends_count	user_id	user_listed_count	user_statuses_count	avg_f0	std_f0	avg_f1	std_f1	avg_f2	std_f2
1	12	11443652_12	2	8	1	3	0	0	8	0	891	0	1	0	0	0	0	0	4	3	0	2	118	613	240	11443652	9	11	41.404.470.450.721.000	4.176.160.707.273.330	-15.408.571.290.233.200	3.601.847.170.068.900	-7.097.189.500.693.380	11.511.106.159.224.000
2	13	11443652_13	319	132	28	28	0	3	0	0	40086	0	3	0	0	0	0	12	2	452	0	4	120	624	244	11443652	9	478	46.809.370.590.716.300	8.803.067.494.809.560	-47.641.388.640.385.500	6.960.735.020.367.250	-15.000.472.305.033.700	2.898.050.272.441.820
3	14	11443652_14	27	5	4	3	0	1	0	0	1466	0	7	0	0	0	0	12	1	1	1	9	122	635	249	11443652	9	32	9.565.476.325.467.460	18.444.142.109.583.400	-8.723.792.343.939.540	14.536.072.059.446.900	-14.723.923.621.636.400	3.903.143.428.233.020
4	15	11443652_15	85	44	23	25	1	11	0	1	6911	2	10	2	0	1	1	26	6	27	1	7	124	646	253	11443652	9	149	10.490.325.787.764.200	1.420.977.879.804.030	-13.351.705.734.639.700	18.113.797.604.565.100	-13.401.091.090.864.100	44.608.357.080.721.700
5	16	11443652_16	55	30	13	33	2	12	2	2	6241	3	6	1	0	2	4	6	6	16	22	30	127	657	257	11443652	9	97	9.916.551.924.867.300	14.455.521.992.163.400	-11.969.979.812.735.700	1.745.770.847.772.800	-15.465.323.173.794.800	14.097.568.788.475.800
6	17	11443652_17	375	218	65	83	73	144	12	159	45990	14	28	7	0	4	25614	48	36	80	253	167	129	668	262	11443652	10	633	12.210.209.256.789.200	21.815.477.849.151.500	-11.326.962.447.167.300	18.526.319.908.328.900	-9.240.051.641.418.290	1.928.758.741.839.110
7	18	11443652_18	71	49	5	9	8	15	7	18	9155	1	9	1	0	2	3327	2	11	8	30	20	131	679	266	11443652	10	117	10.424.158.652.731.100	17.447.295.068.156.500	-10.639.673.618.611.700	15.650.450.732.250.900	-2.871.500.814.676.460	20.324.707.341.128.400
8	19	11443652_19	117	65	2	34	3	8	30	153	10373	4	5	1	0	1	17	6	2	16	34	12	133	690	270	11443652	10	148	784.136.245.255.055	15.662.542.270.162.600	-880.985.520.952.101	14.494.012.279.477.700	-20.622.046.010.868.600	1.342.336.440.054.220
9	20	11443652_20	156	121	30	45	2	6	119	551	24844	10	13	0	0	6	15	13	12	23	118	51	135	701	275	11443652	10	268	8.537.367.746.073.260	17.195.497.507.273.200	-7.834.541.330.694.840	13.602.255.026.081.000	-31.674.285.987.532.500	11.945.979.065.340.700
10	21	11443652_21	14	5	7	2	0	1	18	35	2059	0	4	0	0	0	2	2	1	4	12	5	137	712	279	11443652	10	22	1.745.711.920.051.160	23.325.315.690.205.800	-12.595.293.002.505.200	25.047.166.094.261.500	41.020.990.838.147.200	3.417.498.221.071.030
11	22	11443652_22	4	2	0	1	0	0	3	11	400	0	0	0	0	0	0	0	0	0	0	1	139	723	283	11443652	10	5	22.159.729.753.281.700	36.780.022.493.004.800	19.840.317.230.677.300	28.536.267.926.714.800	14.947.091.532.803.000	3.695.004.952.056.950
12	23	11443652_23	1	0	0	0	0	0	0	1	140	0	0	0	0	0	0	0	0	2	0	0	142	735	288	11443652	11	1	3.552.895.319.723.510	0.0	-4.239.745.188.126.210	0.0	-8.244.569.702.052.240	0.0
1	34	12721322_34	206	212	36	18	8	35	8	6	31022	16	30	3	0	4	7598	12	2	29	94	55	281	329	645	12721322	7	443	4.857.857.280.889.700	1.429.543.646.288.900	-4.012.801.797.132.750	11.675.211.500.264.400	-52.715.892.372.257.700	10.739.291.043.043.100
2	35	12721322_35	294	203	60	11	17	16	8	111	37035	17	16	4	0	8	7531	14	8	27	54	68	285	334	656	12721322	7	534	4.842.863.033.724.400	14.880.743.277.192.400	-3.409.933.363.353.360	11.361.369.952.045.200	66.957.380.396.097.300	12.922.030.387.936.900
3	36	12721322_36	90	58	5	9	3	13	7	18	9745	5	15	0	0	1	176	2	5	25	24	6	290	340	666	12721322	7	122	10.002.115.256.029.200	2.156.754.617.207.780	-60.064.697.936.077.200	16.031.236.341.325.700	15.038.637.616.660.500	16.891.022.505.928.900
4	37	12721322_37	242	140	47	10	0	13	18	4	23268	10	14	12	0	4	3	6	7	35	17	26	295	345	677	12721322	7	362	6.646.193.870.281.920	1.886.310.927.256.870	-3.873.464.137.939.220	12.045.955.278.819.500	-18.257.050.046.037.800	8.617.045.534.460.010
5	38	12721322_38	257	85	88	23	9	11	32	6	20983	14	9	14	0	8	7101	7	8	67	17	9	299	351	688	12721322	7	277	1.188.407.729.673.830	2.343.990.619.411.150	-8.159.320.875.758.630	17.405.028.160.053.900	-11.023.129.022.491.700	16.084.362.764.955.500
6	39	12721322_39	279	185	66	11	17	30	51	8	27811	11	26	13	3	13	46881	7	6	43	30	27	304	356	699	12721322	7	379	103.018.935.984.125	27.831.627.284.468.500	-54.206.463.294.653.600	19.600.352.853.190.200	-22.234.015.528.102.500	14.753.478.584.387.000
7	40	12721322_40	203	135	42	20	2	23	92	1	22922	23	12	10	0	9	27	5	6	36	20	9	309	362	710	12721322	7	297	1.285.808.224.623.470	245.003.283.294.637	-5.826.176.513.972.230	177.354.619.422.408	-8.741.360.883.549.780	12.333.961.625.690.100
8	41	12721322_41	103	103	78	14	0	40	21	4	16045	15	12	8	1	5	2	6	8	35	5	9	314	368	721	12721322	8	213	9.732.581.611.166.460	2.087.892.052.609.080	-5.638.957.157.434.110	16.572.286.094.254.700	-27.742.852.879.318.400	16.086.935.450.736.100
1	23	13993372_23	107	79	8	21	39	3	1	41	18064	3	21	2	0	5	1	5	19	50	86	37	673	337	469	13993372	16	193	29.478.553.243.617.200	42.810.298.798.296.200	-9.142.070.139.134.100	268.729.695.791.505	24.536.866.476.405.500	22.975.541.592.733.900
2	24	13993372_24	73	105	6	17	86	2	1	45	17177	9	14	2	0	7	2291	2	15	77	72	46	685	343	477	13993372	16	178	17.915.824.368.701.500	279.138.923.059.786	-11.069.966.552.047.300	23.487.574.140.677.400	2.169.186.092.193.080	24.472.092.654.586.600
3	25	13993372_25	114	114	4	12	111	6	1	71	22123	5	21	2	0	12	30201	10	10	57	118	28	696	349	485	13993372	16	227	19.074.723.886.278.500	2.979.118.860.342.710	-10.272.225.571.410.200	22.622.941.369.451.600	-2.195.700.050.500.990	20.739.632.263.769.800
4	26	13993372_26	21	28	2	4	26	1	0	18	5234	1	2	0	0	3	4833	1	8	28	22	5	708	355	494	13993372	17	50	1.456.754.383.829.250	2.010.341.040.917.230	-16.049.591.569.006.400	24.457.275.163.159.100	-6.930.249.016.775.080	16.492.233.493.852.000
5	27	13993372_27	4	40	0	1	12	1	4	9	4039	0	2	0	0	1	15743	1	4	5	52	4	720	361	502	13993372	17	41	22.159.383.750.111.700	2.957.583.296.590.130	-10.999.256.838.439.400	2.609.943.516.472.630	16.475.710.357.323.800	20.651.835.527.308.600
6	28	13993372_28	52	31	2	13	34	7	22	26	7575	2	4	0	0	0	7811	0	4	27	41	12	732	367	510	13993372	17	81	19.127.156.663.140.300	30.635.904.652.651.500	-14.476.370.281.659.200	2.660.714.968.303.460	24.366.299.575.682.600	3.326.652.511.198.970
7	29	13993372_29	48	46	5	5	30	10	8	57	9122	1	9	0	1	1	32063	1	7	24	49	4	744	373	518	13993372	17	98	12.240.221.466.231.400	2.097.164.744.732.450	-12.133.869.944.759.200	20.847.830.370.221.800	10.886.207.197.219.200	23.661.627.915.727.300
8	30	13993372_30	11	11	0	0	14	0	1	12	2038	2	2	0	1	0	221180	0	0	6	4	2	756	379	527	13993372	18	20	1.154.852.500.443.640	9.233.089.581.174.430	-18.625.797.111.903.100	155.379.712.311.772	31.734.255.777.027.500	1.870.258.834.569.000
9	31	13993372_31	7	1	0	0	5	1	1	2	653	0	1	2	0	0	17648	0	0	1	1	1	768	385	535	13993372	18	6	920.391.441.540.111	947.746.940.544.631	-14.491.683.951.234.200	1.672.657.390.667.230	1.366.884.608.550.830	31.216.327.141.963.400

Εικόνα 7.9: Στιγμιότυπο του dataset

Κεφάλαιο 8

8 Πειραματική μελέτη

Στο παρόν κεφάλαιο, θα γίνει χρήση των αλγορίθμων μηχανικής μάθησης στο dataset που περιγράφηκε στην ενότητα 7.7 με τη γλώσσα προγραμματισμού Python, όπως αυτοί παρουσιάστηκαν στην ενότητα 3.2 για την εφαρμογή της παλινδρόμησης και στην ενότητα 3.3 για την εφαρμογή της ταξινόμησης ώστε να πραγματοποιήσουν τις εκτιμήσεις τους για την ηλικία των χρηστών του Twitter. Για την εύρεση του βέλτιστου συνδυασμού των παραμέτρων για την εκτέλεση του εκάστοτε αλγορίθμου, όπως αναλύεται στην ενότητα 3.5.1, πραγματοποιείται τυχαία αναζήτηση (randomized search) με 5-fold cross-validation μέσω της μεθόδου RandomizedSearchCV.

Επίσης, με τον αλγόριθμο ταξινόμησης τυχαίων δέντρων (Extremely Randomized Trees Classifier ή Extra Trees Classifier) έγινε ο υπολογισμός της σπουδαιότητας των features αναδεικνύοντας τα 25 πιο σημαντικά για κάθε προσέγγιση, τα αποτελέσματα του οποίου παρουσιάζονται στην ενότητα 8.2 για την παλινδρόμηση και στην ενότητα 8.3 για την ταξινόμηση. Για καθέναν από τους αλγορίθμους δείχνονται οι κώδικες που τους υλοποιούν, παρουσιάζονται συγκριτικά οι επιδόσεις και τα σφάλματα που εμφανίζουν καθώς και οι τελικές προβλεπόμενες τιμές της ηλικίας συγκρινόμενες με τις πραγματικές με τη μορφή γραφήματος.

8.1 Κανονικοποίηση Δεδομένων Εισόδου

Εξαιτίας της διαφορετικότητας των τιμών που μπορεί να εμφανίσουν τα διάφορα χαρακτηριστικά των δεδομένων εισόδου κρίνεται απαραίτητη η συνολική κανονικοποίησή τους σε ένα συγκεκριμένο εύρος τιμών. Αυτό συμβαίνει γιατί η διαφορά των μεγεθών των τιμών μπορεί να οδηγήσει και έτσι να μειωθεί η απόδοση του μοντέλου. Στην παρούσα εργασία εφαρμόστηκε ο MinMaxScaler που κανονικοποιεί τα δεδομένα με εύρος τιμών μεταξύ του 0 και του 1. Ο κώδικας που τον εφαρμόζει φαίνεται παρακάτω:

```
from sklearn.preprocessing import MinMaxScaler
min_max_scaler = preprocessing.MinMaxScaler()
X = min_max_scaler.fit_transform(X_in)
```

8.2 Ανίχνευση ηλικίας με χρήση αλγορίθμων παλινδρόμησης

Για την αξιολόγηση των μοντέλων παλινδρόμησης που εκπαιδεύτηκαν χρησιμοποιήθηκαν οι μετρικές MAE, MAPE, MSE, RMSE, R-squared και η τυπική απόκλιση (STD), οι οποίες αναλύθηκαν στην ενότητα 3.5.2. Επίσης, θεωρούμε για την παλινδρόμηση την μετρική ακρίβεια (accuracy) που υπολογίζεται ως εξής:

$$accuracy = 100\% - MAPE$$

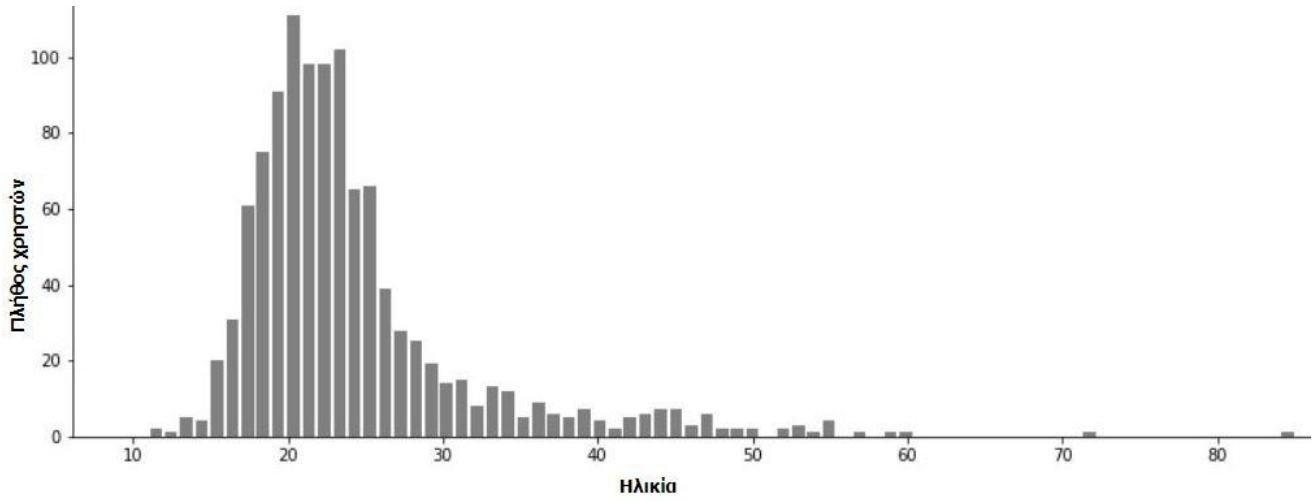
Για την επιλογή του βέλτιστου αλγορίθμου συνυπολογίζονται όλες οι μετρικές μαζί με το accuracy, ώστε σε περιπτώσεις που οι τιμές τους δεν διαφέρουν σε μεγάλο βαθμό θα προκριθεί

ως βέλτιστο το μοντέλο που καταγράφει τις περισσότερες καλύτερες επιδόσεις στο σύνολο των μετρικών που εξετάζονται.

Τα δεδομένα χωρίστηκαν τυχαία σε 80% δεδομένα εκπαίδευσης (training set) και 20% δεδομένα δοκιμής (test set) για τις ανάγκες της παρούσας διπλωματικής εργασίας. Ο κώδικας που εφαρμόζει τη συνάρτηση αυτή φαίνεται παρακάτω:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2,
                                                    random_state=42)
```

Ωστόσο, υπάρχουν ακόμα κάποιες ηλικίες που έχουν πολύ μικρή παρουσία στο σύνολο των δεδομένων εισόδου με μία ή δύο εμφανίσεις καθώς και ορισμένες που δεν αντιπροσωπεύονται καθόλου. Αυτό το γεγονός όμως θα δυσκολέψει σημαντικά την εκπαίδευση των αλγορίθμων παλινδρόμησης στο να προβλέψουν αποτελεσματικά τιμές για όλες τις ηλικίες αφού ενδέχεται να μην υπάρχουν καθόλου δείγματα για αυτές στο test set. Η κατανομή των χρηστών στο test set απεικονίζεται με τη μορφή γραφήματος στην Εικόνα 8.1.

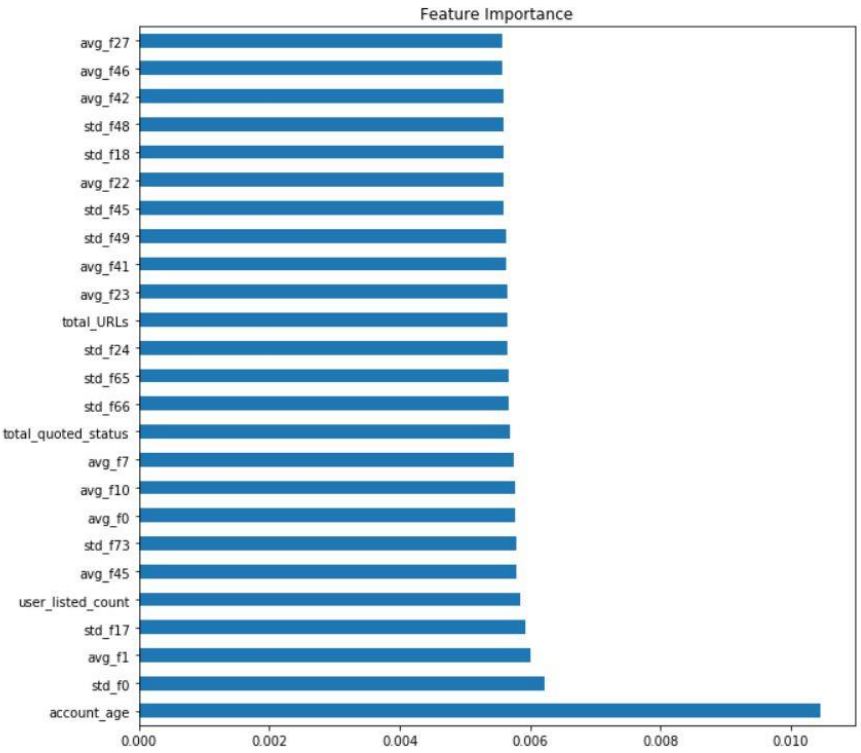


Εικόνα 8.1: Κατανομή χρηστών στο test set για την παλινδρόμηση

Σχετικά με την σπουδαιότητα των χαρακτηριστικών αυτή υλοποιήθηκε με τον αλγόριθμο ταξινόμησης τυχαίων δέντρων (Extremely Randomized Trees Classifier ή Extra Trees Classifier). Υπολόγισε τα 25 σημαντικότερα features για την είσοδο στους αλγορίθμους παλινδρόμησης και το πιο σπουδαίο αναδείχθηκε το “account_age”, ενώ βαρύτητα είχαν και τα “user_listed_count”, “total_quoted_status” και “total_URLs”. Τα υπόλοιπα πιο σημαντικά χαρακτηριστικά ήταν ορισμένα από τα αριθμητικά δεδομένα των tweets που αναλύθηκαν στην ενότητα 7.6. Ο κώδικας που εφαρμόζει τη μέθοδο και απεικονίζει τα αποτελέσματα της σε γράφημα παρουσιάζεται παρακάτω, ενώ στην Εικόνα 8.2 παρουσιάζεται το αντίστοιχο γράφημα που δημιουργείται.

```
from sklearn.ensemble import ExtraTreesClassifier
model = ExtraTreesClassifier(random_state=42)
model.fit(X_in,y)
# plot graph of feature importance
```

```
feat_importances = pd.Series(model.feature_importances_, index=X_in.columns)
feat_importances.nlargest(25).plot(kind='barh')
plt.gcf().set_size_inches(10, 10)
plt.title("Feature Importance")
plt.show()
```



Εικόνα 8.2: Σπουδαιότητα χαρακτηριστικών για την παλινδρόμηση

Ο υπολογισμός των μετρικών για κάθε μοντέλο παλινδρόμησης έγινε με την συνάρτηση:

```
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from math import sqrt
from statistics import stdev
def evaluation_metrics(model, model_name, y_train_pred,
                       y_test_pred, y_train, y_test):
    # mae
    mae = mean_absolute_error(y_test, y_test_pred)
    # std
    std = pd.np.r_[y_test, y_test_pred].std()
    #mape
    mape = np.mean(np.abs((y_test - y_test_pred) / y_test)) * 100
    # mse
    mse = mean_squared_error(y_test, y_test_pred)
    #rmse
    rmse_train = sqrt(mean_squared_error(y_train, y_train_pred))
    rmse_test = sqrt(mean_squared_error(y_test, y_test_pred))
    #r2 score
    r2_train = r2_score(y_train, y_train_pred)
    r2_test = r2_score(y_test, y_test_pred)
```

```
    # logs
    print("Model:", model)
    print("MAE:", mae)
    print("STD:", std)
    print("MAPE:", mape)
    print("MSE:", mse)
    print("RMSE TRAIN:", rmse_train)
    print("RMSE TEST:", rmse_test)
    print("R2 SCORE TRAIN:", r2_train)
    print("R2 SCORE TEST:", r2_test)
```

Για την αποθήκευση των μοντέλων και των παραγόμενων τιμών για την πρόβλεψη της ηλικίας από την παλινδρόμηση χρησιμοποιήθηκε η παρακάτω συνάρτηση που έχει ως παραμέτρους εισόδου το test set (y_test) και τις προβλέψεις (y_pred) του αλγορίθμου:

```
from sklearn.externals import joblib
def save_res(model ,model_name, y_pred, y_test):
    # create df with y_test
    list_l = []
    list_l.append(y_test)
    df_1 = pd.DataFrame(list_l)
    df_1 = df_1.transpose()
    df_1 = df_1.reset_index(drop=True)
    # create df with y_pred
    list_n = []
    list_n.append(y_pred)
    df_2 = pd.DataFrame(list_n)
    df_2 = df_2.transpose()
    df_2 = df_2.reset_index(drop=True)
    # merge all in one df
    df_1['pred'] = df_2
    df_1.to_csv('/path/' + model_name + '_test_df.csv',
                quoting=csv.QUOTE_NONNUMERIC, index=False)
    # Save the model using joblib in a file
    joblib.dump(model, '/path/model_' + model_name + '.pkl')
```

Για την απεικόνιση των αποτελεσμάτων κάθε αλγορίθμου παλινδρόμησης σε μορφή διαγράμματος αναπτύχθηκε μία συνάρτηση που απεικονίζει στο ίδιο γράφημα το test set, με την ετικέτα test και πράσινο χρώμα, και τις προβλέψεις, με την ετικέτα pred και μπλε χρώμα. Ο κώδικας που την υλοποιεί είναι ο παρακάτω:

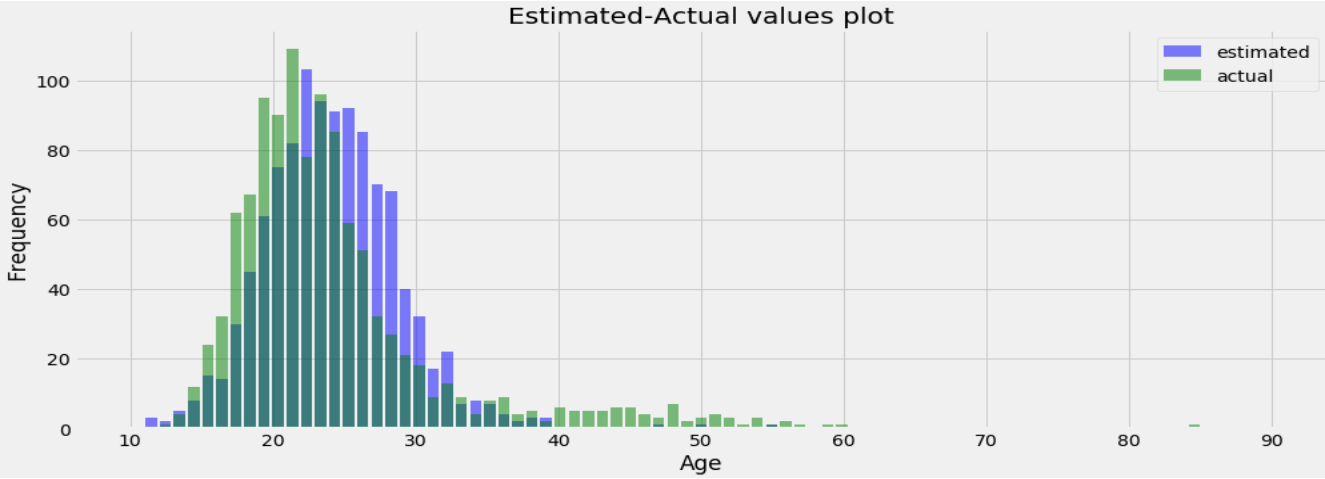
```
from mathplotlib import pyplot
def doublePlotFunction(y_pred, y_test):
    pyplot.hist(y_pred, bins=81, range=[10, 90], color='blue',
                rwidth=0.8, align='mid', alpha=0.5, label='pred')
    pyplot.hist(y_test, bins=81, range=[10, 90], color='green',
                rwidth=0.8, align='mid', alpha=0.5, label='test')
    pyplot.legend(loc='upper right')
    plt.title("Predictions-Test plot")
    pyplot.gcf().set_size_inches(15, 5)
    pyplot.show()
```

8.2.1 Γραμμική παλινδρόμηση (Linear Regression)

Για την επιλογή του βέλτιστου μοντέλου της Γραμμικής Παλινδρόμησης χρησιμοποιήθηκε η βελτίωση υπερπαραμέτρων μέσω του αλγορίθμου RandomizedSearchCV. Έγινε αναζήτηση για συγκεκριμένες παραμέτρους και τις αντίστοιχες τιμές τους οι οποίες ορίστηκαν στον κώδικα. Πρόκειται για μία σύντομη διεργασία που διήρκησε 0,8 δευτερόλεπτα. Η εκπαίδευση του μοντέλου, η παρουσίαση και η αποθήκευση των αποτελεσμάτων του ήταν μία εξίσου γρήγορη διαδικασία και χρειάστηκε περίπου 1,4 δευτερόλεπτα για να ολοκληρωθεί. Ο κώδικας που υλοποιεί τη Γραμμική Παλινδρόμηση φαίνεται στη συνέχεια.

```
from sklearn.linear_model import LinearRegression
linear = LinearRegression()
# fit intercept values
fit_intercept = [True, False]
# normalize values
normalize = [True, False]
# Create the random grid
random_grid = {'fit_intercept': fit_intercept, 'normalize': normalize}
#perform hyperparameter tuning
linear_random = RandomizedSearchCV(estimator = linear,
                                   param_distributions = random_grid, n_iter = 5,
                                   cv = 5, verbose=2, random_state=42, n_jobs = -1)
linear_random.fit(X_train, y_train)
#best estimator model from randomsearch process
best_model = linear_random.best_estimator_
best_model.fit(X_train,y_train)
#Evaluate
model_name = 'Linear'
y_linear_train = best_model.predict(X_train)
y_linear_test = best_model.predict(X_test)
evaluation_metrics(best_model, model_name, y_linear_train,
                   y_linear_test, y_train, y_test)
save_res(best_model, model_name, y_linear_test, y_test)
plotFunction(y_linear_test, "Linear Regression predictions")
doublePlotFunction(y_linear_test, y_test)
```

Το διάγραμμα πραγματικών τιμών και εκτιμήσεων είναι:



Εικόνα 8.3: Διάγραμμα εκτιμήσεων-πραγματικών τιμών για τη Γραμμική Παλινδρόμηση

Για την πρόβλεψη της ακριβούς τιμής της ηλικίας μέσω της γραμμικής παλινδρόμησης παρουσιάζεται MAE ίσο με 4,74 και accuracy 80,99% όπως ορίστηκε νωρίτερα. Ιδιαίτερα χαμηλή ίση με 0,09 είναι η τιμή του R2. Συνολικά τα αποτελέσματα των μετρικών είναι τα εξής:

Πίνακας 8.1: Αποτελέσματα αξιολόγησης για τη Γραμμική Παλινδρόμηση

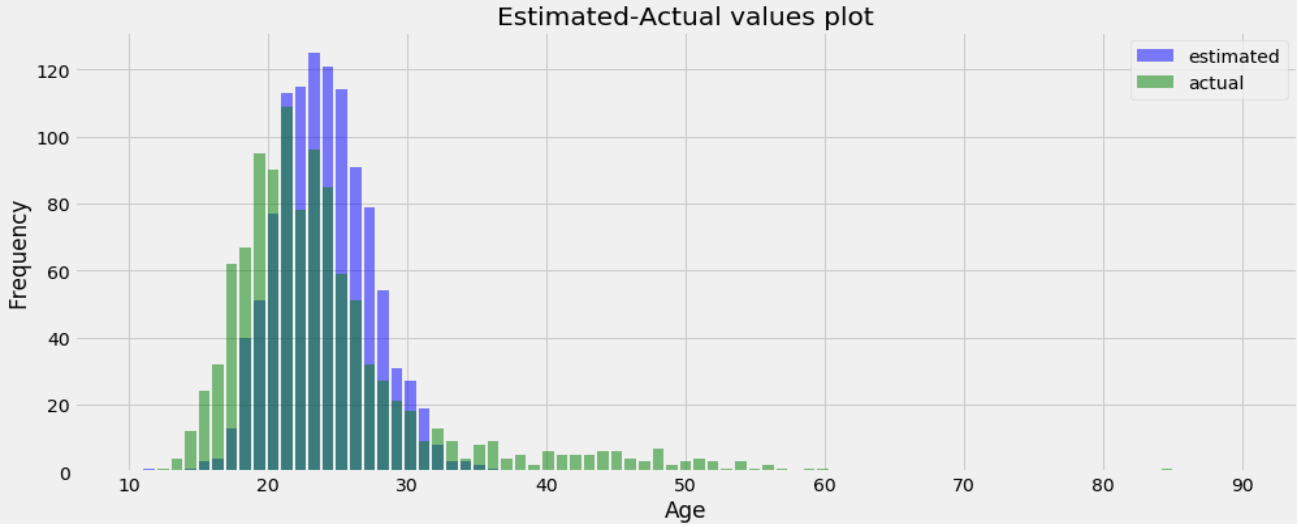
Γραμμική Παλινδρόμηση						
MAE	**MAPE**	**MSE**	**RMSE**	**R2**	**STD**	**ACCURACY**
4,74	19,01	55,44	7,45	0,09	6,64	80,99%

8.2.2 Παλινδρόμηση Lasso

Το βέλτιστο μοντέλο της Παλινδρόμησης Lasso βρέθηκε μέσω της τεχνικής της βελτίωσης υπερπαραμέτρων μέσω του αλγορίθμου RandomizedSearchCV. Η αναζήτηση έγινε για συγκεκριμένες παραμέτρους και τις αντίστοιχες τιμές τους όπως δηλώθηκαν στον κώδικα. Η διεργασία αυτή ήταν αρκετά σύντομη και διήρκησε 4,43 δευτερόλεπτα μέχρι να βρεθεί το πιο αποδοτικό μοντέλο. Η εκπαίδευση του μοντέλου, η παρουσίαση και η αποθήκευση των αποτελεσμάτων του ήταν μία εξίσου γρήγορη διαδικασία και χρειάστηκε περίπου 2,2 δευτερόλεπτα για να ολοκληρωθεί. Ο κώδικας που εφαρμόζει τον αλγόριθμο της Παλινδρόμησης Lasso είναι ο παρακάτω.

```
from sklearn.linear_model import Lasso
lasso = Lasso()
# fit intercept values
fit_intercept = [True, False]
# alpha values
alpha = [0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100]
# max_iter values
max_iter = [10, 100, 200, 500, 1000, 1500, 2000]
# Create the random grid
random_grid = {'alpha': alpha, 'fit_intercept': fit_intercept,
               'max_iter': max_iter}
lasso_random = RandomizedSearchCV(estimator = lasso,
                                  param_distributions = random_grid,
                                  n_iter = 5, cv = 5, verbose=2,
                                  random_state=42, n_jobs = -1)
lasso_random.fit(X_train, y_train)
#best estimator model from randomsearch process
best_model = lasso_random.best_estimator_
best_model.fit(X_train,y_train)
#Evaluate
model_name = 'Lasso'
y_lasso_train = best_model.predict(X_train)
y_lasso_test = best_model.predict(X_test)
evaluation_metrics(best_model, model_name, y_lasso_train,
                   y_lasso_test, y_train, y_test)
save_res(best_model, model_name, y_lasso_test, y_test)
plotFunction(y_lasso_test, "Lasso Regression predictions")
doublePlotFunction(y_lasso_test, y_test)
```

Το διάγραμμα πραγματικών τιμών και εκτιμήσεων είναι:



Εικόνα 8.4: Διάγραμμα εκτιμήσεων-πραγματικών τιμών για την Παλινδρόμηση Lasso

Η πρόβλεψη για την ακριβή τιμή της ηλικίας μέσω της παλινδρόμησης Lasso παρουσίασε MAE ίσο με 4,58 και accuracy 81,94% όπως ορίστηκε νωρίτερα. Συνολικά τα αποτελέσματα των μετρικών είναι τα εξής:

Πίνακας 8.2: Αποτελέσματα αξιολόγησης για την Παλινδρόμηση Lasso

Παλινδρόμηση Lasso						
MAE	**MAPE**	**MSE**	**RMSE**	**R2**	**STD**	**ACCURACY**
4,58	18,06	49,25	7,02	0,19	6,02	81,94%

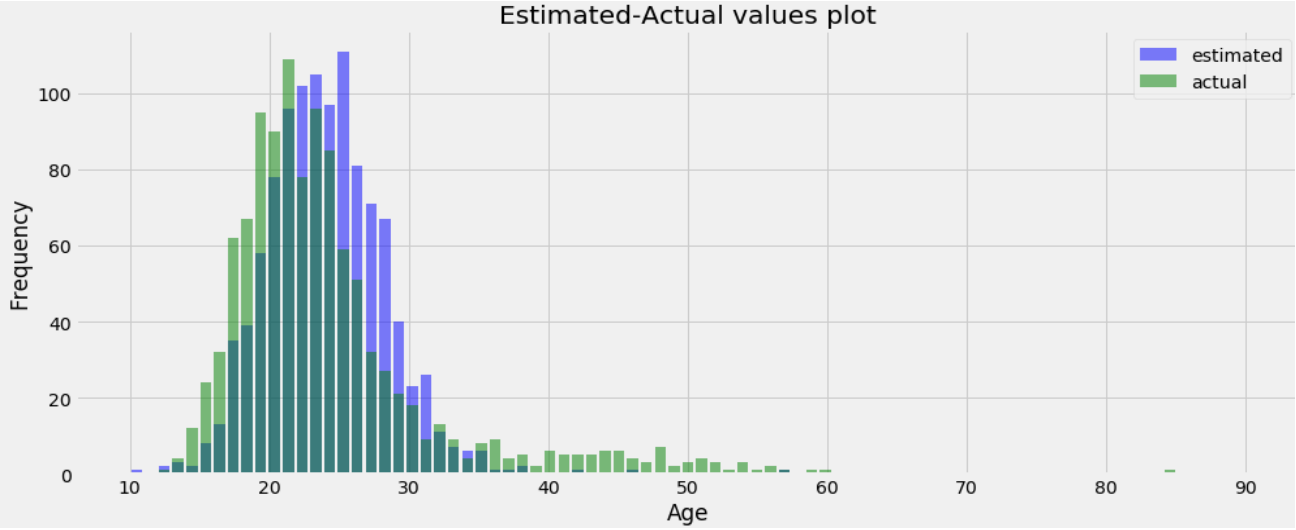
8.2.3 Παλινδρόμηση Ridge

Η επιλογή του βέλτιστου μοντέλου της Παλινδρόμησης Ridge μέσω της μεθόδου του hyperparameter tuning με τον αλγόριθμο RandomizedSearchCV ήταν σύντομη διεργασία και διήρκησε περίπου 0,5 δευτερόλεπτα μέχρι να βρεθεί το πιο αποδοτικό μοντέλο. Για την εκπαίδευση του μοντέλου, την παρουσίαση και την αποθήκευση των αποτελεσμάτων χρειάστηκαν περίπου 1,45 δευτερόλεπτα, όντως μία αρκετά γρήγορη διαδικασία. Ο κώδικας που εφαρμόζει τον αλγόριθμο της Παλινδρόμησης Ridge φαίνεται παρακάτω.

```
from sklearn.linear_model import Ridge
ridge = Ridge()
# fit intercept values
fit_intercept = [True, False]
# alpha values
alpha = [0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100]
# max_iter values
max_iter = [10, 100, 200, 500, 1000, 1500, 2000]
# Create the random grid
random_grid = {'alpha': alpha, 'fit_intercept': fit_intercept,
               'max_iter': max_iter}
ridge_random = RandomizedSearchCV(estimator = ridge,
```

```
                                    param_distributions = random_grid,
                                    n_iter = 5, cv = 5, verbose=2,
                                    random_state=42, n_jobs = -1)
ridge_random.fit(X_train, y_train)
#best estimator model from randomsearch process
best_model = ridge_random.best_estimator_
best_model.fit(X_train,y_train)
#Evaluate
model_name = 'Ridge'
y_ridge_train = best_model.predict(X_train)
y_ridge_test = best_model.predict(X_test)
evaluation_metrics(best_model, model_name, y_ridge_train,
                   y_ridge_test, y_train, y_test)
save_res(best_model, model_name, y_ridge_test, y_test)
plotFunction(y_ridge_test, "Ridge Regression predictions")
doublePlotFunction(y_ridge_test, y_test)
```

Το διάγραμμα πραγματικών τιμών και εκτιμήσεων είναι:



Εικόνα 8.5: Διάγραμμα εκτιμήσεων-πραγματικών τιμών για την Παλινδρόμηση Ridge

Η πρόβλεψη της παλινδρόμησης Ridge για την ακριβή τιμή της ηλικίας παρουσίασε MAE ίσο με 4,55 και accuracy 81,93% όπως ορίστηκε νωρίτερα. Συνολικά τα αποτελέσματα των μετρικών είναι τα εξής:

Πίνακας 8.3: Αποτελέσματα αξιολόγησης για την Παλινδρόμηση Ridge

Παλινδρόμηση Ridge						
MAE	**MAPE**	**MSE**	**RMSE**	**R2**	**STD**	**ACCURACY**
4,55	18,07	48,91	6,99	0,20	6,29	81,93%

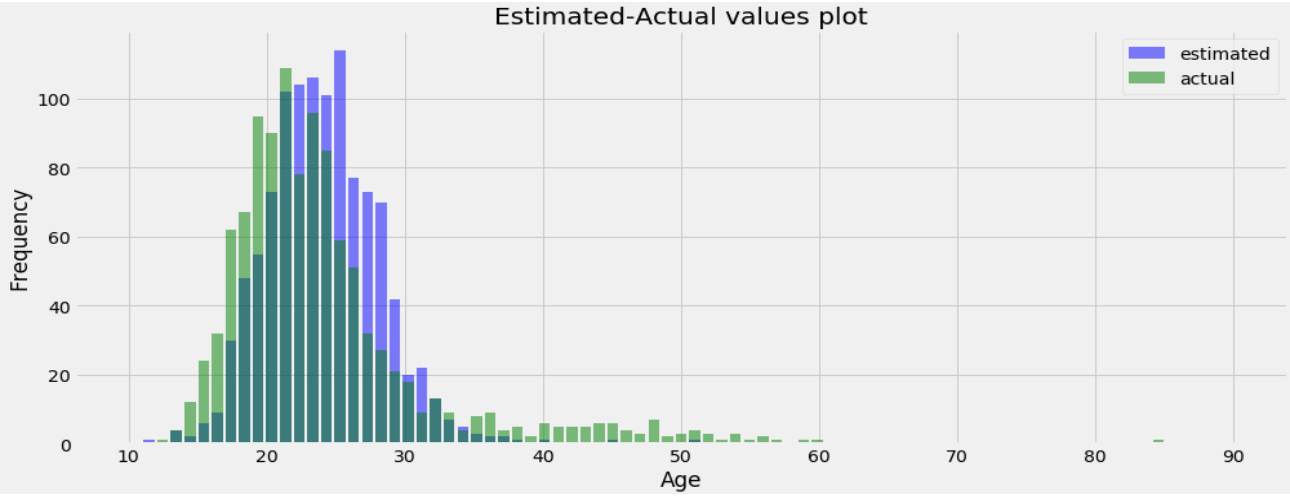
8.2.4 Παλινδρόμηση ElasticNet

Ο προσδιορισμός του βέλτιστου μοντέλου της Παλινδρόμησης ElasticNet έγινε με τον αλγόριθμο RandomizedSearchCV για την βελτίωση των υπερπαραμέτων. Η αναζήτηση για συγκεκριμένες παραμέτρους και τις αντίστοιχες τιμές τους όπως αυτές ορίστηκαν στον κώδικα

διήρκησε περίπου 2,58 δευτερόλεπτα ώστε να εξαχθεί το πιο αποδοτικό μοντέλο. Μία εξίσου σύντομη διεργασία ήταν και η εκπαίδευση του μοντέλου, η παρουσίαση και η αποθήκευση των αποτελεσμάτων του που διήρκησε περίπου 1,35 δευτερόλεπτα. Ο κώδικας για την Παλινδρόμηση Ridge είναι ο παρακάτω.

```
from sklearn.linear_model import ElasticNet
elasticNet = ElasticNet()
# fit intercept values
fit_intercept = [True, False]
# alpha values
alpha = [0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100]
# l1_ratio
l1_ratio = [0.25, 0.5, 0.75]
# max_iter values
max_iter = [10, 100, 200, 500, 1000, 1500, 2000]
# Create the random grid
random_grid = {'alpha': alpha, 'fit_intercept': fit_intercept,
               'l1_ratio': l1_ratio, 'max_iter': max_iter}
elasticNet_random = RandomizedSearchCV(estimator = elasticNet,
                                       param_distributions = random_grid,
                                       n_iter = 5, cv = 5, verbose=2,
                                       random_state=42, n_jobs = -1)
elasticNet_random.fit(X_train, y_train)
#best estimator model from randomsearch process
best_model = elasticNet_random.best_estimator_
best_model.fit(X_train,y_train)
#Evaluate
model_name = 'ElasticNet'
y_elasticNet_train = best_model.predict(X_train)
y_elasticNet_test = best_model.predict(X_test)
evaluation_metrics(best_model, model_name, y_elasticNet_train,
                   y_elasticNet_test, y_train, y_test)
save_res(best_model, model_name, y_elasticNet_test, y_test)
plotFunction(y_elasticNet_test, "ElasticNet Regression predictions")
doublePlotFunction(y_elasticNet_test, y_test)
```

Το διάγραμμα πραγματικών τιμών και εκτιμήσεων είναι:



Εικόνα 8.6: Διάγραμμα εκτιμήσεων-πραγματικών τιμών για την Παλινδρόμηση ElasticNet

Η πρόβλεψη της ακριβούς τιμής της ηλικίας μέσω της παλινδρόμησης ElasticNet παρουσίασε MAE ίσο με 4,53 και accuracy 82,05% όπως ορίστηκε νωρίτερα. Ο συντελεστής προσδιορισμού R2 ήταν ίσος με 0,21. Συνολικά τα αποτελέσματα των μετρικών είναι τα εξής:

Πίνακας 8.4: Αποτελέσματα αξιολόγησης για την Παλινδρόμηση ElasticNet

Παλινδρόμηση ElasticNet						
MAE	**MAPE**	**MSE**	**RMSE**	**R2**	**STD**	**ACCURACY**
4,53	17,95	48,37	6,95	0,21	6,23	82,05%

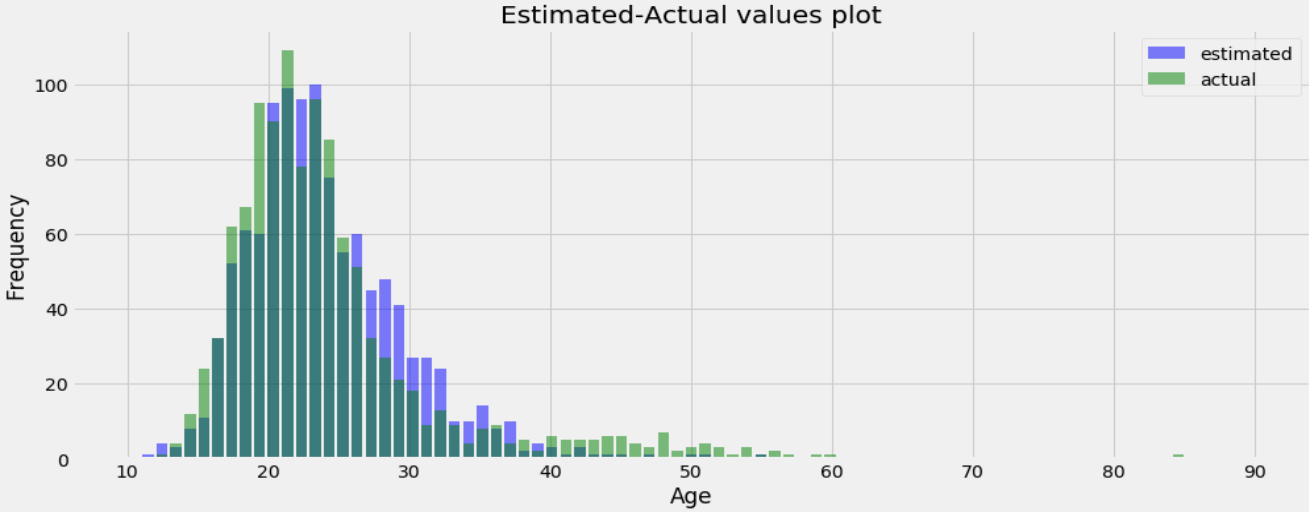
8.2.5 Παλινδρόμηση XGBoost

Η τεχνική της βελτίωσης υπερπαραμέτρων μέσω του αλγορίθμου RandomizedSearchCV χρησιμοποιήθηκε για την εύρεση του βέλτιστου μοντέλου για την Παλινδρόμησης XGBoost, όπου πραγματοποιήθηκε αναζήτηση για συγκεκριμένες παραμέτρους και τις αντίστοιχες τιμές τους όπως αυτές ορίστηκαν στον κώδικα. Διήρκησε περίπου 7 λεπτά και 17 δευτερόλεπτα μέχρι να βρεθεί το πιο αποδοτικό μοντέλο. Η εκπαίδευση του μοντέλου, η παρουσίαση και η αποθήκευση των αποτελεσμάτων του διήρκησε περίπου 27 δευτερόλεπτα για να ολοκληρωθεί. Ο παρακάτω κώδικας εφαρμόζει τον αλγόριθμο της Παλινδρόμησης XGBoost.

```
import xgboost as xgb
from xgboost import XGBRegressor
xgboost = XGBRegressor()
# fit intercept values
booster = ['gbtree', 'gblinear', 'dart']
# gamma
gamma = [0.01, 0.1, 0, 1, 2, 5, 10, 20, 50]
# normalize values
n_estimators = [10, 20, 50, 100, 200, 500, 1000]
# max_depth
max_depth = [2, 3, 4, 5, 6, 10, 16, 20, 32, 48, 54]
# alpha values
alpha = [0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100]
# eta0 values
eta0 = [0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1]
# min_child_weight
min_child_weight = [1, 2, 4, 8 ,10]
# Create the random grid
random_grid = {'n_estimators': n_estimators, 'gamma': gamma,
               'alpha': alpha, 'max_depth': max_depth,
               'eta0': eta0, 'min_child_weight': min_child_weight,
               'booster': booster}
xgboost_random = RandomizedSearchCV(estimator = xgboost,
                                    param_distributions = random_grid,
                                    n_iter = 5, cv = 5, verbose=2,
                                    random_state=42,
                                    n_jobs = -1)
xgboost_random.fit(X_train, y_train)
#best estimator model from randomsearch process
best_model = xgboost_random.best_estimator_
best_model.fit(X_train,y_train)
```

```
#Evaluate
model_name = 'XGBoost'
y_xgb_train = best_model.predict(X_train)
y_xgb_test = best_model.predict(X_test)
evaluation_metrics(best_model, model_name, y_xgb_train,
                   y_xgb_test, y_train, y_test)
save_res(best_model, model_name, y_xgb_test, y_test)
plotFunction(y_xgb_test, "XGBoost predictions")
doublePlotFunction(y_xgb_test, y_test)
```

Το διάγραμμα πραγματικών τιμών και εκτιμήσεων είναι:



Εικόνα 8.7: Διάγραμμα εκτιμήσεων-πραγματικών τιμών για την Παλινδρόμηση XGBoost

Η πρόβλεψη για την ακριβή τιμή της ηλικίας μέσω της παλινδρόμησης XGBoost παρουσίασε MAE ίσο με 4,09 και accuracy 83,52% όπως ορίστηκε νωρίτερα. Τα MSE και RMSE είχαν τις μικρότερες τιμές με 37,27 και 6,1 αντίστοιχα. Επίσης, ο συντελεστής R2 παρουσίασε την πιο υψηλή τιμή ίση με 0,39. Συνολικά τα αποτελέσματα των μετρικών είναι τα εξής:

Πίνακας 8.5: Αποτελέσματα αξιολόγησης για την Παλινδρόμηση XGBoost

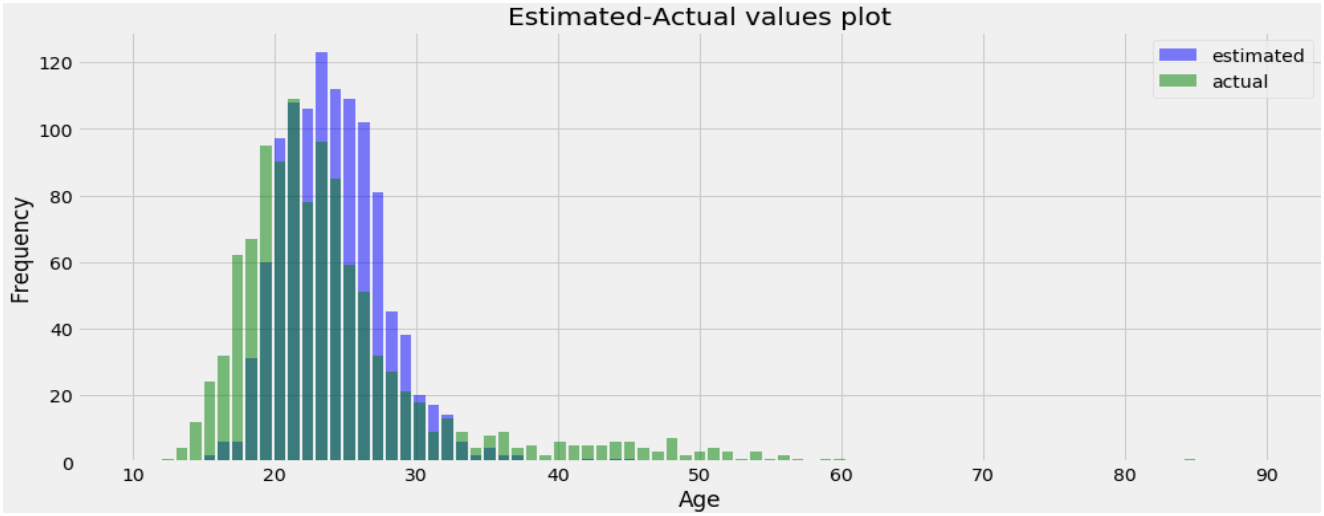
Παλινδρόμηση XGBoost						
MAE	**MAPE**	**MSE**	**RMSE**	**R2**	**STD**	**ACCURACY**
4,09	16,48	37,27	6,1	0,39	6,59	83,52%

8.2.6 Παλινδρόμηση με Τυχαία Δάση (Random Forest)

Το βέλτιστο μοντέλο για την Παλινδρόμηση με Τυχαία Δάση βρέθηκε με την τεχνική της βελτίωσης υπερπαραμέτρων εφαρμόζοντας τον αλγόριθμο RandomizedSearchCV, όπως και στα προηγούμενα πειράματα. Η αναζήτηση πραγματοποιήθηκε για αρκετούς συνδυασμούς παραμέτρων και των αντίστοιχων τιμών τους που δηλώθηκαν στον κώδικα. Η διεργασία αυτή ήταν αρκετά χρονοβόρα και διήρκησε 1 ώρα περίπου μέχρι να ανακαλυφθεί το πιο αποδοτικό μοντέλο. Σχετικά με την εκπαίδευση του μοντέλου, την παρουσίαση και την αποθήκευση των αποτελεσμάτων του η διαδικασία είχε διάρκεια περίπου 11 λεπτά και 30 δευτερόλεπτα. Ο κώδικας που εφαρμόστηκε για τον αλγόριθμο της Παλινδρόμησης με Τυχαία Δάση είναι ο παρακάτω.

```
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor(random_state = 42)
# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 10, stop = 2000, num = 200)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(2, 110, num = 55)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 8, 10]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 4]
# Method of selecting samples for training each tree
bootstrap = [True, False]
# Create the random grid
random_grid = {'n_estimators': n_estimators, 'max_features': max_features,
               'max_depth': max_depth, 'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf, 'bootstrap': bootstrap}
rf_random = RandomizedSearchCV(estimator = rf, param_distributions = random_grid,
                       n_iter = 5, cv = 5, verbose=2, random_state=42, n_jobs = -1)
rf_random.fit(X_train, y_train)
#best estimator model from randomsearch process
best_model = rf_random.best_estimator_
best_model.fit(X_train,y_train)
#Evaluate
model_name = 'Random Forest'
y_rf_train = best_model.predict(X_train)
y_rf_test = best_model.predict(X_test)
evaluation_metrics(best_model, model_name, y_rf_train, y_rf_test, y_train, y_test)
save_res(best_model, model_name, y_rf_test, y_test)
plotFunction(y_rf_test, "Random Forest Regression predictions")
doublePlotFunction(y_rf_test, y_test)
```

Το διάγραμμα πραγματικών τιμών και εκτιμήσεων είναι:



Εικόνα 8.8: Διάγραμμα εκτιμήσεων-πραγματικών τιμών για την Παλινδρόμηση με Τυχαία Δάση

Η πρόβλεψη της ακριβούς τιμής της ηλικίας για την παλινδρόμηση με Τυχαία Δάση εμφάνισε MAE ίσο με 4,17 και accuracy 83,24% όπως ορίστηκε νωρίτερα, ενώ επέδειξε συντελεστή R2 ίσο με 0,32. Συνολικά τα αποτελέσματα των μετρικών είναι τα εξής:

Πίνακας 8.6: Αποτελέσματα αξιολόγησης για την Παλινδρόμηση με Τυχαία Δάση

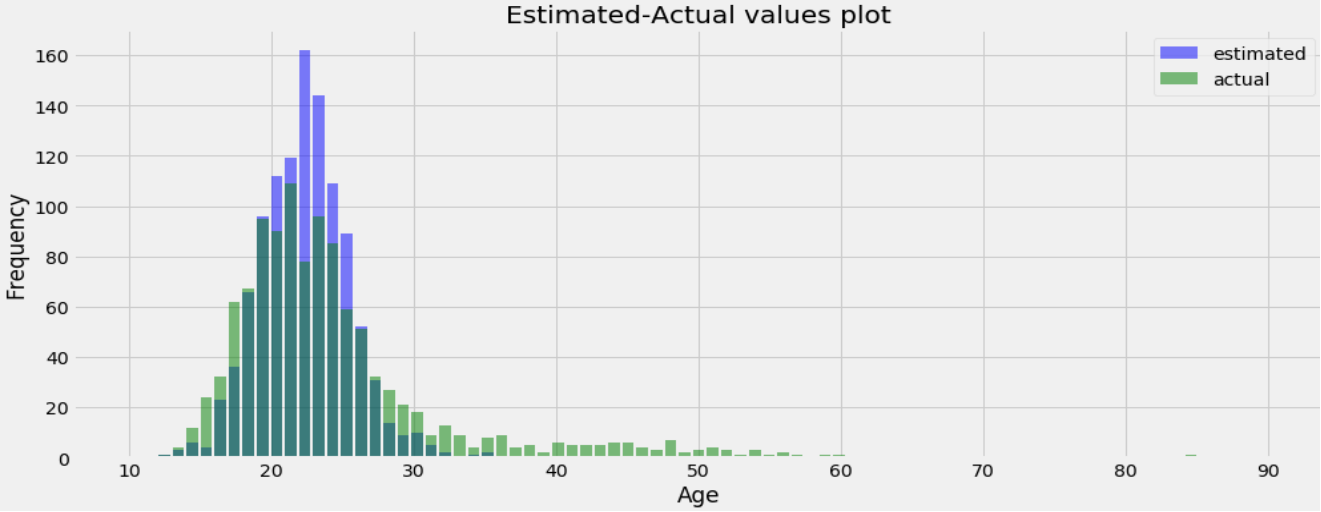
Παλινδρόμηση με Τυχαία Δάση						
MAE	**MAPE**	**MSE**	**RMSE**	**R2**	**STD**	**ACCURACY**
4,17	16,67	41,38	6,43	0,32	6,09	83,33%

8.2.7 Παλινδρόμηση με SVR

Η διεργασία της επιλογής του βέλτιστου μοντέλου της Παλινδρόμησης με SVR που έγινε μέσω της μεθόδου της βελτίωσης υπερπαραμέτρων μέσω του αλγορίθμου RandomizedSearchCV διήρκησε 1 λεπτό και 36 δευτερόλεπτα. Η διαδικασία της εκπαίδευσης του μοντέλου, της παρουσίασης και της αποθήκευσης των αποτελεσμάτων εκτελέστηκε για περίπου 47 δευτερόλεπτα μέχρι να ολοκληρωθεί. Η εφαρμογή της Παλινδρόμησης με SVR φαίνεται στον παρακάτω κώδικα.

```
from sklearn import svm
from sklearn.svm import SVR
svr = SVR()
# C values
C = [0.001, 0.01, 0.1, 1, 2, 5, 10, 100]
# kernel values
kernel = ['linear', 'poly', 'rbf', 'sigmoid', 'precomputed']
# gamma
gamma = ['scale', 'auto']
# Create the random grid
random_grid = {'C': C, 'gamma': gamma,
               'kernel': kernel}
svr_random = RandomizedSearchCV(estimator = svr,
                                param_distributions = random_grid,
                                n_iter = 5,
                                cv = 5,
                                verbose=2,
                                random_state=42, n_jobs = -1)
svr_random.fit(X_train, y_train)
#best estimator model from randomsearch process
best_model = svr_random.best_estimator_
best_model.fit(X_train,y_train)
#Evaluate
model_name = 'SVR'
y_svr_train = best_model.predict(X_train)
y_svr_test = best_model.predict(X_test)
evaluation_metrics(best_model, model_name, y_svr_train,
                   y_svr_test, y_train, y_test)
save_res(best_model, model_name, y_svr_test, y_test)
plotFunction(y_svr_test, "SVR Regression predictions")
doublePlotFunction(y_svr_test, y_test)
```

Το διάγραμμα πραγματικών τιμών και εκτιμήσεων είναι:



Εικόνα 8.9: Διάγραμμα εκτιμήσεων-πραγματικών τιμών για την Παλινδρόμηση με SVR

Το μοντέλο της παλινδρόμησης με SVR παρουσίασε MAE ίσο με 4,07 και accuracy 85,41% όπως ορίστηκε νωρίτερα για την πρόβλεψη της ακριβούς ηλικίας. Είχε επίσης MSE 50,81 και RMSE 7,13. Ο συντελεστής προσδιορισμού R2 είχε χαμηλή τιμή 0,17, ενώ τα συνολικά τα αποτελέσματα των μετρικών είναι τα εξής:

Πίνακας 8.7: Αποτελέσματα αξιολόγησης για την Παλινδρόμηση με SVR

Παλινδρόμηση με SVR						
MAE	**MAPE**	**MSE**	**RMSE**	**R2**	**STD**	**ACCURACY**
4,07	14,59	50,81	7,13	0,17	6,0	85,41%

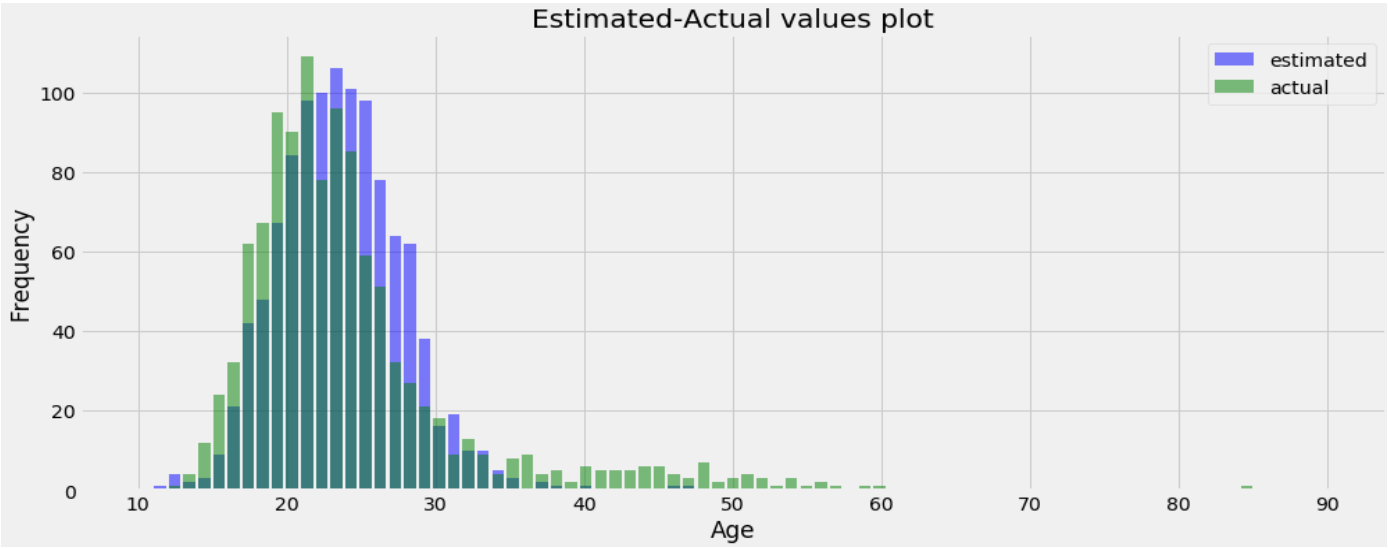
8.2.8 Παλινδρόμηση με SGD

Η εύρεση του βέλτιστου μοντέλου της Παλινδρόμησης με SGD έγινε με τη χρήση της τεχνικής βελτίωσης υπερπαραμέτρων με τον αλγορίθμου RandomizedSearchCV. Η αναζήτηση πραγματοποιήθηκε για ένα σύνολο παραμέτρων και των τιμών τους όπως αυτές ορίστηκαν στον κώδικα. Η διεργασία αυτή ήταν σύντομη και διήρκησε περίπου 26 δευτερόλεπτα μέχρι να βρεθεί το καλύτερο μοντέλο. Η εκπαίδευση του μοντέλου, η παρουσίαση και η αποθήκευση των αποτελεσμάτων του ήταν μία ταχύτατη διαδικασία και χρειάστηκε περίπου 1,5 δευτερόλεπτα για να ολοκληρωθεί. Ο κώδικας που εφαρμόζει τον αλγόριθμο της Παλινδρόμησης SGD είναι ο παρακάτω.

```
from sklearn.linear_model import SGDRegressor
sgd = SGDRegressor()
# fit intercept values
fit_intercept =  [True, False]
# alpha values
alpha = [0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100]
# l1_ratio values
l1_ratio = [0.15, 0.25, 0.5, 0.75]
# eta0 values
```

```
eta0 = [0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1]
# learning_rate values
learning_rate = ['invscaling', 'constant', 'adaptive']
# penalty values
penalty= ['l1', 'l2', 'elasticnet']
# max_iter values
max_iter = [10, 100, 200, 500, 1000, 1500, 2000]
# Create the random grid
random_grid = {'alpha': alpha, 'fit_intercept': fit_intercept,
               'l1_ratio': l1_ratio, 'eta0': eta0,
               'penalty': penalty, 'learning_rate': learning_rate,
               'max_iter': max_iter}
sgd_random = RandomizedSearchCV(estimator = sgd,
                                param_distributions = random_grid,
                                n_iter = 5, cv = 5, verbose=2,
                                random_state=42, n_jobs = -1)
sgd_random.fit(X_train, y_train)
#best estimator model from randomsearch process
best_model = sgd_random.best_estimator_
best_model.fit(X_train,y_train)
#Evaluate
model_name = 'SGD'
y_sgd_train = best_model.predict(X_train)
y_sgd_test = best_model.predict(X_test)
evaluation_metrics(best_model, model_name, y_sgd_train,
                   y_sgd_test, y_train, y_test)
save_res(best_model, model_name, y_sgd_test, y_test)
plotFunction(y_sgd_test, "SGD Regression predictions")
doublePlotFunction(y_sgd_test, y_test)
```

Το διάγραμμα πραγματικών τιμών και εκτιμήσεων είναι:



Εικόνα 8.10:Διάγραμμα εκτιμήσεων-πραγματικών τιμών για την Παλινδρόμηση με SGD

Η πρόβλεψη της ακριβούς τιμής της ηλικίας μέσω της παλινδρόμησης Lasso παρουσίασε MAE ίσο με 4,47 και accuracy 82,25% όπως ορίστηκε νωρίτερα. Συνολικά τα αποτελέσματα των μετρικών είναι τα εξής:

Πίνακας 8.8: Αποτελέσματα αξιολόγησης για την Παλινδρόμηση με SGD

Παλινδρόμηση με SGD						
MAE	**MAPE**	**MSE**	**RMSE**	**R2**	**STD**	**ACCURACY**
4,46	17,39	48,48	6,96	0,21	6,28	82,61%

8.2.9 Συνοπτικά αποτελέσματα των Αλγορίθμων Παλινδρόμησης

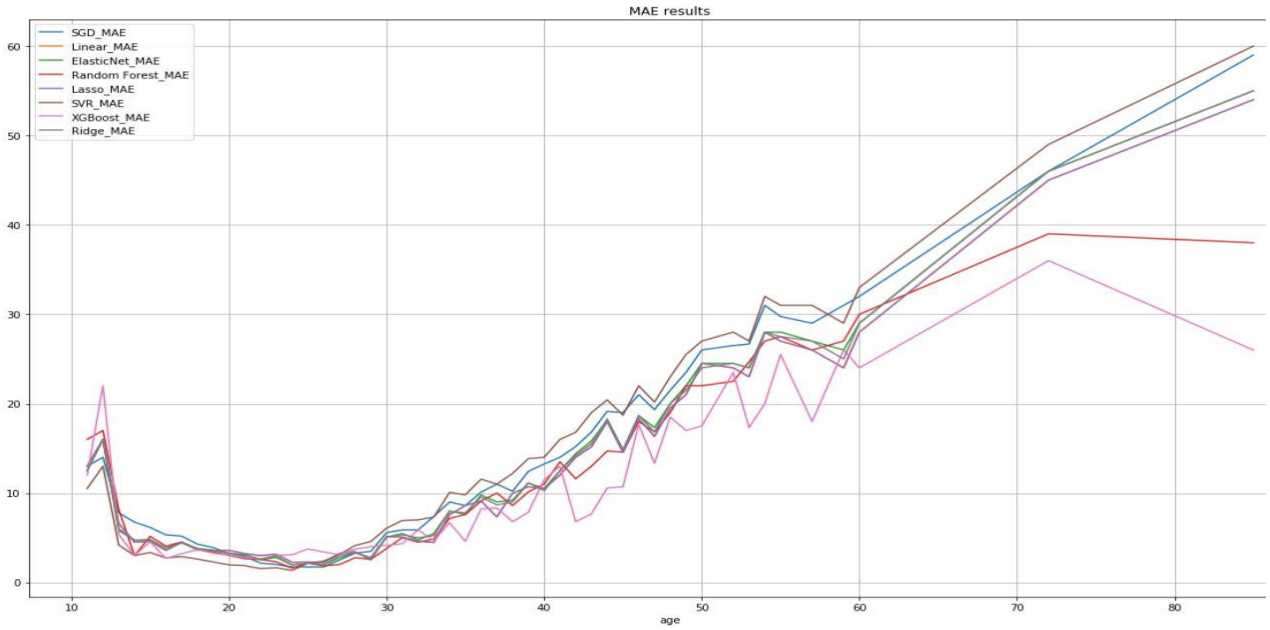
Συγκρίνοντας τα αποτελέσματα μεταξύ των αλγορίθμων παλινδρόμησης που εξετάστηκαν και φαίνονται συγκεντρωτικά στον πίνακα 8.9, διαπιστώνουμε ότι το μοντέλο που παράχθηκε από την εκτέλεση του αλγορίθμου **XGBoost** παρουσιάζει συνολικά την καλύτερη απόδοση. Πιο συγκεκριμένα, εμφανίζει εξαιρετικά χαμηλό **MAE ίσο με 4,09 έτη**, που αποτελεί τη δεύτερη καλύτερη επίδοση με ελάχιστη διαφορά μόλις 0,02 έτη από αυτή του SVR. Επίσης, σημειώνει ιδιαίτερα ικανοποιητική ακρίβεια (**accuracy) με 83,52%**, ενώ σε αυτή τη μετρική ο SVR έχει την καλύτερη με 85,41%. Όσον αφορά τις μετρικές **MSE και RMSE**, το μοντέλο XGBoost πετυχαίνει, με διαφορά από τα υπόλοιπα, **τις μικρότερες τιμές με 37,27 και 6,1 αντίστοιχα**, ενώ ακόμα έχει τον βέλτιστο συντελεστή **R2 με τιμή ίση με 0,39**. Το παραγόμενο μοντέλο του SVR υστερεί σημαντικά στις μετρικές MSE, RMSE και R2 έχοντας για κάθε μία τη δεύτερη μικρότερη τιμή με αποτέλεσμα να μην προκρίνεται ως η καλύτερη επιλογή. Επιπλέον, το μοντέλο παλινδρόμησης με Τυχαία Δάση παρουσιάζει ικανοποιητικές τιμές για τις διάφορες μετρικές που ελέγχθηκαν. Αντίθετα, η γραμμική παλινδρόμηση έχει τη χαμηλότερη απόδοση συνολικά για κάθε μετρική μέθοδο που αξιολογήθηκε. Συνεπώς, η παρούσα μελέτη, όντας ιδιαίτερα καινοτόμα αφού δεν έχει πραγματοποιηθεί παρόμοια έρευνα στο παρελθόν, προτείνει για την επίλυση του προβλήματος ανίχνευσης της ακριβούς ηλικίας των χρηστών του Twitter με την τεχνική της παλινδρόμησης (regression), το παραγόμενο μοντέλο του **XGBoost** αλγορίθμου. Ο Πίνακας 8.9 παρουσιάζει τα αποτελέσματα κάθε αλγορίθμου που δοκιμάστηκε και με πράσινο χρώμα σημειώνονται οι καλύτερες επιδόσεις για κάθε μετρική, ενώ με κόκκινο οι χειρότερες.

Πίνακας 8.9: Συνολικά αποτελέσματα αξιολόγησης των μοντέλων Παλινδρόμησης

Αλγόριθμοι Παλινδρόμησης	**Μετρικές**						
	MAE	**MAPE**	**MSE**	**RMSE**	**R2**	**STD**	**ACCURACY**
XGBoost	4,09	16,48	37,27	6,1	0,39	6,59	83,52%
SVR	4,07	14,59	50,81	7,13	0,17	6,0	85,41%
Random Forest	4,17	16,67	41,38	6,43	0,32	6,09	83,33%
SGD	4,46	17,39	48,48	6,96	0,21	6,28	82,61%
ElasticNet	4,53	17,95	48,37	6,95	0,21	6,23	82,05%
Ridge	4,55	18,07	48,91	6,99	0,20	6,29	81,93%
Lasso	4,58	18,06	49,25	7,02	0,19	6,02	81,94%
Linear	4,74	19,01	55,44	7,45	0,09	6,64	80,99%

Επιπλέον, δοκιμάστηκε και υπολογίστηκε το μέσο απόλυτο σφάλμα για κάθε ηλικία. Το μοντέλο εμφάνισε πολύ καλές επιδόσεις για τις μικρές ηλικίες όπου υπήρχαν πολλά περισσότερα δείγματα από τις μεγάλες και έτσι εκπαιδεύτηκε καλύτερα σε αυτές. Πιο συγκεκριμένα, για ηλικίες από 23 έως 26 παρουσίασε απόκλιση μόλις 2 ετών, ενώ γενικά για τους χρήστες από 14 μέχρι 32 ετών είχε ικανοποιητικό μέσο απόλυτο σφάλμα με τη διαφορά μεταξύ της πραγματικής τιμής της ηλικίας και της πρόβλεψης να είναι μικρότερη των 7 ετών. Στο μοντέλο του XGBoost, που ήταν το πιο αποδοτικό, είχε MAE μικρότερο των 10 ετών για τις ηλικίες 13 έως 40 ετών. Παρατηρείται πως το μοντέλου του XGBoost εμφανίζει με διαφορά τις χαμηλότερες αποκλίσεις για τις μεγάλες ηλικίες. Η Εικόνα 8.11 απεικονίζει σε μορφή διαγράμματος το MAE ανά ηλικία με κάθε αλγόριθμο που εξετάστηκε.

Σε αυτό εδώ το σημείο πρέπει να επισημανθεί το γεγονός πως το δείγμα των χρηστών ήταν ανομοιόμορφα κατανεμημένο και ανεπαρκές για ηλικίες άνω των 50 ετών έχοντας το πολύ 10 χρήστες ανά ηλικία. Ειδικότερα, για ηλικίες άνω των 55 υπήρχαν περιπτώσεις χωρίς κανένα δείγμα. Έτσι η σωστή εκπαίδευση του αλγορίθμου για εκείνες τις ηλικίες ήταν αδύνατη με αποτέλεσμα να μην θεωρούνται αξιόπιστες οι αντίστοιχες εκτιμήσεις. Όπως φαίνεται στην Εικόνα 8.11 για αυτές τις ηλικίες υπάρχει μεγάλη απόκλιση τιμών ενώ αντίθετα για τις ηλικίες όπου το δείγμα ήταν ικανοποιητικό το μοντέλο αποδίδει εξαιρετικά ικανοποιητικά και ανιχνεύει την ηλικία με μικρές αποκλίσεις.



Εικόνα 8.11: MAE ανά ηλικία για κάθε αλγόριθμο

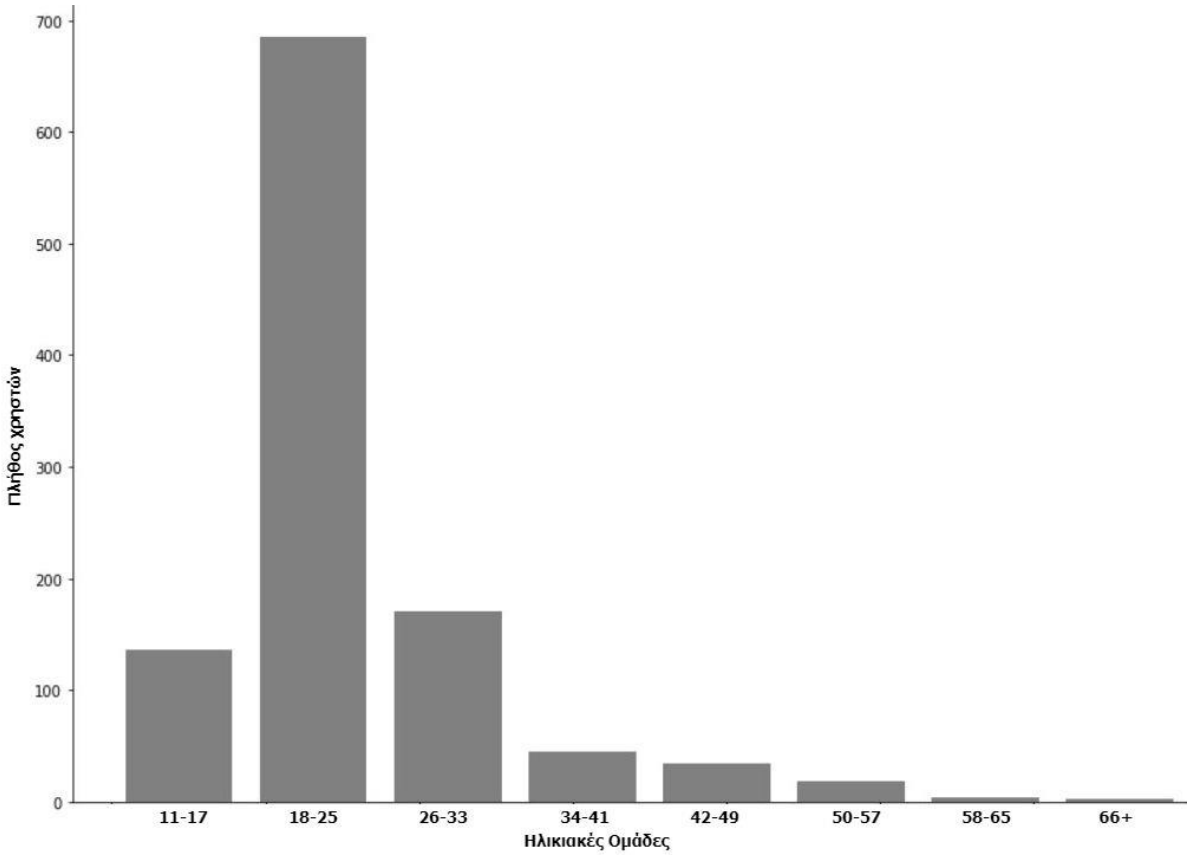
8.3 Ανίχνευση ηλικιακής ομάδας με χρήση αλγορίθμων ταξινόμησης

Για την αξιολόγηση των μοντέλων ταξινόμησης που εκπαιδεύτηκαν έχουν χρησιμοποιηθεί οι μετρικές Accuracy, Precision, Recall και F1-score που εξηγήθηκαν στην ενότητα 3.5.3. Οι μετρικές Precision, Recall και F1-score μπορούν να υπολογιστούν με διαφορετικό τύπο μέσου όρου τιμές ανάλογα με την παράμετρο που θα επιλεχθεί. Τα τρία είδη μέσων όρων είναι ο micro όπου κάνει τον υπολογισμό μετρώντας τα συνολικά true positives, false negatives και false positives, ο macro ο οποίος υπολογίζει για κάθε κλάση το μη σταθμισμένο μέσο όρο χωρίς να λαμβάνει υπόψη την πιθανή ανισορροπία των δεδομένων και ο weighted που βρίσκει για κάθε ομάδα τον σταθμισμένο μέσο όρο της βάσει του αριθμού των δειγμάτων της συνεκτιμώντας έτσι την ανισορροπία των δεδομένων ανά ομάδα. Στην παρούσα εργασία έγινε υπολογισμός και των τριών όπως παρατηρείται στον αντίστοιχο κώδικα. Επίσης, παρουσιάζεται το classification report για κάθε αλγόριθμο που περιλαμβάνει τις ανωτέρω μετρικές και τον αριθμό των δειγμάτων για κάθε κλάση (στήλη support) στο test set.

Τα δεδομένα χωρίστηκαν τυχαία σε 80% δεδομένα εκπαίδευσης (training set) και 20% δεδομένα δοκιμής (test set) για τις ανάγκες της παρούσας διπλωματικής εργασίας. Κατά την εφαρμογή της μεθόδου τέθηκε εξαρτώμενη από την στήλη y (target), που περιέχει τις ηλικιακές ομάδες, η μεταβλητή stratify επειδή το δείγμα δεν έχει ομοιόμορφη κατανομή των χρηστών. Ο κώδικας που εφαρμόζει τη συνάρτηση αυτή φαίνεται παρακάτω:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
                                                    random_state=42,
                                                    stratify=y)
```

Με αυτόν τον τρόπο κάθε κλάση θα αντιπροσωπεύεται στο test set και έτσι αποφεύχθηκε η αύξηση των σφαλμάτων. Ωστόσο, και πάλι ορισμένες κατηγορίες έχουν πολύ λίγες τιμές στο test set κάνοντας έτσι πολύ δύσκολη την εκπαίδευση των αλγορίθμων για την πραγματοποίηση προβλέψεων για αυτές. Το γεγονός αυτό θα επηρεάσει σε μεγάλο βαθμό την απόδοση των αλγορίθμων. Η κατανομή των χρηστών στο test set απεικονίζεται με την μορφή γραφήματος στην Εικόνα 8.12 καθώς και στον πίνακα 8.10 παρακάτω.



Εικόνα 8.12: Κατανομή χρηστών στο test set για την ταξινόμηση

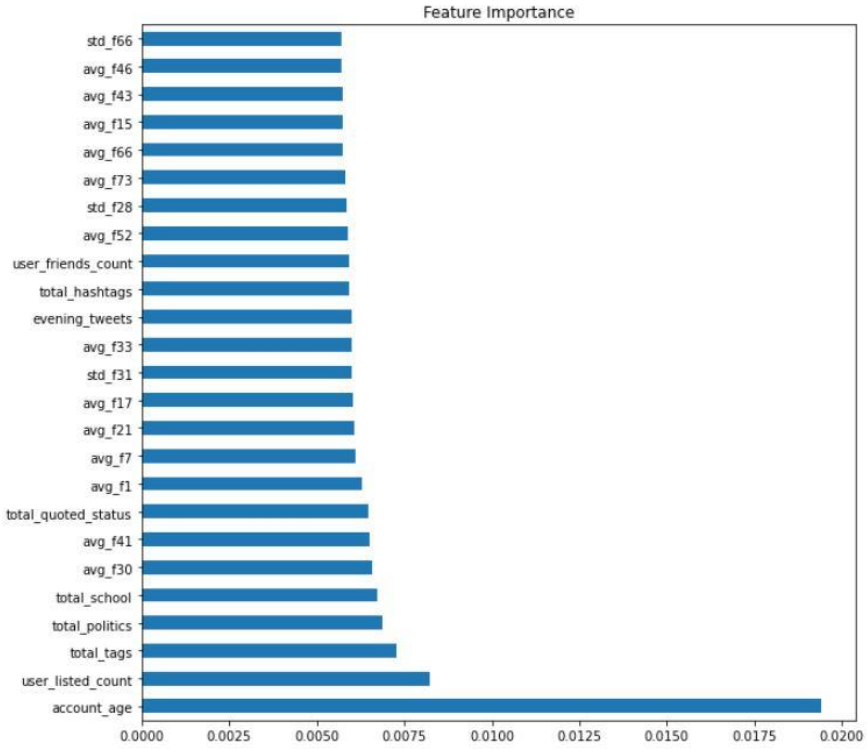
Πίνακας 8.10: Κατανομή χρηστών στο test set για την ταξινόμηση

Δείκτης	Ηλικιακές ομάδες	Χρήστες
1	11-17	136
2	18-25	685
3	26-33	171
4	34-41	45
5	42-49	34
6	50-57	18
7	58-65	4
8	66+	3

Όπως αναφέρθηκε παραπάνω η σπουδαιότητα των χαρακτηριστικών έγινε με χρήση του αλγορίθμου ταξινόμησης τυχαίων δέντρων (Extremely Randomized Trees Classifier ή Extra Trees Classifier). Υπολόγισε τα 25 σημαντικότερα features για την είσοδο στους αλγορίθμους ταξινόμησης και το πιο σπουδαίο με διαφορά αναδείχθηκε το “account_age”, ενώ δεύτερο ήρθε το “user_listed_count”. Σημαντικό ρόλο αποτελούν τα features σχετικά με τα topic των tweets

που δημοσιεύει ο χρήστης καθώς ξεχωρίζουν τα “total_politics” και “total_school”. Επίσης ο αριθμός των friends (followings) συμβάλλουν στην εκμάθηση. Ακόμη, ορισμένα από τα πιο σημαντικά χαρακτηριστικά είναι κάποια από τις αριθμητικές τιμές που προέκυψαν για τα tweets όπως περιγράφηκε στην ενότητα 7.6. Ο κώδικας που εφαρμόζει τη μέθοδο και απεικονίζει τα αποτελέσματα της σε γράφημα παρουσιάζεται παρακάτω, ενώ στην Εικόνα 8.13 παρουσιάζεται το αντίστοιχο γράφημα που δημιουργείται.

```
from sklearn.ensemble import ExtraTreesClassifier
model = ExtraTreesClassifier(random_state=42)
model.fit(X_in,y)
print(model.feature_importances_)
#plot graph of feature importances for better visualization
feat_importances = pd.Series(model.feature_importances_,
                             index=X_in.columns)
feat_importances.nlargest(25).plot(kind='barh')
plt.gcf().set_size_inches(10, 10)
plt.title("Feature Importance")
plt.show()
```



Εικόνα 8.13: Σπουδαιότητα χαρακτηριστικών για την ταξινόμηση

Για τον υπολογισμό των μετρικών για κάθε μοντέλο ταξινόμησης έγινε χρήση της παρακάτω συνάρτησης με την οποία απεικονίζεται το συνολικό classification report και ο confusion matrix. Έχουν υπολογιστεί οι:

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
```

```
from sklearn.metrics import classification_report, confusion_matrix

def evaluation_metrics(model, model_name, y_pred, y_test):
    # accuracy
    acc = accuracy_score(y_test, y_pred)
    # precision
    macro_precision = precision_score(y_test, y_pred, average='macro')
    micro_precision = precision_score(y_test, y_pred, average='micro')
    weighted_precision = precision_score(y_test, y_pred, average='weighted')
    precision = precision_score(y_test, y_pred, average=None)
    # recall
    macro_recall = recall_score(y_test, y_pred, average='macro')
    micro_recall = recall_score(y_test, y_pred, average='micro')
    weighted_recall = recall_score(y_test, y_pred, average='weighted')
    recall = recall_score(y_test, y_pred, average=None)
    # f1-score
    macro_f1 = f1_score(y_test, y_pred, average='macro')
    micro_f1 = f1_score(y_test, y_pred, average='micro')
    weighted_f1 = f1_score(y_test, y_pred, average='weighted')
    f1 = f1_score(y_test, y_pred, average=None)
    # logs
    print("Model:", model)
    print("Accuracy:", acc*100.0)
    print("MACRO PRECISION:", macro_precision)
    print("MICRO PRECISION:", micro_precision)
    print("WEIGHTED PRECISION:", weighted_precision)
    print("PRECISION:", precision)
    print("MACRO RECALL:", macro_recall)
    print("MICRO RECALL:", micro_recall)
    print("WEIGHTED RECALL:", weighted_recall)
    print("RECALL:", recall)
    print("MACRO F1:", macro_f1)
    print("MICRO F1:", micro_f1)
    print("WEIGHTED F1:", weighted_f1)
    print("F1:", f1)
    print("Classification Report:")
    print(metrics.classification_report(y_test, y_pred))
    print("Confusion Matrix:")
    print(confusion_matrix(y_test, y_pred))
```

Για την αποθήκευση των μοντέλων και των παραγόμενων τιμών από τις προβλέψεις μέσω ταξινόμησης χρησιμοποιήθηκε η ίδια συνάρτηση που περιγράφηκε και για την περίπτωση της παλινδρόμησης στην ενότητα 8.2 και έχει ως παραμέτρους εισόδου το test set (y_test) και τις προβλέψεις (y_pred) του αλγορίθμου.

Για την απεικόνιση των αποτελεσμάτων κάθε αλγορίθμου ταξινόμησης σε μορφή διαγράμματος αναπτύχθηκαν μία συνάρτηση που απεικονίζει σε κοινό γράφημα το test set, με την ετικέτα test και πράσινο χρώμα, και τις προβλέψεις, με την ετικέτα pred και μπλε χρώμα. Ο κώδικας που την υλοποιεί είναι ο εξής:

```
from mathplotlib import pyplot
def doublePlotFunction(y_pred, y_test):
```

```
    pyplot.hist(y_pred, bins=8, range=[1, 8], color='blue',
        rwidth=0.8, align='mid', alpha=0.5, label='pred')
    pyplot.hist(y_test, bins=8, range=[1, 8], color='green',
        rwidth=0.8, align='mid', alpha=0.5, label='test')
    pyplot.legend(loc='upper right')
    plt.title("Predictions-Test plot")
    pyplot.gcf().set_size_inches(14, 10)
    pyplot.show()
```

8.3.1 Ταξινόμηση XGBoost

Για την επιλογή του βέλτιστου μοντέλου του XGBoost εφαρμόστηκε η τεχνική της βελτίωσης υπερπαραμέτρων μέσω του αλγορίθμου RandomizedSearchCV. Ο αλγόριθμος αυτός πραγματοποίησε την αναζήτηση για συγκεκριμένες παραμέτρους και τις αντίστοιχες τιμές τους όπως αυτές ορίστηκαν στον κώδικα. Η διεργασία αυτή διήρκησε 15 λεπτά και 33 δευτερόλεπτα μέχρι να βρεθεί το πιο αποδοτικό μοντέλο. Η εκπαίδευση του μοντέλου, η παρουσίαση και η αποθήκευση των αποτελεσμάτων του ήταν μία γρήγορη διαδικασία και χρειάστηκαν περίπου 29 δευτερόλεπτα για να ολοκληρωθεί. Ο κώδικας που εφαρμόζει τον XGBoost φαίνεται στη συνέχεια.

```
import xgboost is xgb
from xgboost import XGBClassifier
xgboost = XGBClassifier()
# fit intercept values
booster = ['gbtree', 'gblinear', 'dart']
# gamma
gamma = [0.01, 0.1, 0, 1, 2, 5, 10, 20, 50]
# normalize values
n_estimators = [10, 20, 50, 100, 200, 500, 1000]
# max_depth
max_depth = [2, 3, 4, 5, 6, 10, 16, 20, 32, 48, 54]
# alpha values
alpha = [0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1, 1, 10, 100]
# eta0 values
eta0 = [0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1]
# min_child_weight
min_child_weight = [1, 2, 4, 8 ,10]
# Create the random grid
random_grid = {'n_estimators': n_estimators, 'gamma': gamma,
               'alpha': alpha, 'max_depth': max_depth,
               'eta0': eta0, 'min_child_weight': min_child_weight,
               'booster': booster}
#perform hyperparameter tuning
xgboost_random = RandomizedSearchCV(estimator = xgboost,
                                    param_distributions = random_grid,
                                    n_iter = 5, cv = 5, verbose=2,
                                    random_state=42, n_jobs = -1)
xgboost_random.fit(X_train, y_train)
#best estimator model from randomsearch process
best_model = xgboost_random.best_estimator_
best_model.fit(X_train,y_train)
```

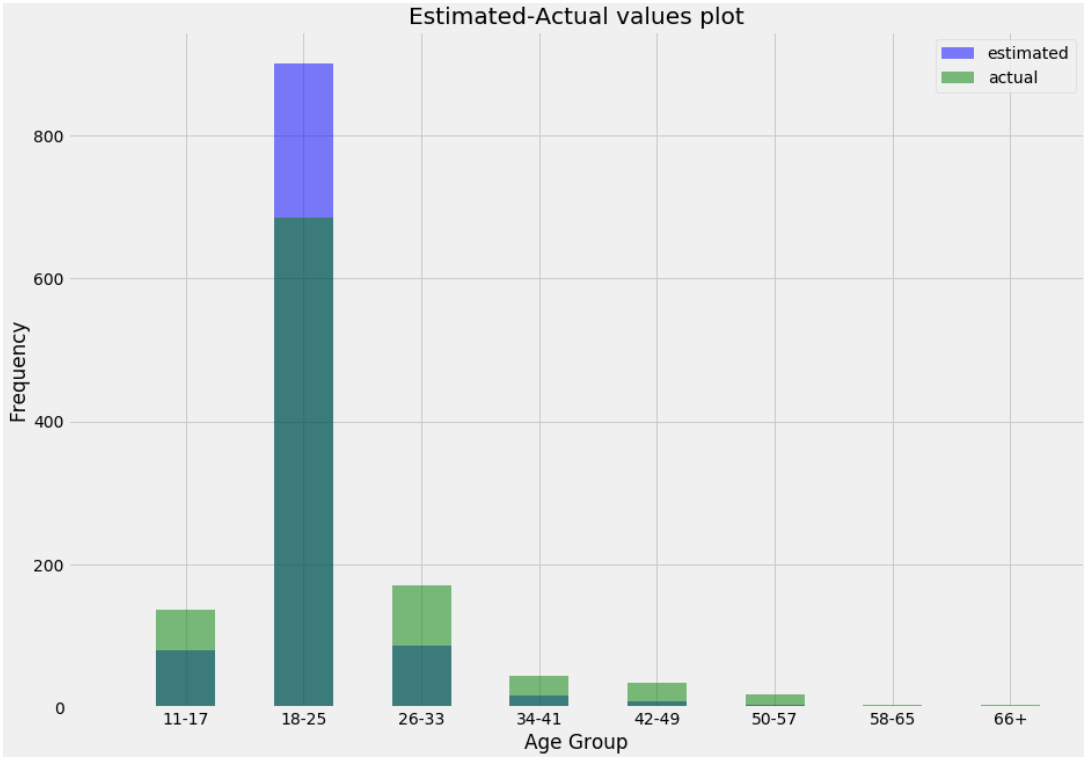
```
#Evaluate
model_name = 'XGBoost'
y_xgboost_test = best_model.predict(X_test)
evaluation_metrics(best_model, model_name, y_xgboost_test, y_test)
save_res(best_model, model_name, y_xgboost_test, y_test)
plotFunction(y_xgboost_test, "XGBoost predictions")
doublePlotFunction(y_xgboost_test, y_test)
```

Στον πίνακα 8.11 φαίνεται ο confusion matrix του μοντέλου του αλγορίθμου XGBoost, όπου με πράσινο χρώμα έχουν μαρκαριστεί τα κελιά της διαγωνίου που είναι οι TP προβλέψεις. Στις δύο μεγαλύτερες ηλικιακές ομάδες δεν πραγματοποιεί κάποια σωστή πρόβλεψη.

Πίνακας 8.11: Ο Confusion matrix του XGBoost

		Πραγματικές τιμές (actual)							
		11-17	**18-25**	**26-33**	**34-41**	**42-49**	**50-57**	**58-65**	**66+**
Εκτιμήσεις (estimations)	**11-17**	57	15	3	2	3	0	0	0
	18-25	78	648	113	28	18	11	4	1
	26-33	1	18	51	6	5	5	0	0
	34-41	0	3	3	7	1	0	0	2
	42-49	0	0	1	2	5	1	0	0
	50-57	0	1	0	0	2	1	0	0
	58-65	0	0	0	0	0	0	0	0
	66+	0	0	0	0	0	0	0	0

Το διάγραμμα πραγματικών τιμών και εκτιμήσεων είναι:



Εικόνα 8.14: Διάγραμμα εκτιμήσεων-πραγματικών τιμών για την ταξινόμηση XGBoost

Το classification report περιλαμβάνει συνοπτικά τα αποτελέσματα που κατέγραψε για τις μετρικές το βέλτιστο μοντέλο ταξινόμησης του αλγορίθμου XGBoost είναι:

Age Groups	precision	recall	f1-score	support
11-17	0.71	0.42	0.53	136
18-25	0.72	0.95	0.82	685
26-33	0.59	0.30	0.40	171
34-41	0.44	0.16	0.23	45
42-49	0.56	0.15	0.23	34
50-57	0.25	0.06	0.09	18
58-65	0.00	0.00	0.00	4
66+	0.00	0.00	0.00	3
accuracy			0.70	1096
macro avg	0.41	0.25	0.29	1096
weighted avg	0.67	0.70	0.66	1096

Εικόνα 8.15: Αποτελέσματα αξιολόγησης ταξινόμησης XGBoost

Παρατηρώντας τα αποτελέσματα βλέπουμε ότι ο αλγόριθμος XGBoost επιτυγχάνει συνολικά accuracy 70%, precision 0,67, recall 0,70 και f1-score 0,66. Επίσης, το μοντέλο σημειώνει μεγάλες τιμές precision ίσες με 0,71 και 0,72 για τις ηλικιακές ομάδες 11-17 και 18-25 αντίστοιχα. Η υψηλότερη τιμή για το recall είναι 0,95 και για το f1-score 0,82 και εμφανίζονται στην ηλικιακή ομάδα 18-25. Στις μεγάλες ηλικίες 58-65 και 66+, όπου το test set περιλαμβάνει ελάχιστες εμφανίσεις γι αυτές, το μοντέλο παρουσιάζει μηδενικά αποτελέσματα για τις μετρικές precision, recall και f1-score αδυνατώντας να κάνει σωστές προβλέψεις λόγω του μικρού δείγματος. Ακόμη, από το διάγραμμα πραγματικών τιμών και προβλέψεων καθώς και από τα δείγματα κάθε κλάσης στο classification report, διαπιστώνουμε ότι η 2^{η} κατηγορία καταλαμβάνει συντριπτικά το μεγαλύτερο μέρος των δειγμάτων. Αυτό είναι λογικό διότι και στο αρχικό δείγμα εισόδου οι ηλικίες 18-25 είχαν την μεγαλύτερη εκπροσώπηση. Έτσι ο αλγόριθμος εκπαιδεύεται πολύ καλά για αυτήν την ομάδα και πετυχαίνει τις καλύτερες επιδόσεις.

8.3.2 Ταξινόμηση με Λογιστική Παλινδρόμηση (Logistic Regression)

Η επιλογή του βέλτιστου μοντέλου για τη Λογιστική Παλινδρόμηση πραγματοποιήθηκε με την τεχνική της βελτίωσης υπερπαραμέτρων μέσω του αλγορίθμου RandomizedSearchCV. Η αναζήτηση έγινε για συγκεκριμένες παραμέτρους και τιμές τους που δηλώθηκαν ορίστικαν στον κώδικα. Η διάρκεια της διαδικασίας αυτής ήταν περίπου 6 λεπτά και 34 δευτερόλεπτα με σκοπό να βρεθεί το καλύτερο μοντέλο. Η εκπαίδευση του αλγορίθμου, η παρουσίαση καθώς και η αποθήκευση των αποτελεσμάτων ήταν μία διεργασία που χρειάστηκε περίπου 9 δευτερόλεπτα για να ολοκληρωθεί. Ο κώδικας που υλοποίησε τη λογιστική παλινδρόμηση είναι ο εξής.

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
# fit intercept values
penalty = ['l1', 'l2', 'elasticnet', 'none']
```

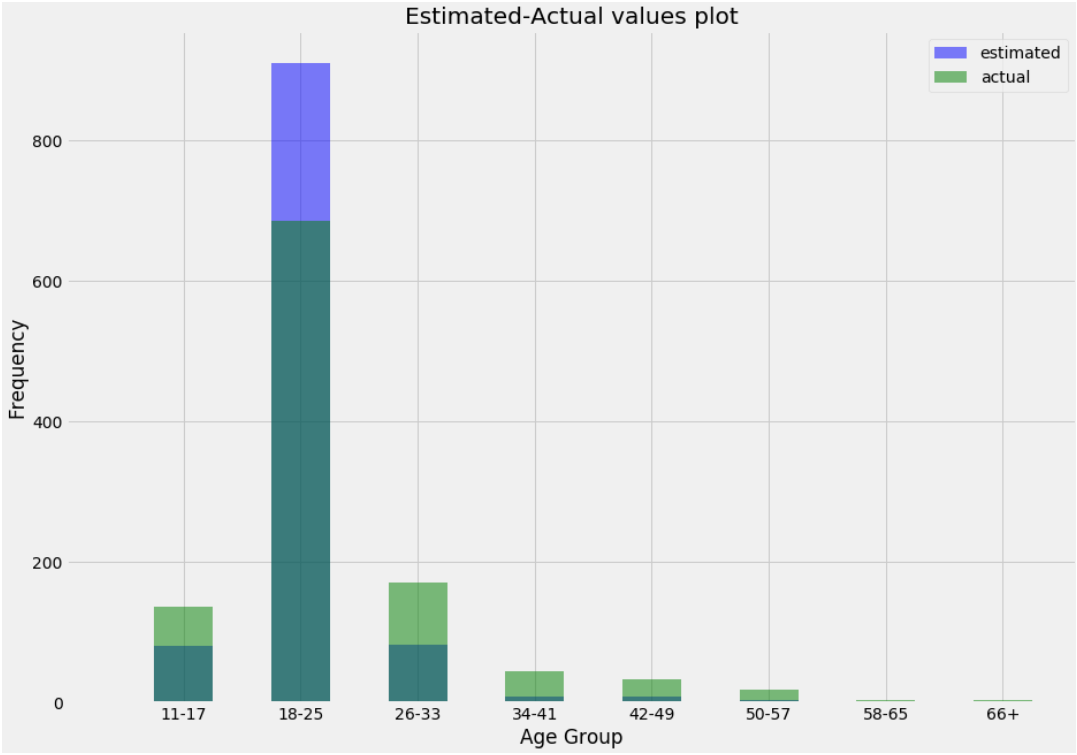
```
# gamma
C = [0.01, 0.1, 0, 1, 2, 5, 10, 20, 50]
# normalize values
solver = ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga']
# max_iter
max_iter = [10, 20, 50, 100, 200, 500, 1000]
# Create the random grid
random_grid = {'penalty': penalty,
               'C': C,
               'max_iter': max_iter,
               'solver': solver}
lr_random = RandomizedSearchCV(estimator = lr,
                               param_distributions = random_grid,
                               n_iter = 5,
                               cv = 5,
                               verbose=2,
                               random_state=42,
                               n_jobs = -1)
lr_random.fit(X_train, y_train)
#best estimator model from randomsearch process
best_model = lr_random.best_estimator_
best_model.fit(X_train,y_train)
#Evaluate
model_name = 'Logistic Regression'
y_lr_test = best_model.predict(X_test)
evaluation_metrics(best_model, model_name, y_lr_test, y_test)
save_res(best_model, model_name, y_lr_test, y_test)
plotFunction(y_lr_test, "Logistic Regression predictions")
doublePlotFunction(y_lr_test, y_test)
```

Ο Πίνακας 8.12 παρακάτω παρουσιάζει τον confusion matrix του αλγορίθμου Λογιστικής Παλινδρόμησης, όπου με πράσινο χρώμα έχουν επισημανθεί όλα τα κελιά της διαγωνίου που αποτελούν τις TP προβλέψεις. Παρατηρείται πως για τις 3 μεγαλύτερες ηλικιακές ομάδες δεν καταφέρνει να πραγματοποιήσει κάποια επιτυχημένη πρόβλεψη. Επίσης, πολύ λίγες είναι οι προβλέψεις του μοντέλου για τις ηλικιακές κατηγορίες 34-41 και 42-49. Αντίθετα πραγματοποιεί αρκετές προβλέψεις στις 3 ομάδες για τις ηλικίες των νεότερων χρηστών του δείγματος.

Πίνακας 8.12: Ο Confusion matrix της Λογιστικής Παλινδρόμησης

		Πραγματικές τιμές (actual)							
		11-17	**18-25**	**26-33**	**34-41**	**42-49**	**50-57**	**58-65**	**66+**
Εκτιμήσεις (estimations)	**11-17**	48	23	5	2	1	0	0	1
	18-25	87	605	143	38	23	10	2	2
	26-33	1	42	21	3	6	7	2	0
	34-41	0	7	1	1	0	0	0	0
	42-49	0	4	1	1	2	1	0	0
	50-57	0	3	0	0	1	0	0	0
	58-65	0	0	0	0	0	0	0	0
	66+	0	1	0	0	1	0	0	0

Το διάγραμμα πραγματικών τιμών και εκτιμήσεων φαίνεται στη συνέχεια:



Εικόνα 8.16: Διάγραμμα εκτιμήσεων-πραγματικών τιμών για την ταξινόμηση με Λογιστική Παλινδρόμηση

Το classification report που παρουσιάζεται παρακάτω αφορά το βέλτιστο μοντέλο ταξινόμησης που προέκυψε από τον αλγόριθμο της λογιστικής παλινδρόμησης:

Classification Report:

Age Groups	precision	recall	f1-score	support
11-17	0.60	0.35	0.44	136
18-25	0.66	0.88	0.76	685
26-33	0.26	0.12	0.17	171
34-41	0.11	0.02	0.04	45
42-49	0.22	0.06	0.09	34
50-57	0.00	0.00	0.00	18
58-65	0.00	0.00	0.00	4
66+	0.00	0.00	0.00	3
accuracy			0.62	1096
macro avg	0.23	0.18	0.19	1096
weighted avg	0.54	0.62	0.56	1096

Εικόνα 8.17: Αποτελέσματα αξιολόγησης ταξινόμησης με τη Λογιστική Παλινδρόμηση

Κοιτώντας τις επιδόσεις της λογιστικής παλινδρόμησης παρατηρούμε ότι καταγράφει συνολικά accuracy 62%, precision 0,54, recall 0,62 και f1-score 0,56. Ακόμη, εμφανίζει τις καλύτερες τιμές για τη μετρική precision ίσες με 0,60 και 0,66 στις ηλικιακές ομάδες 11-17 και 18-25 αντίστοιχα. Οι υψηλότερες τιμές για το recall είναι 0,88 και για το f1-score 0,76 και αφορούν την ηλικιακή ομάδα 18-25. Στις τρεις μεγάλες ηλικιακές ομάδες, όπου το test set

περιλαμβάνει ελάχιστες εμφανίσεις για αυτές, το μοντέλο έχει μηδενικά αποτελέσματα για τις όλες τις μετρικές και αποτυγχάνει προβλέψει σωστά λόγω του μικρού δείγματος. Επίσης, η 2η ομάδα έχει τις περισσότερες εμφανίσεις στο δείγμα και στις επιτυχημένες προβλέψεις όπως φαίνεται στο διάγραμμα αληθινών τιμών και προβλέψεων αλλά και στα δείγματα των κλάσεων στο classification report.

8.3.3 Ταξινόμηση με SVC

Η εύρεση του βέλτιστου μοντέλου ταξινόμησης με SVC πραγματοποιήθηκε μέσω της μεθόδου βελτίωσης υπερπαραμέτρων με τον αλγόριθμο RandomizedSearchCV. Η αναζήτηση έγινε για συγκεκριμένες παραμέτρους και τιμές που είχαν οριστεί στον κώδικα. Η διεργασία αυτή διήρκησε 1 λεπτό και 12 δευτερόλεπτα ώστε να βρεθεί το καλύτερο μοντέλο. Η διεργασία εκπαίδευσης του μοντέλου, παρουσίασης και αποθήκευσης του ήταν σύντομη και χρειάστηκαν περίπου 8 δευτερόλεπτα για να ολοκληρωθεί. Ο κώδικας για την εφαρμογή του SVC είναι ο εξής.

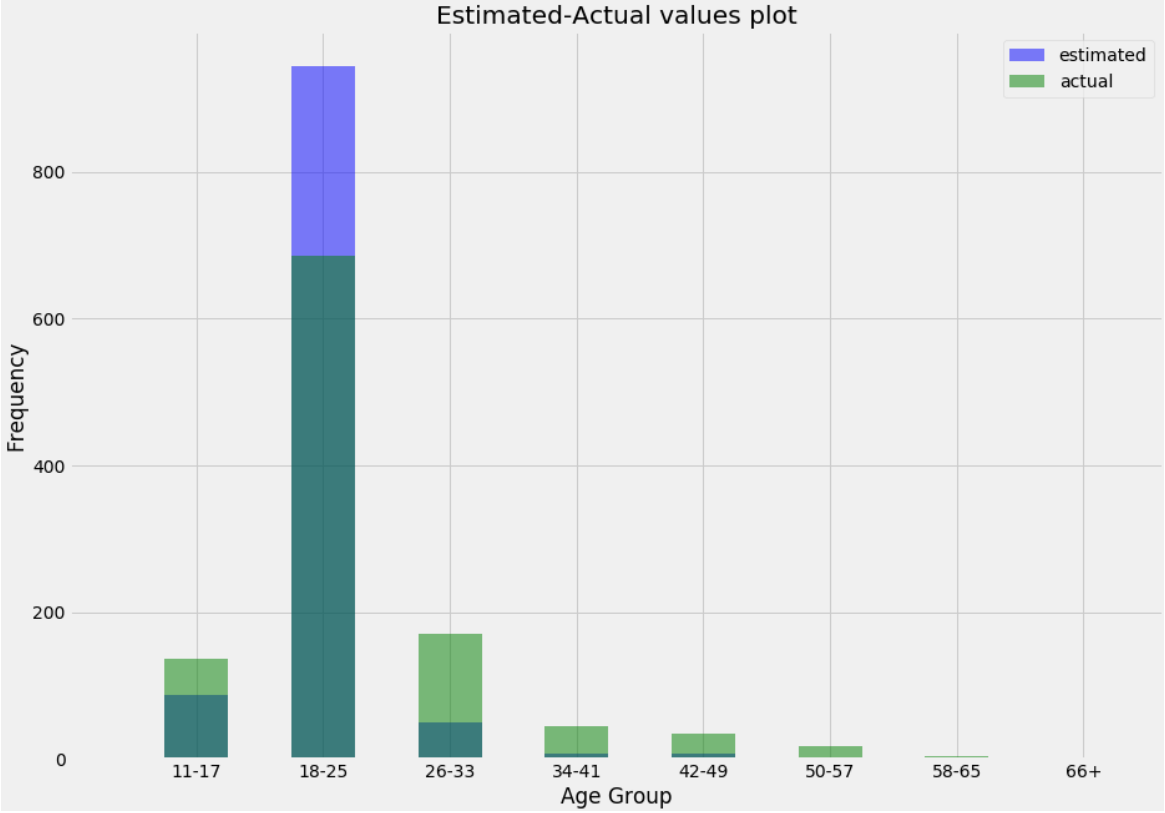
```
from sklearn import svm
from sklearn.svm import SVC
svc = SVC()
# C values
C = [0.001, 0.01, 0.1, 1, 2, 5, 10, 100]
# kernel values
kernel = ['linear', 'poly', 'rbf', 'sigmoid', 'precomputed']
# gamma
gamma = [0.1, 0.2 , 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 10]
# Create the random grid
random_grid = {'C': C,
               'gamma': gamma,
               'kernel': kernel}
svc_random = RandomizedSearchCV(estimator = svc,
                                param_distributions = random_grid,
                                n_iter = 5, cv = 5,
                                verbose=2,
                                random_state=42,
                                n_jobs = -1)
svc_random.fit(X_train, y_train)
#best estimator model from randomsearch process
best_model = svc_random.best_estimator_
best_model.fit(X_train,y_train)
#Evaluate
model_name = 'SVC'
y_svc_test = best_model.predict(X_test)
evaluation_metrics(best_model, model_name, y_svc_test, y_test)
save_res(best_model, model_name, y_svc_test, y_test)
plotFunction(y_svc_test, "SVC predictions")
doublePlotFunction(y_svc_test, y_test)
```

Ο confusion matrix της ταξινόμησης SVC δείχνεται στον πίνακα 8.13 παρακάτω, όπου με πράσινο χρώμα έχουν σημειωθεί οι τιμές της διαγωνίου για τις TP προβλέψεις. Και σε αυτή την λύση παρατηρείται ότι για τις 3 μεγάλες ηλικιακές κλάσεις 50-57, 58-65 και 66+ ο αλγόριθμος αδυνατεί να κάνει προβλέψεις.

Πίνακας 8.13: Ο Confusion matrix του SVC

		Πραγματικές τιμές (actual)							
		11-17	**18-25**	**26-33**	**34-41**	**42-49**	**50-57**	**58-65**	**66+**
Εκτιμήσεις (estimations)	**11-17**	61	19	5	1	0	0	0	2
	18-25	75	644	142	39	27	13	3	1
	26-33	0	19	21	1	4	3	1	0
	34-41	0	2	1	4	0	0	0	0
	42-49	0	1	2	0	2	2	0	0
	50-57	0	0	0	0	1	0	0	0
	58-65	0	0	0	0	0	0	0	0
	66+	0	0	0	0	0	0	0	0

Το διάγραμμα των εκτιμήσεων σε σχέση με τις πραγματικές τιμές είναι το εξής:



Εικόνα 8.18: Διάγραμμα εκτιμήσεων-πραγματικών τιμών για την ταξινόμηση με SVC

Το classification report για το βέλτιστο μοντέλο ταξινόμησης του αλγορίθμου SVC που αποσκοπεί να προβλέψει την ηλικιακή ομάδα που ανήκουν οι χρήστες του Twitter φαίνεται στην παρακάτω εικόνα:

Classification Report:

Age Groups	precision	recall	f1-score	support
11-17	0.69	0.45	0.54	136
18-25	0.68	0.94	0.79	685
26-33	0.43	0.12	0.19	171
34-41	0.57	0.09	0.15	45
42-49	0.29	0.06	0.10	34
50-57	0.00	0.00	0.00	18
58-65	0.00	0.00	0.00	4
66+	0.00	0.00	0.00	3
accuracy			0.67	1096
macro avg	0.33	0.21	0.22	1096
weighted avg	0.61	0.67	0.60	1096

Εικόνα 8.19: Αποτελέσματα αξιολόγησης ταξινόμησης SVC

Στα αποτελέσματα του μοντέλου ταξινόμησης με SVC που φαίνονται στο classification report καταγράφονται συνολικά accuracy 67%, precision 0,61, recall 0,67 και f1-score 0,60. Το μοντέλο έχει επίσης μεγάλες τιμές για precision ίσες με 0,69 και 0,68 για τις 2 μικρότερες ηλικιακές ομάδες. Για τις μετρικές recall και f1-score 0,82 εμφανίζει καλύτερες τιμές τις 0,94 και 0,79 αντίστοιχα στην ηλικιακή ομάδα 18-25 ετών. Οι μεγάλες ηλικίες 50-57, 58-65 και 66+ δυσκολεύουν το μοντέλο να πραγματοποιήσει προβλέψεις και βλέπουμε οτι παρουσιάζει μηδενικά αποτελέσματα για precision, recall και f1-score. Γενικά, καταφέρνει τις καλύτερες επιδόσεις στη $2^{η}$ ηλικιακή κατηγορία που αποτελεί το μεγαλύτερο μέρος των δειγμάτων, όπως και οι υπόλοιποι αλγόριθμοι. Αυτό φαίνεται και από το διάγραμμα πραγματικών τιμών και προβλέψεων καθώς και από το classification report.

8.3.4 Ταξινόμηση με Δέντρα Απόφασης (Decision Trees)

Για την επιλογή του βέλτιστου μοντέλου ταξινόμησης με Δέντρα απόφασης εφαρμόστηκε η τεχνική της βελτίωσης υπερπαραμέτρων με τον αλγόριθμο RandomizedSearchCV. Αναζητήθηκαν οι πιο αποδοτικές παράμετροι μέσα από ένα μεγάλο σύνολο που ορίστηκε στον κώδικα. Ήταν μία γρήγορη διεργασία που διήρκησε 0,4 δευτερόλεπτα. Όσον αφορά την εκπαίδευση, την παρουσίαση και την αποθήκευση των αποτελεσμάτων του μοντέλου ήταν και αυτή μία ταχύτατη διαδικασία που ολοκληρώθηκε περίπου σε 0,9 δευτερόλεπτα. Ο παρακάτω κώδικας εφαρμόζει την ταξινόμηση με Δέντρα Απόφασης.

```
from sklearn.tree import DecisionTreeClassifier
decTree = DecisionTreeClassifier()

# fit intercept values
criterion = ['gini', 'entropy']
# normalize values
max_features = ['auto', 'sqrt', 'log2']
# max_depth
max_depth = [2, 3, 4, 5, 6, 10, 16, 20, 32, 48, 54]
# min_child_weight
min_samples_leaf = [1, 2, 4, 8 ,10]
```

```
# Create the random grid
random_grid = {'max_depth': max_depth,
               'max_features': max_features,
               'min_samples_leaf': min_samples_leaf,
               'criterion': criterion}
decTree_random = RandomizedSearchCV(estimator = decTree,
                                    param_distributions = random_grid,
                                    n_iter = 5, cv = 5, verbose=2,
                                    random_state=42, n_jobs = -1)
decTree_random.fit(X_train, y_train)
#best estimator model from randomsearch process
best_model = decTree_random.best_estimator_
best_model.fit(X_train,y_train)
#Evaluate
model_name = 'Decision Tree'
y_decTree_test = best_model.predict(X_test)
evaluation_metrics(best_model, model_name, y_decTree_test, y_test)
save_res(best_model, model_name, y_decTree_test, y_test)
plotFunction(y_decTree_test, "Decision Tree predictions")
doublePlotFunction(y_decTree_test, y_test)
```

Στον πίνακα 8.14 απεικονίζεται ο confusion matrix του μοντέλου που παράχθηκε από την εκτέλεση της ταξινόμησης με Δέντρα Απόφασης, όπου με πράσινο χρώμα έχουν μαρκαριστεί τα κελιά της διαγωνίου που είναι οι TP προβλέψεις. Παρατηρούμε ότι κάνει επιτυχημένες προβλέψεις μόνο για την ηλικιακή ομάδα 18-25 αλλά και προβλέπει μόνο αυτή. Αντίθετα για τις υπόλοιπες ηλικιακές κλάσεις δεν πραγματοποιεί καμία πρόβλεψη.

Πίνακας 8.14: Ο Confusion matrix των Δέντρων Απόφασης

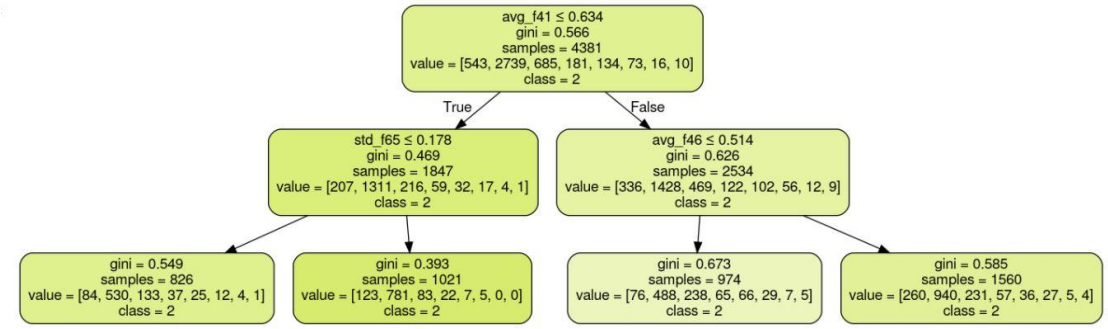
		Πραγματικές τιμές (actual)							
		11-17	**18-25**	**26-33**	**34-41**	**42-49**	**50-57**	**58-65**	**66+**
Εκτιμήσεις (estimations)	**11-17**	0	0	0	0	0	0	0	0
	18-25	136	685	171	45	34	18	4	3
	26-33	0	0	0	0	0	0	0	0
	34-41	0	0	0	0	0	0	0	0
	42-49	0	0	0	0	0	0	0	0
	50-57	0	0	0	0	0	0	0	0
	58-65	0	0	0	0	0	0	0	0
	66+	0	0	0	0	0	0	0	0

Ο κώδικας που σχεδιάζει το αντίστοιχο δέντρο φαίνεται παρακάτω, ενώ για την ηλικιακή ομάδα 18-25 έχει δοθεί η ετικέτα 2.

```
from sklearn.externals.six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus
## make list of target values
arr = y_test.values
list_l = arr.tolist()
mylist = list(dict.fromkeys(list_l))
```

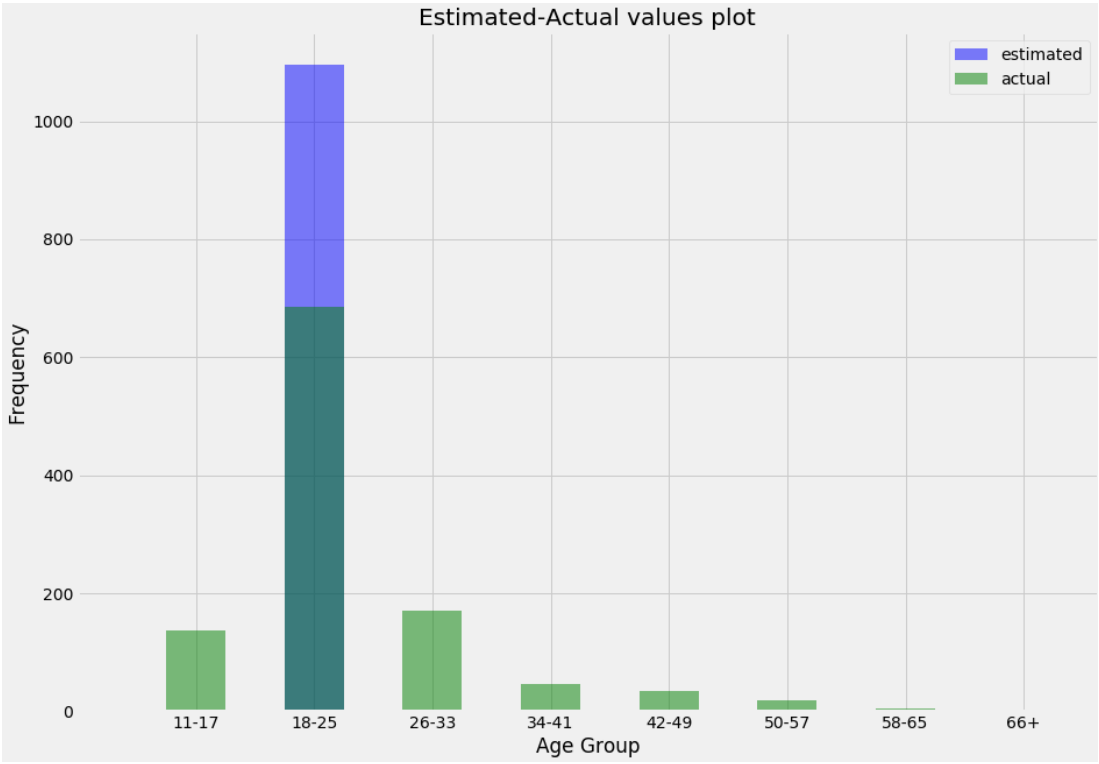
```
numbers_str = [str(i) for i in mylist]
dot_data = StringIO()
export_graphviz(best_model, out_file=dot_data,
                filled=True, rounded=True,
                special_characters=True, feature_names = X_in.columns.values,
                class_names=numbers_str)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png('dec_tree_model.png')
Image(graph.create_png())
```

Το δέντρο που δημιουργήθηκε απεικονίζεται στην Εικόνα 8.20:



Εικόνα 8.20: Τελικό Δέντρο Απόφασης για την ταξινόμηση

Το διάγραμμα αληθινών τιμών σε σχέση με τις εκτιμήσεις είναι:



Εικόνα 8.21: Διάγραμμα εκτιμήσεων-πραγματικών τιμών για την ταξινόμηση με Δέντρα Απόφασης

Το classification report για το βέλτιστο μοντέλο ταξινόμησης με δέντρα απόφασης φαίνεται παρακάτω:

Classification Report:

Age Groups	precision	recall	f1-score	support
11-17	0.00	0.00	0.00	136
18-25	0.62	1.00	0.77	685
26-33	0.00	0.00	0.00	171
34-41	0.00	0.00	0.00	45
42-49	0.00	0.00	0.00	34
50-57	0.00	0.00	0.00	18
58-65	0.00	0.00	0.00	4
66+	0.00	0.00	0.00	3
accuracy			0.62	1096
macro avg	0.08	0.12	0.10	1096
weighted avg	0.39	0.62	0.48	1096

Εικόνα 8.22: Αποτελέσματα αξιολόγησης ταξινόμησης με Δέντρα Απόφασης

Το μοντέλο ταξινόμησης με τα δέντρα απόφασης παρουσιάζει συνολικά accuracy 62%, precision 0,39, recall 0,62 και f1-score 0,48. Το μοντέλο πραγματοποιεί προβλέψεις μόνο για την ηλικιακή ομάδα 18-25 αποτυγχάνοντας για τις υπόλοιπες. Αυτό το γεγονός εξηγεί τη χαμηλή επίδοση που σημειώνει στο σύνολο των μετρικών μεθόδων και το θέτει αναξιόπιστο και ακατάλληλο για την διεξαγωγή της πρόβλεψης της ηλικιακής ομάδας των χρηστών του Twitter.

8.3.5 Ταξινόμηση KNN

Η επιλογή του βέλτιστου μοντέλου για τον αλγόριθμο KNN πραγματοποιήθηκε με τη μέθοδο της βελτίωσης υπερπαραμέτρων εφαρμόζοντας τον αλγόριθμο RandomizedSearchCV. Έγινε αναζήτηση για ένα σύνολο παραμέτρων και των αντίστοιχων τιμών τους ώστε να βρεθεί η καλύτερη λύση, εστιάζοντας στο πλήθος των γειτόνων όπου τελικά επιλέχθηκε να είναι 6. Η διαδικασία ήταν σύντομη και διήρκησε περίπου 13 δευτερόλεπτα. Η εκπαίδευση του μοντέλου, η παρουσίαση και η αποθήκευση των αποτελεσμάτων του εκτελέστηκαν σε περίπου 2,8 δευτερόλεπτα. Ο κώδικας που εφαρμόζει τον KNN φαίνεται στη συνέχεια.

```
from sklearn.neighbors import KneighborsClassifier
knn = KNeighborsClassifier()
# number of neighbors
n_neighbors = [1, 2, 3, 4, 5, 6 ,7, 8, 9, 10, 11, 12]
# normalize values
weights= ['uniform', 'distance']
# Create the random grid
random_grid = {'n_neighbors': n_neighbors, 'weights': weights}
knn_random = RandomizedSearchCV(estimator = knn,
                                param_distributions = random_grid,
                                n_iter = 5, cv = 5, verbose=2,
                                random_state=42, n_jobs = -1)
knn_random.fit(X_train, y_train)
```

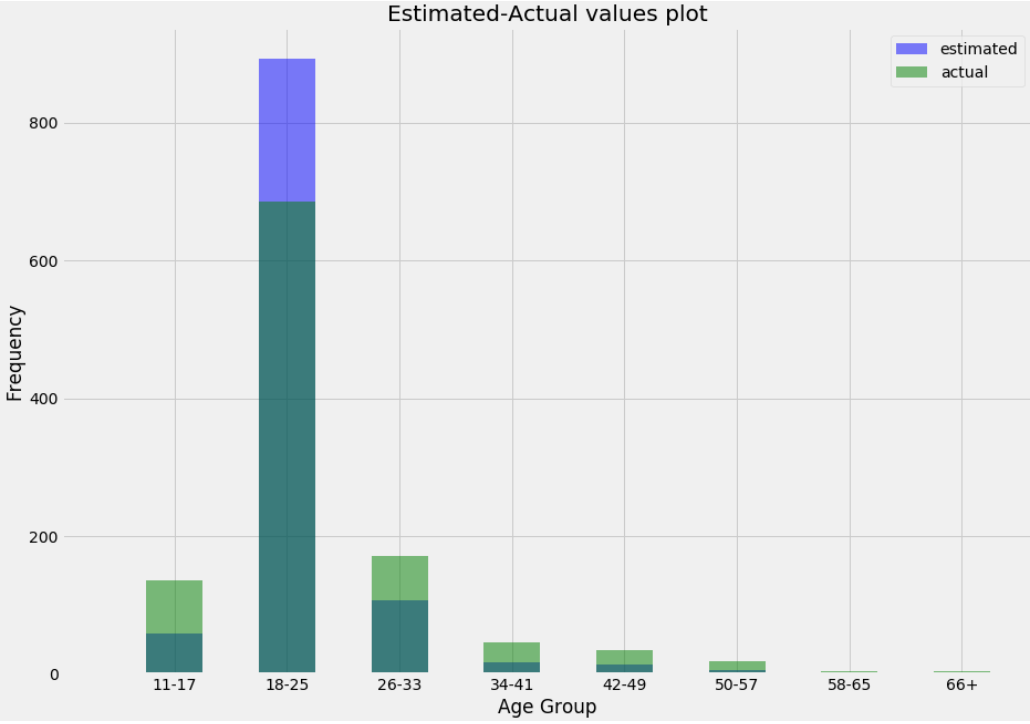
```
#best estimator model from randomsearch process
best_model = knn_random.best_estimator_
best_model.fit(X_train,y_train)
#Evaluate
model_name = 'KNN'
y_knn_test = best_model.predict(X_test)
evaluation_metrics(best_model, model_name, y_knn_test, y_test)
save_res(best_model, model_name, y_knn_test, y_test)
plotFunction(y_knn_test, "KNN predictions")
doublePlotFunction(y_knn_test, y_test)
```

Ο confusion matrix του αλγορίθμου ταξινόμησης KNN φαίνεται στον πίνακα 8.15, όπου με πράσινο χρώμα έχουν σημειωθεί οι τιμές της διαγωνίου που αφορούν τις TP προβλέψεις, ενώ το μοντέλο δεν καταφέρνει να κάνει σωστές προβλέψεις για τις ηλικίες 58-65 και 66+.

Πίνακας 8.15: Ο Confusion matrix του KNN

		Πραγματικές τιμές (actual)							
		11-17	**18-25**	**26-33**	**34-41**	**42-49**	**50-57**	**58-65**	**66+**
Εκτιμήσεις (estimations)	**11-17**	34	21	2	1	0	1	0	0
	18-25	96	609	120	33	21	8	3	3
	26-33	5	43	43	4	5	5	1	0
	34-41	1	4	3	7	1	1	0	0
	42-49	0	5	2	0	6	0	0	0
	50-57	0	2	1	0	0	3	0	0
	58-65	0	0	0	0	0	0	0	0
	66+	0	1	0	0	1	0	0	0

Το διάγραμμα παρακάτω απεικονίζει τη σχέση μεταξύ εκτιμήσεων και αληθινών τιμών.



Εικόνα 8.23: Διάγραμμα εκτιμήσεων-πραγματικών τιμών για την ταξινόμηση με KNN

Το classification report για το βέλτιστο μοντέλο ταξινόμησης του αλγορίθμου KNN είναι το εξής:

Classification Report:

Age Groups	precision	recall	f1-score	support
11-17	0.58	0.25	0.35	136
18-25	0.68	0.89	0.77	685
26-33	0.41	0.25	0.31	171
34-41	0.41	0.16	0.23	45
42-49	0.46	0.18	0.26	34
50-57	0.50	0.17	0.25	18
58-65	0.00	0.00	0.00	4
66+	0.00	0.00	0.00	3
accuracy			0.64	1096
macro avg	0.38	0.24	0.27	1096
weighted avg	0.60	0.64	0.60	1096

Εικόνα 8.24: Αποτελέσματα αξιολόγησης ταξινόμησης με KNN

Από τα αποτελέσματα του μοντέλου KNN παρατηρούμε ότι επιτυγχάνει συνολικά accuracy 64%, precision 0,60, recall 0,64 και f1-score 0,60. Επίσης, το μοντέλο εμφανίζει τις καλύτερες τιμές για precision ίσες με 0,58 και 0,68 για τις ηλικιακές ομάδες 11-17 και 18-25 αντίστοιχα. Η υψηλότερη τιμή της μετρικής recall είναι 0,89 και f1-score 0,77 που αφορούν την ηλικιακή ομάδα 18-25. Αντίθετα στις μεγάλες ηλικίες 58-65 και 66+, όπου το test set περιλαμβάνει ελάχιστες εμφανίσεις γι αυτές, το μοντέλο δεν μπορεί να κάνει καμία πρόβλεψη και έχει μηδενικά αποτελέσματα για τις μετρικές precision, recall και f1-score.

8.3.6 Ταξινόμηση με Τυχαία Δάση (Random Forest)

Για την εύρεση του βέλτιστου μοντέλου για την ταξινόμηση με Τυχαία Δάση έγινε εφαρμογή του αλγορίθμου RandomizedSearchCV ώστε να προκριθούν οι καλύτερες υπερπαράμετροι. Ο αλγόριθμος αναζήτησε μέσα από ένα μεγάλο σύνολο παραμέτρων και τιμών που δηλώθηκαν στον κώδικα τις πιο αποδοτικές. Η διεργασία διήρκησε 7 λεπτά και 27 δευτερόλεπτα. Για την εκπαίδευση, την παρουσίαση και την αποθήκευση των αποτελεσμάτων του μοντέλου χρειάστηκε 1 λεπτό και 8 δευτερόλεπτα. Η εφαρμογή του αλγορίθμου παρουσιάζεται παρακάτω.

```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(random_state = 42)
# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 10, stop = 2000, num = 200)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(2, 110, num = 55)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 8, 10]
```

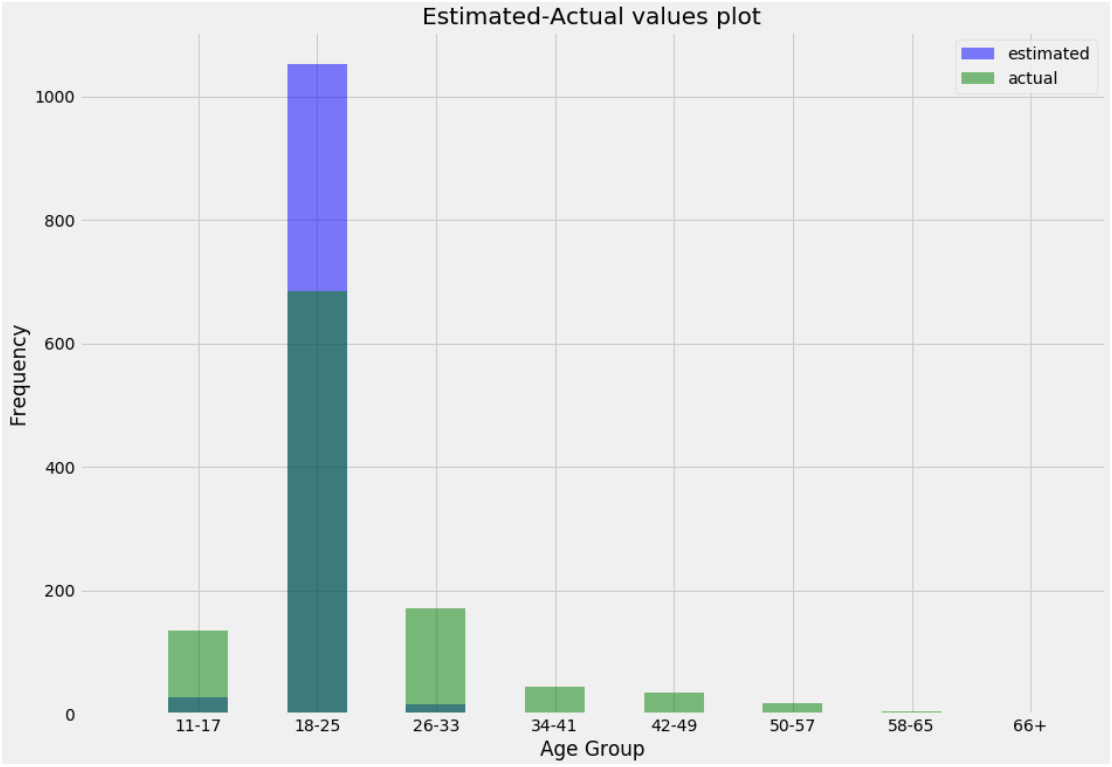
```
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 4]
# Method of selecting samples for training each tree
bootstrap = [True, False]
# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}
rf_random = RandomizedSearchCV(estimator = rf,
                               param_distributions = random_grid,
                               n_iter = 5,
                               cv = 3,
                               verbose=2,
                               random_state=42,
                               n_jobs = -1)
rf_random.fit(X_train, y_train)
#best estimator model from randomsearch process
best_model = rf_random.best_estimator_
best_model.fit(X_train,y_train)
#Evaluate
model_name = 'Random Forest'
y_rf_test = best_model.predict(X_test)
evaluation_metrics(best_model, model_name, y_rf_test, y_test)
save_res(best_model, model_name, y_rf_test, y_test)
plotFunction(y_rf_test, "Random Forest predictions")
doublePlotFunction(y_rf_test, y_test)
```

Στον πίνακα 8.16 παρουσιάζεται ο confusion matrix για το μοντέλο του αλγορίθμου ταξινόμησης με Τυχαία Δάση και με πράσινο χρώμα έχουν σημανθεί τα κελιά της διαγωνίου που είναι οι TP προβλέψεις. Διαπιστώνουμε πως ο αλγόριθμος αποτυγχάνει πλήρως να πραγματοποιήσει προβλέψεις για τις 4 ηλικιακές κατηγορίες των μεγάλων, τις 42-49, 50-57, 58-65 και 66+.

Πίνακας 8.16: Ο Confusion matrix για τα Τυχαία Δάση

		Πραγματικές τιμές (actual)							
		11-17	**18-25**	**26-33**	**34-41**	**42-49**	**50-57**	**58-65**	**66+**
Εκτιμήσεις (estimation)	**11-17**	24	2	0	0	1	0	0	0
	18-25	112	682	160	43	32	17	4	3
	26-33	0	1	11	1	1	1	0	0
	34-41	0	0	0	1	0	0	0	0
	42-49	0	0	0	0	0	0	0	0
	50-57	0	0	0	0	0	0	0	0
	58-65	0	0	0	0	0	0	0	0
	66+	0	0	0	0	0	0	0	0

Το αντίστοιχο διάγραμμα για τη σχέση εκτιμήσεων και αληθινών τιμών είναι.



Εικόνα 8.25: Διάγραμμα εκτιμήσεων-πραγματικών τιμών για την ταξινόμηση με Τυχαία Δάση

Το classification report για το βέλτιστο μοντέλο ταξινόμησης με τυχαία δάση είναι:

Classification Report:

Age Groups	precision	recall	f1-score	support
11-17	0.89	0.18	0.29	136
18-25	0.65	1.00	0.78	685
26-33	0.73	0.06	0.12	171
34-41	1.00	0.02	0.04	45
42-49	0.00	0.00	0.00	34
50-57	0.00	0.00	0.00	18
58-65	0.00	0.00	0.00	4
66+	0.00	0.00	0.00	3
accuracy			0.66	1096
macro avg	0.41	0.16	0.16	1096
weighted avg	0.67	0.66	0.55	1096

Εικόνα 8.26: Αποτελέσματα αξιολόγησης ταξινόμησης με Τυχαία Δάση

Μελετώντας τα αποτελέσματα του μοντέλου ταξινόμησης με Τυχαία Δάση βλέπουμε ότι συνολικά καταγράφει accuracy 66%, precision 0,67, recall 0,66 και f1-score 0,55. Το μοντέλο σημειώνει βέλτιστες τιμές precision ίσες με 0,89 και 1,00 για τις ηλικιακές ομάδες 11-17 και 26-33 αντίστοιχα. Η υψηλότερη τιμή για το recall είναι 1,00 και για το f1-score 0,78 και αφορούν την ηλικιακή ομάδα 18-25. Στις 4 μεγάλες ηλικιακές κλάσεις το μοντέλο έχει μηδενικά αποτελέσματα για όλες τις μετρικές αφού αδυνατεί να κάνει σωστές προβλέψεις.

8.3.7 Ταξινόμηση με Bernoulli

Το βέλτιστο μοντέλο ταξινόμησης Bernoulli με βρέθηκε μέσω της μεθόδου βελτίωσης υπερπαραμέτρων κάνοντας χρήση του αλγορίθμου RandomizedSearchCV. Η αναζήτηση πραγματοποιήθηκε για συγκεκριμένες παραμέτρους και τις αντίστοιχες τιμές τους που ορίστηκαν στον κώδικα. Η διαδικασία ώστε να ανακαλυφθεί ο καλύτερος συνδυασμός παραμέτρων διήρκησε περίπου 0,4 δευτερόλεπτα. Σχετικά με την εκπαίδευση του μοντέλου, την παρουσίαση και την αποθήκευση των αποτελεσμάτων του χρειάστηκαν περίπου 0,9 δευτερόλεπτα. Ο κώδικας που το εφαρμόζει είναι ο εξής.

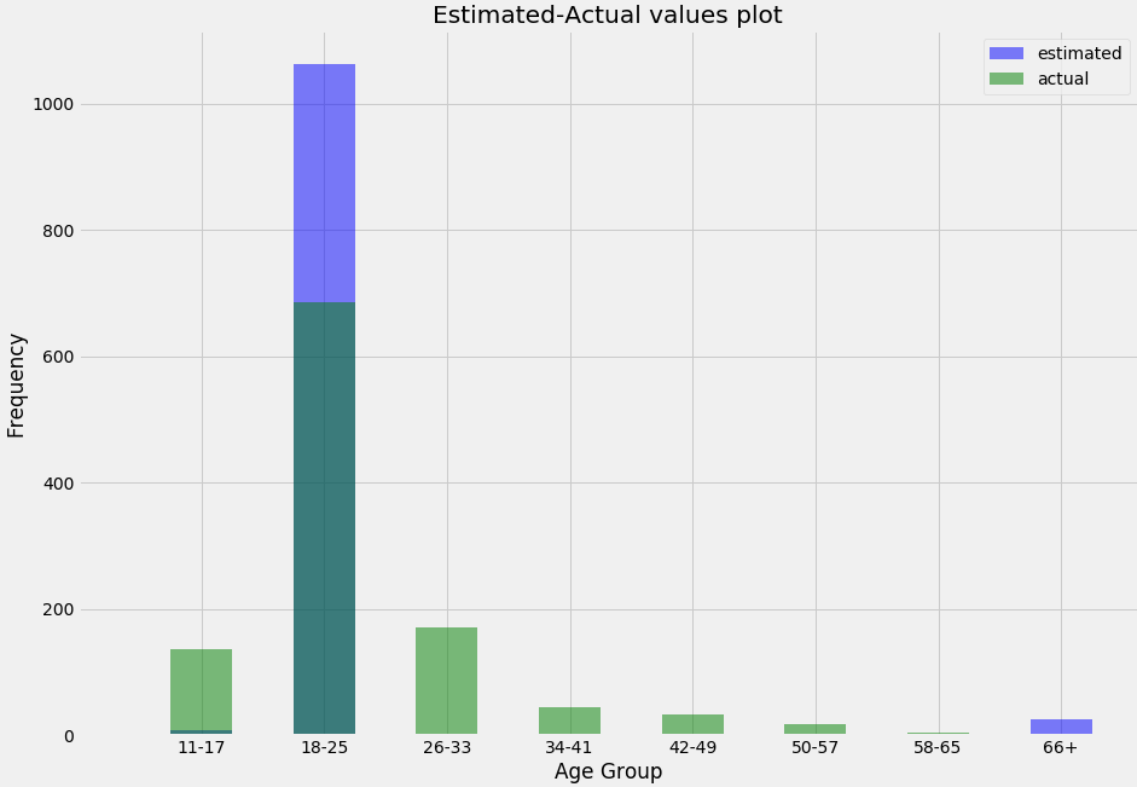
```
from sklearn.naive_bayes import BernoulliNB
bernoulli = BernoulliNB()
# aplha
alpha = [0.01, 0.1, 0, 1, 2, 5, 10, 20, 50]
# Create the random grid
random_grid = {'alpha': alpha}
bernoulli_random = RandomizedSearchCV(estimator = bernoulli,
                                      param_distributions = random_grid,
                                      n_iter = 5, cv = 5, verbose=2,
                                      random_state=42, n_jobs = -1)
bernoulli_random.fit(X_train, y_train)
#best estimator model from randomsearch process
best_model = bernoulli_random.best_estimator_
best_model.fit(X_train,y_train)
#Evaluate
model_name = 'Bernoulli'
y_bernoulli_test = best_model.predict(X_test)
evaluation_metrics(best_model, model_name, y_bernoulli_test, y_test)
save_res(best_model, model_name, y_bernoulli_test, y_test)
plotFunction(y_bernoulli_test, "Bernoulli predictions")
doublePlotFunction(y_bernoulli_test, y_test)
```

Ο confusion matrix του αλγορίθμου Bernoulli απεικονίζεται στον πίνακα 8.17, όπου με πράσινο χρώμα έχουν μαρκαριστεί οι τιμές της διαγωνίου που είναι οι TP προβλέψεις και εμφανίζονται μόνο για τις ηλικιακές κατηγορίες 11-17 και 18-25.

Πίνακας 8.17: Ο Confusion matrix του Bernoulli

		Πραγματικές τιμές (actual)							
		11-17	**18-25**	**26-33**	**34-41**	**42-49**	**50-57**	**58-65**	**66+**
Εκτιμήσεις (estimation)	**11-17**	3	4	1	0	0	0	0	0
	18-25	131	660	168	45	34	18	4	3
	26-33	0	0	0	0	0	0	0	0
	34-41	0	0	0	0	0	0	0	0
	42-49	0	0	0	0	0	0	0	0
	50-57	0	0	0	0	0	0	0	0
	58-65	0	0	0	0	0	0	0	0
	66+	2	21	2	0	0	0	0	0

Στο παρακάτω διάγραμμα φαίνεται η σχέση πραγματικών τιμών και εκτιμήσεων.



Εικόνα 8.27: Διάγραμμα εκτιμήσεων-πραγματικών τιμών για την ταξινόμηση με Bernoulli

Το classification report για το βέλτιστο μοντέλο ταξινόμησης μέσω του αλγορίθμου Bernoulli είναι:

Classification Report:

Age Groups	precision	recall	f1-score	support
11-17	0.38	0.02	0.04	136
18-25	0.62	0.96	0.76	685
26-33	0.00	0.00	0.00	171
34-41	0.00	0.00	0.00	45
42-49	0.00	0.00	0.00	34
50-57	0.00	0.00	0.00	18
58-65	0.00	0.00	0.00	4
66+	0.00	0.00	0.00	3
accuracy			0.60	1096
macro avg	0.12	0.12	0.10	1096
weighted avg	0.43	0.60	0.48	1096

Εικόνα 8.28: Αποτελέσματα αξιολόγησης ταξινόμησης με Bernoulli

Το μοντέλο ταξινόμησης που παράχθηκε από την εκπαίδευση του αλγορίθμου Bernoulli παρουσιάζει συνολικά accuracy 60%, precision 0,43, recall 0,60 και f1-score 0,48. Το μοντέλο πραγματοποιεί σωστές προβλέψεις μόνο για τις ηλικιακές κλάσεις 11-17 και 18-25 αποτυγχάνοντας για τις υπόλοιπες. Αυτό δικαιολογεί τις χαμηλές επιδόσεις του στο σύνολο των μετρικών και έτσι το καθιστούν αναξιόπιστο για τη διεξαγωγή προβλέψεων.

8.3.8 Συνοπτικά Αποτελέσματα των Αλγορίθμων Ταξινόμησης

Συγκρίνοντας τα αποτελέσματα μεταξύ των αλγορίθμων που εφαρμόστηκαν για την ανίχνευση της ηλικιακής ομάδας που ανήκουν οι χρήστες του Twitter, όπως αυτά συνοψίζονται στον πίνακα 8.18, διαπιστώνουμε ότι ο **XGBoost** παρουσιάζει τη μεγαλύτερη ακρίβεια, δηλαδή **accuracy 70%**, καθώς και τις καλύτερες τιμές για τις συνολικές μετρικές (weighted) **precision με 0,67, recall με 0,70 και f1-score με 0,66**. Όπως παρατηρήθηκε και από τα classification reports κάθε μοντέλου, ο XGBoost παράγει τα καλύτερα αποτελέσματα για κάθε ηλικιακή κατηγορία. Επίσης, είναι το μοντέλο που αποτυγχάνει να προβλέψει μόλις 2 ηλικιακές ομάδες, μαζί με αυτό που παράγει ο KNN, όμως οι επιδόσεις του XGBoost είναι αρκετά καλύτερες. Ενδιαφέρουσα κρίνεται η συνολική αλλά και η επιμέρους απόδοση των αλγορίθμων SVC, Τυχαία Δάση και Λογιστική Παλινδρόμηση. Ωστόσο και αυτοί αδυνατούν να πραγματοποιήσουν προβλέψεις για κάποιες ηλικιακές κατηγορίες. Αντίθετα, τα μοντέλα των Δέντρων Απόφασης και του Bernoulli έχουν τις λιγότερο ικανοποιητικές επιδόσεις. Πιο συγκεκριμένα, αποτυγχάνουν να κάνουν σωστές προβλέψεις για πολλές ηλικιακές ομάδες, 7 και 6 αντίστοιχα. Σε αυτό το σημείο πρέπει να τονιστεί ότι λόγω του ανεπαρκούς δείγματος για της μεγάλες ηλικίες άνω των 57 ετών, τα αποτελέσματα δεν είναι απόλυτα αξιόπιστα για αυτές και συνολικά επηρεάζουν την καλή απόδοση του μοντέλου αφού δημιουργούν θόρυβο στο δείγμα. Συνεπώς, η παρούσα μελέτη προτείνει για την επίλυση του προβλήματος ανίχνευσης της ηλικιακής ομάδας που ανήκουν οι χρήστες του Twitter όταν έχουν χωριστεί σε **8 ομάδες** με την τεχνική της ταξινόμησης (classification), το παραγόμενο μοντέλο του **XGBoost** αλγορίθμου. Ο Πίνακας 8.18 παρουσιάζει τα αποτελέσματα κάθε αλγορίθμου που δοκιμάστηκε και με πράσινο χρώμα σημειώνονται οι καλύτερες επιδόσεις για κάθε μετρική, ενώ με κόκκινο οι χειρότερες.

Πίνακας 8.18: Συνολικά αποτελέσματα αξιολόγησης των μοντέλων Ταξινόμησης

Αλγόριθμοι ταξινόμησης	**Μετρικές**			
	ACCURACY	**PRECISION**	**RECALL**	**F1-SCORE**
XGBoost	70%	0,67	0,70	0,66
Random Forest	66%	0,67	0,66	0,55
SVC	67%	0,61	0,67	0,60
Decision Tree	62%	0,39	0,63	0,48
Logistic Regression	62%	0,54	0,62	0,56
KNN	64%	0,60	0,64	0,60
Bernoulli	60%	0,43	0,60	0,48

Η προτεινόμενη λύση της παρούσας έρευνας σημειώνει σπουδαίες επιδόσεις σε σχέση με άλλες παρόμοιες μελέτες που διεξήχθησαν στο παρελθόν. Αυτό το πετυχαίνει παρότι χωρίζει τους χρήστες σε 8 ηλικιακές κατηγορίες. Επίσης, το δείγμα είναι ανομοιόμορφο αφού κάποιες κατηγορίες έχουν πολύ μεγάλη και άλλες πολύ μικρή παρουσία. Πιο συγκεκριμένα η κλάση 18-25 ετών ξεχωρίζει και καταλαμβάνει περίπου το 60% του συνολικού dataset, ενώ για τις μεγάλες ηλικίες άνω των 58 βλέπουμε ότι αποτελούν λιγότερο από το 1% των δεδομένων. Ωστόσο, το μοντέλο είναι πιο αποτελεσματικό από άλλα παρόμοιων ερευνών.

Αρχικά μία πρώτη σύγκριση του παραγόμενου μοντέλου γίνεται με αυτό του άρθρου [77], όπου οι συγγραφείς του χωρίσανε τους χρήστες του Twitter σε 6 ηλικιακές ομάδες και αξιοποίησαν τα γλωσσολογικά δεδομένα τους. Η προτεινόμενη λύση της παρούσας εργασίας που συνδυάζει τα στοιχεία του προφίλ με τα λεξικογραφικά δεδομένα των χρηστών, πετυχαίνει accuracy 70%

με ομαδοποίηση σε 8 κλάσεις, γεγονός που ελαττώνει την ακρίβεια και δυσκολεύει τις προβλέψεις, ξεπερνώντας το αντίστοιχο accuracy του άρθρου [77] που σημείωσε 61%. Ταυτόχρονα το δείγμα μας, σε αντίθεση με αυτό του άρθρου [77], δεν είναι ομοιόμορφα κατανεμημένο και σε αυτό οφείλεται η μειωμένη απόδοση στις μεγάλες ηλικιακές κατηγορίες. Η δεύτερη σύγκριση του παρόντος μοντέλου γίνεται με το αντίστοιχο του άρθρου [74]. Το πρόβλημα της ανίχνευσης της ηλικιακής ομάδας των χρηστών προσεγγίζεται από την έρευνά μας ταξινομώντας τους χρήστες σε 8 ομάδες και εμφανίζει παρόμοια αποτελέσματα με αυτά του άρθρου [74] που χωρίζει τους χρήστες μόλις σε 3 κλάσεις. Πιο συγκεκριμένα, το μοντέλο του άρθρου παρουσιάζει accuracy 74%, precision 0,74, recall 0,74 και f1-score 0,74, ενώ αντίστοιχα το προτεινόμενο μοντέλο της μελέτης μας πετυχαίνει accuracy 70%, precision 0,67, recall 0,70 και f1-score 0,66. Οι δύο έρευνες εκμεταλλεύονται τόσο τα δεδομένα του προφίλ όσο και τα γλωσσολογικά στοιχεία των χρηστών.

Επιπρόσθετα, η τρίτη σύγκριση της προτεινόμενης λύσης μπορεί να γίνει με αυτή του άρθρου [76], όπου εκεί οι ερευνητές χώρισαν τους χρήστες σε 10 ηλικιακές ομάδες, έχοντας βέβαια διαφορετικό εύρος στα έτη σε σχέση με εμάς και ικανότερο δείγμα δεδομένων. Από τον έλεγχο της μετρικής F1-score (micro), παρατηρούμε ότι η λύση μας σημειώνει 0,70 έναντι 0,31 του άρθρου. Επίσης, όσον αφορά τις μετρικές precision και recall για τις επιμέρους ηλικιακές κλάσεις, η παρούσα προσέγγιση καταγράφει κατά μέσο όρο παρόμοιες τιμές, με 0,41 και 0,26 αντίστοιχα, σε σχέση με αυτές του άρθρου [76] που είναι 0,35 και 0,34 αντίστοιχα.

Επιπλέον, υλοποιήθηκε μία διαδικασία με σκοπό να δείξει το ηλικιακό εύρος λάθους για τις αποτυχημένες εκτιμήσεις. Έτσι για κάθε λανθασμένη πρόβλεψη υπολογίστηκε η διαφορά της πραγματικής τιμής της ηλικίας από το αντίστοιχο άνω ή κάτω άκρο της κατηγορίας που ταξινομήθηκε. Για παράδειγμα στην περίπτωση που κάποιος χρήστης είναι 16 ετών και ταξινομήθηκε λανθασμένα στην ομάδα 18 έως 25 ετών, βρίσκεται η απόλυτη τιμή της διαφοράς του 16 από το 18, που είναι το κάτω όριο της κλάσης που τοποθετήθηκε. Στο τέλος υπολογίστηκε ο μέσος όρος των διαφορών για κάθε αλγόριθμο. Επίσης υπολογίστηκε και το συνολικό MAE για όλες τις εκτιμήσεις.

Πίνακας 8.19: Απόλυτα Σφάλματα Αλγορίθμων Ταξινόμησης

Αλγόριθμοι	**MAE (εσφαλμένων εκτιμήσεων)**	**MAE (συνολικό)**
XGBoost	6,75	1,87
Logistic Regression	7,03	2,48
SVC	6,94	2,13
Decision Trees	6,89	2,59
KNN	7,06	2,38
Random Forest	7,01	2,44
Bernoulli	9,18	3,58

Στην υπάρχουσα βιβλιογραφία που ασχολείται με τον προσδιορισμό της ηλικίας των χρηστών του Twitter δεν έχει διεξαχθεί ανάλογη μελέτη που αξιολογεί την απόδοση του αλγορίθμου και κάνει εκτιμήσεις για την ακριβή τιμή της ηλικίας τους. Η παρούσα προσέγγιση καλύπτει αυτό το κενό και κρίνεται ιδιαίτερα πρωτοποριακή και επιτυχημένη αφού εμφανίζει εξαιρετικά

αποτελέσματα, παρουσιάζοντας πολύ μικρές αποκλίσεις σε έτη μεταξύ εκτιμήσεων και πραγματικών ηλικιών των χρηστών. Συνεπώς, από τον πίνακα 8.19 φαίνεται πως και σε αυτήν την περίπτωση ο **XGBoost** εμφανίζει το μικρότερο μέσο όρο για τα σφάλματα ίσο με **6,75 έτη**, ενώ συνολικά στις προβλέψεις του παρουσιάζει σφάλμα ίσο με **1,87 έτη**. Με πράσινο έχει σημειωθεί η βέλτιστη απόδοση μέσω του XGBoost μοντέλου και με κόκκινο η λιγότερο ικανοποιητική του αλγορίθμου Bernoulli.

Επιχειρήθηκε να γίνει μία πιο λεπτομερής μελέτη των αποτελεσμάτων των αλγορίθμων ταξινόμησης που δοκιμάστηκαν. Για αυτό το λόγο πραγματοποιήθηκε μία ανάλυση κατά την οποία υπολογίστηκε η τιμή σε έτη του μέσου απόλυτου σφάλματος (MAE) για κάθε ηλικιακή ομάδα. Το σφάλμα υπολογίστηκε για κάθε αλγόριθμο με παρόμοιο τρόπο, όπως αυτός που αναφέρθηκε παραπάνω για την κατασκευή του πίνακα 8.19, και αφορούσε το συνολικό σφάλμα για όλες τις εκτιμήσεις της ηλικίας. Τα αποτελέσματα σε αυτό το στάδιο εμφανίζουν αρκετό ενδιαφέρον. Όπως έχει ήδη αναφερθεί, η ανομοιομορφία του συνόλου δεδομένων και η απουσία επαρκούς δείγματος για τις ηλικίες άνω των 50 επηρεάζει αρνητικά τις εκτιμήσεις των μοντέλων, γεγονός που παρατηρείται έντονα αφού για τις αντίστοιχες ηλικιακές κλάσεις το MAE είναι αρκετά μεγάλο. Αντίθετα για τις μικρότερες ηλικίες, όπου υπάρχει κατάλληλος αριθμός δειγμάτων, οι εκτιμήσεις είναι εξαιρετικές και βρίσκονται πολύ κοντά στην πραγματική τιμή της ηλικίας. Συγκεκριμένα, για τη δεύτερη ηλικιακή κατηγορία 18 έως 25 ετών η απόκλιση είναι μικρότερη από μισό χρόνο, ενώ στην πρώτη για ηλικίες 11 μέχρι 17 το σφάλμα υπολογίστηκε περίπου ίσο με 1 έτος μόλις. Σε αυτό το σημείο πρέπει να τονιστεί πως δε λαμβάνεται υπόψη η τιμή 0 για το MAE που εξάγεται από τον αλγόριθμο των δέντρων απόφασης, ο οποίος όπως αναλύθηκε στην ενότητα 8.3.4 προβλέπει σε κάθε περίπτωση μόνο την 2^{η} κατηγορία. Συνεπώς, από τα δεδομένα του πίνακα 8.20 διαπιστώνουμε πως το μοντέλο του **XGBoost** καταγράφει τις καλύτερες επιδόσεις. Στον πίνακα 8.20 έχουν επισημανθεί με πράσινο οι καλύτερες τιμές που παρουσιάστηκαν για το μέσο απόλυτο σφάλμα (MAE) ανά ηλικιακή ομάδα.

Πίνακας 8.20: MAE ανά ηλικιακή ομάδα

Ηλικιακές Ομάδες	**Αλγόριθμοι**						
	SVC	**XGBoost**	**Random Forest**	**Logistic Regression**	**KNN**	**Bernoulli**	**Decision Tree**
11-17	0,84	0,99	1,61	1,18	1,80	2,68	2,02
18-25	0,22	0,24	0,001	0,74	0,70	1,38	0
26-33	2,98	2,29	3,04	2,98	2,74	3,68	3,20
34-41	10,58	8,07	10,89	10,87	9,24	11,56	11,56
42-49	17,62	15,62	19,79	17,41	14,88	20,5	20,5
50-57	24,56	23,28	27,67	24,11	21,33	28,56	28,56
58-65	32,5	34,5	34,5	30,5	32,5	34,5	34,5
66+	53,33	37,33	48,0	50,67	48,0	48,0	48,0

Κεφάλαιο 9

9 Επίλογος

9.1 Σύνοψη και Συμπεράσματα

Στις μέρες μας η χρήση των μέσων κοινωνικής δικτύωσης είναι ευρεία και συνεχώς αυξανόμενη. Το γεγονός αυτό έχει δημιουργήσει την τάση στην ερευνητική κοινότητα να μελετήσει και να αναλύσει τα κοινωνικά δίκτυα εμβαθύνοντας στις πληροφορίες που αυτά παρέχουν για τους χρήστες τους. Αποτελούν χρήσιμη πηγή μεγάλου όγκου δεδομένων με στόχο την πραγματοποίηση προβλέψεων και την εξαγωγή των προτιμήσεων και των ενδιαφερόντων των χρηστών τους.

Στα πλαίσια της παρούσας διπλωματικής εργασίας προτάθηκε ένας αλγόριθμος μηχανικής μάθησης για την ανίχνευση της ηλικίας και ένας για την ανίχνευση της ηλικιακής ομάδας που ανήκουν οι χρήστες του Twitter. Οι αλγόριθμοι αυτοί εκμεταλλεύονται τα στοιχεία που λαμβάνονται από το προφίλ του χρήστη καθώς και το περιεχόμενο των tweets που έχει αναρτήσει ώστε να εκτιμήσει την ζητούμενη ηλικία ή ηλικιακή ομάδα αντίστοιχα. Η συλλογή των στοιχείων αυτών είναι μία σχετικά εύκολη, γρήγορη αλλά και απαιτητική διαδικασία μέσω του API που διαθέτει το Twitter. Αρκετός χρόνος χρειάζεται και για την επεξεργασία των δεδομένων ώστε να έρθουν σε κατάλληλη μορφή για να μπορούν να χρησιμοποιηθούν ως είσοδοι των μοντέλων.

Πιο συγκεκριμένα, για την εκπόνηση της παρούσας έρευνας χρειάστηκε η εξόρυξη των δεδομένων του προφίλ ενός συνόλου χρηστών του Twitter, ταυτοποιημένων ηλικιακά, και η συλλογή των tweets που έχουν αναρτήσει. Αυτά τα δεδομένα, αφού αποθηκεύθηκαν σε αρχεία τύπου CSV, πέρασαν αρκετά στάδια επεξεργασίας μέχρι να λάβουν την τελική τους μορφή και να επιλεχθούν τα κατάλληλα χαρακτηριστικά για την εκπαίδευση των αλγορίθμων. Στο πρώτο στάδιο εξετάστηκαν και εξήχθησαν στατιστικά στοιχεία για κάθε χρήστη σχετικά με το προφίλ του, όπως ο αριθμός των followers, των followings, των likes που έχει κάνει ή το πλήθος των posts που έχει δημοσιεύσει, τα οποία κρατήθηκαν σε νέο αρχείο. Στη συνέχεια, με την εφαρμογή τεχνικών NLP στα tweets των χρηστών παράχθηκαν στατιστικά δεδομένα σχετικά με τα γλωσσολογικά τους γνωρίσματα, όπως το πλήθος των hashtags (#) ή των tags (@) που χρησιμοποιούν, ενώ έγινε και αναγνώριση του θέματος των tweets μεταξύ 8 θεμάτων που ορίστηκαν ως τομείς ενδιαφερόντων. Τα εξαγόμενα δεδομένα αποθηκεύτηκαν σε νέα αρχείο CSV. Το τρίτο στάδιο αποτέλεσε την προσπάθεια εμβάθυνσης στην ανίχνευση του θέματος (topic modelling) των tweets των χρηστών μέσω εφαρμογής των αλγορίθμων LDA και GuidedLDA, η οποία όμως απέτυχε και δεν εμφάνισε ικανοποιητική απόδοση με αποτέλεσμα να απορριφθεί και να εξαιρεθεί από τη συνέχεια της μελέτης. Το τελευταίο βήμα πριν την σύνθεση του τελικού συνόλου δεδομένων ασχολήθηκε με την εξαγωγή αριθμητικών πληροφοριών από τα λεξικογραφικά δεδομένα των χρηστών μελετώντας τα tweets και την αποθήκευσή τους σε ξεχωριστό αρχείο. Στο τέλος, τα αρχεία δεδομένων συγκεντρώθηκαν σε ένα ενιαίο dataset που περιλάμβανε όλα τα features και αποτέλεσε την είσοδο για τους αλγορίθμους μηχανικής μάθησης των πειραμάτων.

Με βάση τα πειράματα που διεξήχθησαν και περιγράφηκαν στις ενότητες 8.2 και 8.3 για τις προσεγγίσεις της παλινδρόμησης (regression) και της ταξινόμησης (classification) αντίστοιχα, διαπιστώνουμε ότι προκύπτουν ιδιαίτερα ενδιαφέροντα αποτελέσματα. Τα δεδομένα που λαμβάνονται από το Twitter μπορούν να αποτελέσουν σημαντική πηγή πληροφορίας για τα ενδιαφέροντα των χρηστών, καθώς και να συμβάλλουν στη διεξαγωγή μελετών για την ανίχνευση της ηλικίας τους. Έπειτα από μία ενδελεχή έρευνα και την πραγματοποίηση αρκετών πειραμάτων, διαπιστώθηκε πως ο ιδιαίτερα δημοφιλής τα τελευταία χρόνια αλγόριθμος μηχανικής μάθησης **XGBoost**, με την κατάλληλη παραμετροποίηση, είναι ικανός να αποδώσει εξαιρετικά εμφανίζοντας μεγάλη ακρίβεια στις εκτιμήσεις της ηλικίας. Αυτό παρατηρήθηκε και στις δύο προσεγγίσεις που επιλέχθηκαν στην παρούσα εργασία, προκρίνοντας τον ως την βέλτιστη λύση τόσο για την πρόβλεψη της ακριβούς τιμής της ηλικίας των χρηστών μέσω παλινδρόμησης (regression), όσο και για την ανίχνευση της ηλικιακής ομάδας που ανήκουν, διαλέγοντας μεταξύ 8 διακριτών κλάσεων, με την τεχνική της ταξινόμησης (classification). Οι εκτιμήσεις που έγιναν στη μέθοδο της παλινδρόμησης με XGBoost οδήγησαν σε μικρή τιμή του μέσου απολύτου σφάλματος MAE ίση με 4,09 έτη και σε εξίσου χαμηλό μέσο ποσοστιαίο απόλυτο σφάλμα MAPE ίσο με 16,48%. Όσον αφορά την υλοποίηση μέσω ταξινόμησης με XGBoost, όπου επιλέχθηκαν 8 ηλικιακές ομάδες, παρατηρήθηκαν και σε αυτή την περίπτωση ικανοποιητικές επιδόσεις με το προτεινόμενο μοντέλο να παρουσιάζει accuracy 70%, precision 0,67, recall 0,70 και f1-score 0,66. Αξίζει να σημειωθεί πως και στα δύο αυτά πειράματα εφαρμόστηκαν μέσω του αλγορίθμου RandomizedSearchCV, η μέθοδος της βελτίωσης υπερπαραμέτρων για την επιλογή των καλύτερων παραμέτρων εισόδου, καθώς και η τεχνική cross-validation με διαφορετικούς διαχωρισμούς στο σύνολο εκπαίδευσης (train set) και στο σύνολο δοκιμής (test set). Με τον ίδιο τρόπο δοκιμάστηκαν αρκετά μοντέλα τόσο για την παλινδρόμηση όσο και για την ταξινόμηση, όμως ο αλγόριθμος Xgboost πέτυχε σαφώς καλύτερη απόδοση από όλους.

Συνεπώς, τα μέσα κοινωνικής δικτύωσης και ιδιαίτερα το Twitter, μπορούν να παρέχουν σημαντική πληροφορία για την ηλικία των χρηστών τους μελετώντας τη δραστηριότητά τους και τα στοιχεία του προφίλ τους στην εκάστοτε πλατφόρμα. Αυτή η διαπίστωση αποτελεί το τελικό συμπέρασμα της διπλωματικής εργασίας και έρχεται να ενισχύσει ακόμα περισσότερο την σχετική άποψη των ερευνητών. Η μελέτη που πραγματοποιήθηκε δύναται να γενικευθεί και για άλλα κοινωνικά δίκτυα, καθώς και για δείγμα χρηστών τόσο μεγαλύτερο σε μέγεθος όσο και σε ποικιλομορφία των χαρακτηριστικών τους.

9.2 Μελλοντικές επεκτάσεις

Η παρούσα μελέτη ενδείκνυται για αρκετές πιθανές μελλοντικές επεκτάσεις. Ορισμένες από αυτές χρησιμεύουν σε λεπτομερέστερη αξιολόγηση της παρούσας προσέγγισης για την ανίχνευση της ηλικίας ή της ηλικιακής ομάδας που ανήκουν οι χρήστες του Twitter ενώ άλλες στοχεύουν στη βελτίωση των προβλέψεών της.

Η χρήση ενός πολύ μεγαλύτερου όγκου δεδομένων με πολλούς παραπάνω χρήστες αποτελεί έναν εύκολο και προφανή τρόπο αξιολόγησης της παρούσας προσέγγισης. Αυτό θα βοηθήσει στο να υπάρξει επιβεβαίωση ότι τα αποτελέσματα που καταγράφηκαν στην παρούσα εργασία μπορούν να γενικευτούν και να επεκταθούν για μεγάλο αριθμό χρηστών γεγονός που ίσως επιφέρει πιθανή βελτίωση στην απόδοση του μοντέλου αφού θα υπάρχει περισσότερη πληροφορία για επεξεργασία στη διάθεση του ερευνητή.

Όσον αφορά, την επέκταση της παρούσας λύσης χωρίς όμως την εισαγωγή και επεξεργασία παραπάνω πληροφορίων για τους χρήστες μια ενδιαφέρουσα προσέγγιση αποτελεί η δοκιμή διαφορετικών μοντέλων επιβλεπόμενης μάθησης για την ανίχνευση της ηλικίας των χρηστών ή της ηλικιακής ομάδας που ανήκουν, όπως είναι τα Δέντρα Αποφάσεων ή ο KNN μέσω της παλινδρόμησης ή ο SGD μέσω της ταξινόμησης. Επίσης, θα μπορούσε να γίνει εφαρμογή μοντέλων μη επιβλεπόμενης μάθησης, όπως η ομαδοποίηση (clustering). Τα μοντέλα μη επιβλεπόμενης μάθησης εκπαιδεύονται χωρίς να απαιτείται σύνολο δεδομένων που περιέχει χρήστες με γνωστή ηλικία, όπως αναφέρθηκε στην ενότητα 3.1. Με αυτόν τον τρόπο θα ήταν δυνατό να εκτελεστεί ένας αλγόριθμος μη επιβλεπόμενης μάθησης σε ένα τέτοιου είδους dataset με τυχαίους χρήστες και να οδηγήσει στην κατασκευή clusters, τα οποία θα αξιολογηθούν με τη βοήθεια των χρηστών ενός συνόλου δεδομένων όπου είναι γνωστή η ηλικία τους ή η ηλικιακή τους ομάδα.

Στην παρούσα λύση, όπως αναφέρθηκε στο κεφάλαιο 6, οι χρήστες του συνόλου δεδομένων αναλύθηκαν και μελετήθηκαν ανά έτος ως μεμονωμένοι χρήστες διευρύνοντας έτσι το μικρό υπάρχον dataset. Μία πιθανή προέκταση αυτής της προσέγγισης θα μπορούσε να εστιάσει στην πρόβλεψη της ηλικίας ή της ηλικιακής ομάδας του χρήστη μελετώντας την εξέλιξη του στο πέρασμα των ετών και όχι να τον ερευνά ως ξεχωριστή περίπτωση. Με αυτόν τον τρόπο γίνεται αξιοποίηση των δεδομένων για τα στοιχεία του προφίλ του αλλά και τις προτιμήσεις του στο παρελθόν. Έτσι το πρόβλημα μπορεί να επεκταθεί και σε ανάλυση χρονοσειράς. Ταυτόχρονα μέσω των ιστορικών δεδομένων μπορεί να εμπλουτιστεί το στάδιο της εκπαίδευσης και να οδηγήσει σε καλύτερες προβλέψεις εξόδου.

Επίσης μια επέκταση της παρούσας προσέγγισης μπορεί να συσχετιστεί με τον τρόπο ανάλυσης του κειμένου των tweets των χρηστών που μπορεί να γίνει με Βαθιά Μηχανική Μάθηση (Deep Learning). Έτσι είναι δυνατό να εφαρμοστούν συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks-CNN), ανατροφοδοτούμενα νευρωνικά δίκτυα (Recurrent Neural Networks-RNN) ή αναδρομικά νευρωνικά δίκτυα (Recursive Neural Networks-RNN) για την επεξεργασία και την εξαγωγή συμπερασμάτων για το περιεχόμενο των tweets. Μία πιθανή μελέτη βασισμένη σε συνδυασμό των ανατροφοδοτούμενων και αναδρομικών νευρωνικών δικτύων είναι η εφαρμογή της LSTM (Long short-term memory) αρχιτεκτονικής. Ακόμη μία προσέγγιση που γίνεται όλο και πιο δημοφιλής στις μέρες για την επίλυση NLP προβλημάτων αφορά τη χρησιμοποίηση της τεχνικής BERT[55] (Bidirectional Encoder Representations from Transformers) που έχει αναπτυχθεί από την Google μέσα στο 2018 και σημειώνει εξαιρετικά αποτελέσματα. Η χρήση deep learning αλγορίθμων είναι πιθανό να συμβάλλει στην εξαγωγή καλύτερων συμπερασμάτων για τα λεξικογραφικά στοιχεία των χρηστών και για τα θέματα που τους απασχολούν ωθώντας σε συνολική βελτίωση της απόδοσης των προβλέψεων για την ηλικία των χρηστών.

Επιπρόσθετα, ένας καλός τρόπος για τη βελτίωση του χρόνου εξόρυξης αποθήκευσης και επεξεργασίας των δεδομένων καθώς και της απόδοσης του μοντέλου θα ήταν η εφαρμογή μιας ικανής μηχανής ανάλυσης μεγάλου όγκου δεδομένων. Η χρήση του Apache Spark[56] για παράδειγμα θα βελτιώσει σημαντικά το χρόνο εκτέλεσης των διεργασιών αφού πρόκειται για ένα λογισμικό που πετυχαίνει μέχρι και 100 φορές ταχύτερες επιδόσεις από ένα απλό υπολογιστικό μηχάνημα. Παρέχει ταυτόχρονα streaming διεπαφή (interface) γεγονός που

[55]Πηγή: https://github.com/google-research/bert
[56] Πηγή: https://spark.apache.org/

μπορεί να εφαρμοστεί σε δεδομένα πραγματικού χρόνου (real-time). Για παράδειγμα, ο ερευνητής μπορεί να συλλέγει tweets και δεδομένα από το προφίλ του real-time ώστε να εισέρχονται στο σύστημα και να υπόκεινται επεξεργασία συνεχώς νέα δεδομένα. Αυτή η επιπλέον πληροφορία μπορεί να προσδώσει σημαντικά στοιχεία για τον χρήστη όπως πχ για τη μεταβολή των ακολούθων του ή των likes που συγκεντρώνει καθώς και να εμπλουτίσει τα δεδομένα για τα γλωσσικά του χαρακτηριστικά. Όλα αυτά τα στοιχεία που προσδίδει στο σύστημα ένα τέτοιου είδους λογισμικό όπως το Apache Spark μπορούν να αποφέρουν σημαντικές βελτιώσεις τόσο στο χρόνο επεξεργασίας όσο και στη βελτίωση του προτεινόμενου μοντέλου.

Μια καλή πρόταση για την βελτίωση αλλά κυρίως για τη γενίκευση της παρούσας προσέγγισης είναι η μελέτη χρηστών από διαφορετικές χώρες που αναρτούν tweets σε άλλη γλώσσα εκτός των αγγλικών. Τα υπάρχοντα πακέτα για την επίλυση προβλημάτων NLP διαθέτουν λεξικά για πολλές γλώσσες οπότε είναι εφικτή μία τέτοια εφαρμογή αφού εύκολα θα επεξεργάζονται τα κείμενα σε οποιαδήποτε από τις διαθέσιμες γλώσσες. Επιπλέον, η μελέτη αυτή μπορεί να φανερώσει διαφορές ή ομοιότητες στα χαρακτηριστικά, τις προτιμήσεις και τα ενδιαφέροντα μεταξύ των χρηστών παρόμοιων ηλικιακών ομάδων αλλά με άλλη εθνικότητα και να δημιουργήσει νέες ομάδες ανά χώρα. Έτσι εξετάζοντας ομάδες πληθυσμών από πολλές χώρες η λύση προεκτείνεται και σε παγκόσμιο επίπεδο αφού θα είναι δυνατόν να προβλεφθούν οι ηλικίες χρηστών από οποιαδήποτε περιοχή. Παράλληλα μέσω αυτής της γενικευμένης έρευνας και την εμφάνιση νέων target groups μπορεί να βελτιωθεί η αποτελεσματικότητα του μοντέλου ανά περίπτωση.

Τέλος, ακόμα μία προσέγγιση που είναι πιθανό να προσφέρει επιπλέον πληροφορίες για κάθε χρήστη με στόχο την αύξηση της ακρίβειας των προβλέψεων για την ηλικία του αποτελεί η βιογραφία (bio) του. Πρόκειται, για ένα μικρό εισαγωγικό κείμενο που όπως αναφέρθηκε στην ενότητα 5.1 βρίσκεται στο προφίλ του και συχνά περιέχει χρήσιμες πληροφορίες για τη ζωή του. Λόγω του μικρού μεγέθους του δεν επηρεάζει την κλιμακωσιμότητα (scalability) και την πολυπλοκότητα της προτεινόμενης λύσης στην παρούσα εργασία αλλά αντίθετα μπορεί να δώσει σημαντική συμπυκνωμένη πληροφορία. Ένας καλός τρόπος χρησιμοποίησης του είναι μέσω της αναζήτησης για λέξεις ή φράσεις κλειδιά που μπορούν να φανερώσουν την ηλικία του χρήστη σε συνδυασμό με την ύπαρξη αριθμών, όπως “age”, “years”, “25 years old” ή “born in 1995”. Κρίνεται όμως απαραίτητο να ελεγχθούν και να αποφευχθούν περιπτώσεις όπου λέξεις σχετικές με την ηλικία χρησιμοποιούνται με άλλο νόημα όπως επισημάνθηκε στο κεφάλαιο 2 και φαίνεται στα παραδείγματα “15 years working” ή “20 years as engineer”. Ωστόσο, η εκμετάλλευση του bio για εξαγωγή δεδομένων προαπαιτεί την χρήση διαφόρων γλωσσών στο στάδιο της επεξεργασίας. Αυτό οφείλεται κυρίως στο γεγονός ότι η ανάλυση του κειμένου εξαρτάται από την γλώσσα που έχει γραφτεί. Βέβαια, αυτός ο επιπλέον φόρτος μπορεί να παρακαμφθεί αν η έρευνα περιοριστεί μόνο σε συγκεκριμένους πληθυσμούς χρηστών του Twitter που μιλούν την ίδια γλώσσα.

Βιβλιογραφία

[1] https://www.statista.com/statistics/970920/monetizable-daily-active-twitter-users-worldwide/

[2] https://www.statista.com/statistics/283119/age-distribution-of-global-twitter-users/

[3] https://www.statista.com/statistics/828092/distribution-of-users-on-twitter-worldwide-gender/

[4] https://business.twitter.com/

[5] https://developer.twitter.com/en

[6] https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object

[7] https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/user-object

[8] A. Culotta, "Detecting influenza outbreaks by analyzing twitter messages," arXiv preprint arXiv:1007.4748, 2010.

[9] https://medium.com/datadriveninvestor/k-fold-and-other-cross-validation-techniques-6c03a2563f1e

[10] https://towardsdatascience.com/top-10-python-libraries-for-data-science-cd82294ec266

[11] https://www.geeksforgeeks.org/history-of-python/

[12] S. Alowibdi, U. A. Buy, and P. Yu. "Language independent gender classification on twitter. In IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining", pages 739–743. ACM, 2013.

[13] J. P. Carvalho, V. Pedro, and F. Batista, "Towards intelligent mining of public social networks' influence in society," in IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), (Edmonton, Canada), pp. 478 – 483, June 2013

[14] L. Sloan, J. Morgan, P. Burnap, M. Williams, "Who Tweets? Deriving the Demographic Characteristics of Age, Occupation and Social Class from Twitter User Meta-Data", March 2015

[15] Schwartz HA, Eichstaedt JC, Kern ML, Dziurzynski L, Ramones SM, "Personality, Gender, and Age in the Language of Social Media: The Open-Vocabulary Approach", 2013

[16] Morgan-Lopez AA, Kim AE, Chew RF, Ruddle P, "Predicting age groups of Twitter users based on language and metadata features.", 2017

[17] Simaki, Vasiliki & Mporas, Iosif & Megalooikonomou, Vasileios, "Age Identification of Twitter Users: Classification Methods and Sociolinguistic Analysis.", 2016

[18] Nguyen, D-P, Trieschnigg, RB, Dogruoz, AS, Gravel, R, Theune, M, Meder, T & de Jong, FMG 2014, "Why Gender and Age Prediction from Tweets is Hard: Lessons from a Crowdsourcing Experiment.", Proceedings of the 25th International Conference on Computational Linguistics, COLING 2014 (pp. 1950-1961), Association for Computational Linguistics (ACL), 2014

[19] https://help.twitter.com/en/managing-your-account/how-to-customize-your-profile

[20] E. Alpaydm, Introduction to Machine Learning Third Edition, The MIT Press, Massachusetts, 2014.

[21] https://towardsdatascience.com/cross-validation-in-machine-learning-72924a69872f

[22] P. Earle, M. Guy, R. Buckmaster, C. Ostrum, S. Horvath, and A. Vaughan, “Omg earthquake! can twitter improve earthquake response?” Seismological Research Letters, vol. 81, no. 2, pp. 246–251, 2010.

[23] https://www.geeksforgeeks.org/ml-extra-tree-classifier-for-feature-selection/

[24] https://towardsdatascience.com/hyperparameter-tuning-c5619e7e6624

[25] Kohavi, Ron. “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection”. International Joint Conference on Artificial Intelligence, vol 14, 1995.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. “Scikit-learn: Machine learning in Python. Journal of Machine Learning Research”, 12:2825–2830, 2011.

[27] G. Louppe, L. Wehenkel, A. Sutera and P. Geurts, “Understanding variable importances in forests of randomized trees”, Advances in Neural Information Processing Systems, vol. 26, 2013.

[28] J. Bergstra, Y. Bengio “Random Search for Hyper-Parameter Optimization”, J. Mach. Learn. Res. 13, (3/1/2012), 281–305, 2012.

[29] https://www.geeksforgeeks.org/history-of-python/

[30] S. Bird, E. Klein, E. Loper, J. Baldridge, “Multidisciplinary instruction with the Natural Language Toolkit”, September 2011

[31] S. Bird, E. Klein, E. Loper, “Natural Language Processing with Python”, O'Reilly Media Inc, 2009

[32] https://towardsdatascience.com/selecting-the-best-machine-learning-algorithm-for-your-regression-problem-20c330bad4ef

[33] https://xgboost.readthedocs.io/en/latest/

[34] T. Chen, C. Guestrin, “XGBoost: A Scalable Tree Boosting System”, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785-794, August 2016

[35] C. L. Chiang, “Statistical methods of analysis, World Scientific”, ISBN 981-238-310-7, 2003

[36] https://www.immagic.com/eLibrary/ARCHIVES/GENERAL/WIKIPEDI/W120702R.pdf

[37] JJM Moreno, AP Pol, AS Abad, BC Blasco, “Using the R-MAPE index as a resistant measure of forecast accuracy”, 2013

[38] https://en.wikipedia.org/wiki/Regression_analysis

[39] https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b

[40] W. Lin, Z. Wu, L. Lin, A. Wen, J. Li, “An Ensemble Random Forest Algorithm for Insurance Big Data Analysis”. IEEE Access, 5, 16568–16575, 2017

[41] https://en.wikipedia.org/wiki/Stochastic_gradient_descent

[42] Tin Kam Ho, “The random subspace method for constructing decision forests”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pp. 832-844, Aug. 1998

[43] https://en.wikipedia.org/wiki/Mean_absolute_error

[44] https://en.wikipedia.org/wiki/Mean_squared_error

[45] https://en.wikipedia.org/wiki/Root-mean-square_deviation

[46] A. Smola, B. Schölkopf, “A tutorial on support vector regression”, Statistics and Computing. 14 (3): 199–222, 2004

[47] H. Drucker, C. Burges, L. Kaufman, A. Smola, V. Vapnik, “Support Vector Regression Machines”, in Advances in Neural Information Processing Systems 9, NIPS 1996, 155–161, MIT Press, 1997

[48] J. A. K. Suykens, “Introduction to Machine Learning”, 2014

[49] https://en.wikipedia.org/wiki/Support_vector_machine

[50] S. Kotsiantis, “Supervised Machine Learning: A Review of Classification Techniques”, Informatica (Ljubljana), 31, July 2007

[51] J. Tolles, W. Meurer, “Logistic Regression: Relating Patient Characteristics to Outcomes”, JAMA, 2016

[52] C.-Y.J. Peng, K.L. Lee, G.M. Ingersoll, “An Introduction to Logistic Regression Analysis and Reporting”, The Journal of Educational Research, 2002

[53] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z-H Zhou, M. Steinbach, D. J. Hand, D. Steinberg, “Top 10 algorithms in data mining”, Knowledge and Information Systems, 14, 2008

[54] V. Metsis, I. Androutsopoulos, G. Paliouras, “Spam filtering with Naive Bayes—which Naive Bayes?”, Third conference on email and anti-spam (CEAS), 17, 2006

[55] Altman, N, “An introduction to kernel and nearest-neighbor nonparametric regression”, The American Statistician, 1992

[56] J.M. Bland, D.G. Altman, “Statistics notes: measurement error”, BMJ 312 (7047), 1996

[57] Y. Sasaki, “The truth of the F-measure”, 2007

[58] D. Powers, “Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation”, Journal of Machine Learning Technologies. 2 (1): 37–63, 2011

[59] L.van der Maaten, E. Postma, J. van den Herik, “Dimensionality Reduction: A Comparative Review”, October 2009

[60] N. Halko, P. G. Martinsson, and J. A. Tropp, “Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions”, 2009

[61] D. Blei, A. Ng, M.I. Jordan,”Latent Dirichlet Allocation", Journal of Machine Learning Research, 3 (4–5): pp. 993–1022, 2003

[62] D. Jurafsky, J. H. Martin, “Speech and Language Processing”, 2008

[63] Y. Goldberg, “A Primer on Neural Network Models for Natural Language Processing”, Journal of Artificial Intelligence Research, 57: 345–420, 2016

[64] R. Jozefowicz, O.Vinyals, M. Schuster, N. Shazeer, Y. Wu, “Exploring the Limits of Language Modeling”, 2016

[65] https://www.geeksforgeeks.org/bag-of-words-bow-model-in-nlp/

[66] https://en.wikipedia.org/wiki/Tf%E2%80%93idf

[67] Q. Fang, J. Sang, C. Xu, M.S. Hossain, "Relational User Attribute Inference in Social Media", IEEE Transactions on Multimedia, 17. 1-1. 10.1109/TMM.2015.2430819, July 2015

[68] J. Brea, J. Burroni, C. Sarraute, Carlos, "Inference of Users Demographic Attributes based on Homophily in Communication Networks", 2015

[69] M. Pennacchiotti, A.M. Popescu, "A Machine Learning Approach to Twitter User Classification", ICWSM. 11, 2011

[70] R. G. Guimarães, R. L. Rosa, D. De Gaetano, D. Z. Rodríguez,G. Bressan, "Age Groups Classification in Social Network Using Deep Learning", IEEE Access, vol. 5, pp. 10805-10816, doi: 10.1109/ACCESS.2017.2706674, 2017

[71] D. Rao, D. Yarowsky, A. Shreevats, M. Gupta, "Classifying latent user attributes in twitter", In Proceedings of the 2Nd International Workshop on Search and Mining User-generated Contents, SMUC '10 (pp. 37–44), 2010

[72] L. Molteni, J.L. Ponce, "Forecasting with twitter data: an application to Usa Tv series audience", Int. Journal of Design & Nature and Ecodynamics, Vol. 11, No. 3, pp. 220–229, July 2016

[73] A. Crisci, V. Grasso, P. Nesi, G. Pantaleo, I. Paoli, I. Zaza, "Predicting TV program audience by using twitter-based metrics", Multimedia Tools and Applications, Vol. 77, No. 10, pp. 12203–12232, 2018

[74] A.A. Morgan-Lopez, A.E. Kim, R.F. Chew, P. Ruddle, "Predicting age groups of Twitter users based on language and metadata features", PLoS ONE 12(8): e0183537, 2017

[75] L. Sloan, J. Morgan, P. Burnap, M. Williams, "Who Tweets? Deriving the Demographic characteristics of Age, Occupation and Social Class from Twitter User Meta-Data", PLoS ONE 10(3): e0115545, 2015

[76] B. Chamberlain, C. Humby, M. Deisenroth, "Probabilistic Inference of Twitter Users' Age Based on What They Follow", 10.1007/978-3-319-71273-4_16, 2017

[77] V. Simaki, I. Mporas, V. Megalooikonomou, "Age Identification of Twitter Users: Classification Methods and Sociolinguistic Analysis", 10.1007/978-3-319-75487-1_30, 2016

[78] X. Zhu, "Semi-Supervised Learning Literature Survey", Comput Sci, University of Wisconsin-Madison, 2, 2008