ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πρόβλεψη τοποθεσίας με χρήση τεχνικών μηχανικής μάθησης για τη βελτιστοποίηση υπηρεσιών υπολογιστικής ομίχλης σε μεγάλα γεγονότα έξυπνων πόλεων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σωτήριος Πελέκης

Επιβλέπουσα : Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

Αθήνα, Ιανουάριος 2019

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ
ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πρόβλεψη τοποθεσίας με χρήση τεχνικών μηχανικής μάθησης για τη βελτιστοποίηση υπηρεσιών υπολογιστικής ομίχλης σε μεγάλα γεγονότα έξυπνων πόλεων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Σωτήριος Πελέκης

Επιβλέπουσα : Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 14η Ιανουαρίου 2019.

.....	.....	.....
Θ. Βαρβαρίγου	Σ. Παπαβασιλείου	Δ. Ασκούνης
Καθ. Ε.Μ.Π	Καθ. Ε.Μ.Π.	Καθ. Ε.Μ.Π.

Αθήνα, Ιανουάριος 2019

.....
Σωτήριος Σ. Πελέκης

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

Περίληψη

Η διαχείριση του πλήθους (crowd management) σε μεγάλες εκδηλώσεις εντός των έξυπνων πόλεων, είναι ένα κεφάλαιο το οποίο απασχολεί σε μεγάλο βαθμό τους ερευνητές τα τελευταία χρόνια. Τέτοιες εκδηλώσεις σχετίζονται με τη συγκέντρωση κόσμου σε μία συγκεκριμένη περιοχή, με σκοπό την παρακολούθηση ενός μουσικού φεστιβάλ, ενός αθλητικού αγώνα, μίας παρέλασης, ή ακόμα και μιας ομιλίας. Στόχος είναι η καλύτερη εξυπηρέτηση του κοινού, καθώς και η λήψη μέτρων ασφαλείας, για την αποφυγή ατυχημάτων σε περίπτωση έκτακτης ανάγκης. Μελετώντας τη συμπεριφορά και την κίνηση του πλήθους, αντλούμε σημαντικές πληροφορίες για την επίτευξη των παραπάνω.

Στην παρούσα διπλωματική, αρχικά προτείνεται μια αρχιτεκτονική υπολογιστικής ομίχλης (fog architecture) για την υποστήριξη μεγάλων εκδηλώσεων. Σύμφωνα με την αρχιτεκτονική αυτή, η διαχείριση των εργασιών κατανέμεται είτε στις τοπικές συσκευές (edge devices) είτε στο νέφος (cloud), βάσει του προβλεπόμενου φόρτου εργασίας.

Στη συνέχεια, εξετάζουμε πως με τη χρήση τεχνικών μηχανικής μάθησης, αξιοποιώντας δεδομένα χωρικών συντεταγμένων και ανοιχτών πηγών (open data), μπορούμε να προβλέψουμε την κατανομή του κόσμου στο χώρο ενός δοθέντος μουσικού φεστιβάλ. Γίνεται αναφορά στον τρόπο με τον οποίο οι καιρικές συνθήκες, καθώς και η δημοτικότητα των καλλιτεχνών επηρεάζουν τη συμπεριφορά του κοινού εντός του φεστιβάλ. Το φεστιβάλ που μελετήσαμε είναι υπαίθριο, ονομάζεται Das Fest και διεξάγεται ετησίως στην Καρλσρούη της Γερμανίας. Τα δεδομένα που χρησιμοποιήσαμε αφορούν ένα υποσύνολο των επισκεπτών του φεστιβάλ για τις χρονιές 2017 και 2018.

Λέξεις Κλειδιά

Μηχανική Μάθηση, Ταξινόμηση, Παλινδρόμηση, Συσταδοποίηση, Εξυπνες Πόλεις, Υπολογιστική Ομίχλη, Υπολογιστικό Νέφος, Βαθιά Μάθηση

Abstract

Crowd management during large events is a very important aspect of smart cities. Such events refer to the grouping of people in a specific area, with the purpose of attending a happening such as a music festival, an athletic event, a parade, a celebration or a public speaking. The organizers are responsible for public safety and services. By observing the crowd's behavior, we are able to get prepared and be vigilant in any case of emergency. In the following diploma thesis, we introduce a fog architecture supporting large events in smart cities. Workload, data and services are allocated into edge devices and cloud resources following a policy related to the visitors' distribution prediction.

Later on, combining machine learning techniques and open data resources, we examine whether it is possible to predict visitors' behavior and movement during large events. We observe how weather conditions and artists' popularity affect the distribution of the visitors in between clustered spaces. The evaluation uses data collected from Das Fest, an open air large festival that took place in 2017 and 2018 in Karlsruhe, a city in southwest Germany.

Keywords

Machine Learning, Classification, Regression, Clustering, Smart Cities, Cloud Computing, Fog Computing, Deep Learning

Περιεχόμενα

Ευρετήριο Σχημάτων

Ευρετήριο Πινάκων

1. Εισαγωγή

Ο όρος Μηχανική Μάθηση και κυρίως η απόδοση του στα αγγλικά, δηλαδή Machine Learning (ML), θα έλεγε κανείς ότι είναι ακόμα και σήμερα ακούσματα "ηχηρά" τα οποία στο μέσο προκαλούν ένα δέος. Παρόμοια συμβαίνει και με τους όρους Βαθιά Μάθηση - Deep Learning (DL) και Τεχνητά Νευρωνικά Δίκτυα - Artificial Neural Networks (ANN). Όλα τα παραπάνω αποτελούν υποκατηγορίες του κλάδου της επιστήμης που ονομάζεται τεχνητή νοημοσύνη (AI). Στον επιστημονικο κόσμο οι έννοιες αυτές αποτελούν καθημερινότητα. Πλήθος επιστημόνων ασχολείται με τη θεμελίωση μεθόδων, τη σύγκριση και τη βελτίωση αλγόριθμων και τεχνικών και την εφαρμογή αλγόριθμων σε νέα προβλήματα πάσης φύσεως.

Τι σημαίνει όμως τεχνητή νοημοσύνη και μηχανική μάθηση; Οι περισσότεροι άνθρωποι οι οποίοι δεν είναι σχετικοί με το αντικείμενο θα φαντάζονταν ένα ρομπότ με κινητά μέλη να κάνει τις δουλειές του σπιτιού ή να μεταμορφώνεται σε αμάξι, επηρεασμένοι προφανώς και από σενάρια επιστημονικής φαντασίας. Βεβαίως, τα παραπάνω δεν αποτελούν, πλέον, σενάρια επιστημονικής φαντασίας, καθώς πρόσφατα το ανθρωποειδές ρομποτάκι της Boston Dynamics, εν ονόματι Atlas ανέπτυξε τη δυνατότητα να κάνει στροφές και backflips μιμούμενος τους ανθρώπινους μηχανισμούς κίνησης, αντίδρασης και ισορροπίας. Η βάση της μηχανικής μάθησης ωστόσο ξεκινάει σε πολύ πιο πρώιμα και χαμηλού επιπέδου στάδια. Δύο γνωστοί ορισμοί είναι οι εξής:

- "Μηχανική μάθηση: Πεδίο έρευνας που δίνει στους υπολογιστές τη δυνατότητα να μαθαίνουν χωρίς να είναι ρητά προγραμματισμένοι", Arthur Samuel (1959)
- "Μηχανική μάθηση είναι οποιαδήποτε διαδικασία με την οποία ένα σύστημα βελτιώνει τις επιδόσεις του μέσω εμπειρίας", Hebert Simon(1970)

Ας εξηγήσουμε τα παραπάνω με μερικά απλά παραδείγματα:

1. Διαθέτουμε ηχογραφήσεις των ήχων που κάνουν διάφορα είδη βατράχων. Γίνεται διάκριση σε ποιο είδος βατράχου ανήκουν οι ηχογραφήσεις αυτές μέσω μιας δοθείσας ετικέτας. Τροφοδοτούμε τον αλγόριθμο μηχανικής μάθησης με τα δεδομένα αυτά. Δεδομένου ότι είναι σωστά παραμετροποιημένος, ο αλγόριθμος μηχανικής μάθησης (στο [1] χρησιμοποιείται ο γνωστός αλγόριθμος SVM) "μαθαίνει" από αυτά τα δεδομένα να διακρίνει σε ποιον βάτραχο ανήκει μια ηχογράφηση. Δηλαδή, όταν εμείς δώσουμε, μετά απ' όλα αυτά μία νέα ηχογράφηση βατράχου χωρίς ετικέτα, ο αλγόριθμος είναι σε θέση, με καλό ποσοστό, να μας πει το είδος βατράχου από το οποίο προέρχεται. Το ποσοστό επιτυχίας του αλγόριθμου κατα κανόνα αυξάνεται όσο περισσότερα έγκυρα δεδομένα του δίνουμε, γεγονός που καθιστά εμφανή την έννοια της "μάθησης" και της εμπειρίας.
2. Το δεύτερο παράδειγμα είναι το κλασικό και δημοφιλές με την αναγνώριση spam μηνυμάτων ηλεκτρονικού ταχυδρομείου. Είναι ένα πρόβλημα στο οποίο έχουν δοκιμαστεί πολλοί αλγόριθμοι, όπως ο SVM και ο Naive Bayes αλλα και νευρωνικά δίκτυα, που θα δούμε στη συνέχεια, για κατηγοριοποίηση κειμένου και εκπαιδεύονται στο να αναγνωρίζουν μηνύματα κακόβουλου περιεχομένου και να τα τοποθετεί απευθείας στον αντίστοιχο φάκελο, προστατεύοντας έτσι τον χρήστη.
3. Αναγνώριση προσώπου σε φωτογραφίες, την οποία έχει χρησιμοποιήσει εδώ και χρόνια το facebook και για την οποία εκπαιδεύεται ένα νευρωνικό δίκτυο ώστε να εντοπίζει ένα συγκεκριμένο πρόσωπο σε μια φωτογραφία. Το δίκτυο χρησιμοποίησε για την εκπαίδευση του ετικέτες (tags) οι οποίες είχαν ήδη τοποθετηθεί από τους χρήστες στο παρελθόν, όταν το facebook, τους προέτρεπε να προσθέτουν ετικέτες στα πρόσωπα των φίλων τους. Το

συγκεκριμένο παράδειγμα προφανώς αποκτά και αμφιλεγόμενο χαρακτήρα καθώς ενδέχεται να έρθει σε συγκρούσεις με ανθρώπινα δικαιώματα όπως αυτά της ανωνυμίας και της ιδιωτικότητας.

Σε επιχειρησιακό και επαγγελματικό επίπεδο, τα εργαλεία αυτά βρίσκονται στο επίκεντρο της προσοχής καθώς εισάγουν νέες ευκαιρίες εξέλιξης και ανάπτυξης, προσφέροντας δυνατότητες και ισχυρά ανταγωνιστικά πλεονεκτήματα στις επιχειρήσεις. Μέσω της μηχανικής μάθησης από εδώ και στο εξής η λήψη αποφάσεων στον επιχειρησιακό κόσμο θα είναι σε μεγάλο βαθμό αποτέλεσμα ανάλυσης μεγάλων δεδομένων (data driven decision making). Τράπεζες χρησιμοποιούν ML για να προστατέψουν τους πελάτες τους από απάτες στις πιστωτικές τους κάρτες, εταιρείες χρησιμοποιούν ML για να χωρίσουν την πελατεία τους σε target groups και σε επίδοξους και μη αγοραστές των προϊόντων τους, επενδυτές το χρησιμοποιούν επίσης για να κάνουν πρόβλεψη των τιμών του χρηματιστηρίου ή κρυπτονομισμάτων, μετεωρολογικές υπηρεσίες βελτιώνουν τις προβλέψεις τους χρησιμοποιώντας μοντέλα που εκπαιδεύονται παρακολουθώντας καιρικά μοτίβα του παρελθόντος.

Χαρακτηριστική είναι, επίσης η διείσδυση του AI στα ετήσια έξοδα και τους σχεδιασμούς των εταιρειών παγκοσμίως, πράγμα που προφανώς επεκτείνεται και στον υποκλάδο της μηχανικής μάθησης. Σύμφωνα με την Adobe:

- Επιχειρηματίες σήμερα επενδύουν έξι φορές περισσότερα χρήματα σε startups που σχετίζονται με AI, σε σχέση με το 2000.
- Μόνο 15% των επιχειρήσεων σήμερα χρησιμοποιούσαν AI τον Απρίλιο του 2018, ποσοστό που αναμένεται να εκτοξευθεί στο 31% στο επόμενο δωδεκάμηνο.
- Το μερίδιο θέσεων εργασίας παγκοσμίως που αφορούν σε AI έχει αυξηθεί κατα 450% σε σχέση με το 2013.

Στον πίνακα 1.1 βλέπουμε ορισμένες από τις συνεισφορές του AI σε διάφορους τομείς της βιομηχανίας, οι οποίες ανοίγουν δρόμους για απελευθέρωση χρόνου αλλά και πιο προσωποποιημένη και πιο ποιοτική παροχή υπηρεσιών:

	Τομέας	Εφαρμογές και συνεισφορές AI
	Υγείας και πρόνοιας	• Καλύτερες διαγνώσεις ασθενειών με βάση τα μοτίβα μεταβλητότητα των δεδομένων υγείας των ασθενών • Διάγνωση ασθενειών από εικόνες • Έγκαιρη πρόβλεψη πανδημιών
	Αυτοκίνηση	• Αυτόνομοι στόλοι αυτοκινήτων προς δημόσια χρήση • Αυτόνομη συντήρηση με δυνατότητα παρακολούθησης και πρόβλεψης βλαβών • Driver assistance
	Χρηματοοικονομικές υπηρεσίες	• Αυτοματοποίηση συναλλαγών με πελάτες • Ανίχνευση και εξάλειψη απατών και εσόδων από παράνομες δραστηριότητες • Προσωποποιημένος οικονομικός σχεδιασμός
	Μεταφορές	• Αυτόνομες μεταφορές και παράδοση φορτίων • Μεγαλύτερη ασφάλεια • Πρόβλεψη, έλεγχος κίνησης και αποφυγή συμφορήσεων,
	Τεχνολογίες, μέσα δικτύωσης και τηλεπικοινωνίες	• Προσωποποιημένο περιεχόμενο, διαφημίσεις και προτάσεις (recommendations) • Αυτοματοποιημένη και αποτελεσματική αρχειοθέτηση.
	Λιανική πώληση και καταναλωτές	• Πρόβλεψη ζήτησης • Προσωποποιημένος σχεδιασμός και παραγωγή • Στοχευμένο marketing • Αποτελεσματικότερη διαχείριση αποθεμάτων και παράδοσης εμπορευμάτων.
	Ηλεκτρική ενέργεια	• Smart metering • Αποτελεσματικότερη λειτουργία δικτύου,ι αποθήκευση ενέργειας, πρόβλεψη ζήτησης • Υποδομές με δυνατότητα πρόβλεψης αναγκών συντήρησης
	Παραγωγή	• Αποτελεσματική παρακολούθηση και έλεγχος, και αυτόματη επιδιόρθωση των διαδικασιών παραγωγής • Αυτορύθμιση παραγωγής βάσει της ζήτησης • Βελτιστοποίηση λειτουργίας γραμμών παραγωγής και προμηθειών.

Πίνακας 1.1. Εφαρμογές και συνεισφορές του AI στον επιχειρησιακό κόσμο

2. Ο επιστημονικός κλάδος της μηχανικής μάθησης

Στην ενότητα αυτή γίνεται μια αναφορά στη διάρθρωση της επιστήμης της μηχανικής μάθησης. Στη συνέχεια γίνεται μια αναφορά στα επιμέρους επιστημονικά πεδία και κατηγορίες που την απαρτίζουν, όπως είναι η επιτηρούμενη, η μη επιτηρούμενη και η ημιεπιτηρούμενη μάθηση. Στις κατηγορίες αυτές εξετάζονται μια σειρά από σημεία ενδιαφέροντος και έννοιες αναγκαίες προς ορισμό και ανάλυση.

2.1 Στάδια μηχανικής μάθησης

Η επιστήμη της μηχανικής μάθησης μπορεί να διακριθεί για λόγους κατανόησης στα παρακάτω στάδια:

1. Συλλογή δεδομένων (data gathering): Απαιτείται να υπάρχει ένα σύνολο δεδομένων τα οποία θέλουμε να επεξεργαστούμε και να εξάγουμε συμπεράσματα.

2. Προεπεξεργασία ή προετοιμασία δεδομένων (data preprocessing): Τα δεδομένα συνήθως βρίσκονται σε μορφή η οποία δεν εξυπηρετεί την τροφοδότηση τους σε ένα μοντέλο μηχανικής μάθησης, συνεπώς είναι πολύ σημαντικό να τα επεξεργαστούμε με συγκεκριμένα εργαλεία και να τα φέρουμε σε κατάλληλη μορφή την οποία να μπορεί να χρησιμοποιήσει το μοντέλο. Χαρακτηριστικό εργαλείο προεπεξεργασίας δεδομένων της Python είναι η βιβλιοθήκη preprocessing. Παραδείγματα προεπεξεργασίας είναι η τακτοποίηση των δεδομένων μας σε μορφή διακριτών χαρακτηριστικών με αποδεκτό format (βλ. βιβλιοθήκες Pandas [2] και Numpy [3] της Python) η μετατροπή του χρόνου σε ένα συνεπές σύστημα μονάδων (βλ. κλάση time της Python) , η κανονικοποίηση των δεδομένων σε αποδεκτή από το μοντέλο κλίμακα για παράδειγμα στο διάστημα [0,1], η μετατροπή κατηγορικών μεταβλητών σε ακέραιους αριθμούς ή ακολουθίες από 0,1 (βλ. κλάσεις LabelEncoder, LabelBinarizer της preprocessing), η διαχείριση missing data (βλ. κλάση Imputer της preprocessing) κ.α.

3. Εύρεση και εξαγωγή χρήσιμων χαρακτηριστικών: Κατα τη διαδικασία αυτή ο αναλυτής εντοπίζει ακόμα και δημιουργεί ή φαντάζεται χαρακτηριστικά τα οποία είναι κρυμμένα μέσα στην πολύπλοκη δομή του dataset. Στη συνέχεια, καλείται να εξετάσει ποια από αυτά τα χαρακτηριστικά είναι σημαντικά και ποια όχι. Διαπιστώνεται, λοιπόν, ποια βοηθούν πραγματικά το μοντέλο στο να μάθει και ποια είναι εντελώς ασυσχέτιστα με την έξοδο ή είναι αλληλοεξαρτώμενα και χρήζουν απόρριψης. Πίσω από αυτό το στάδιο υπάρχει ένας ολόκληρος υποκλάδος της μηχανικής μάθησης με το όνομα feature engineering ο οποίος αποκτά όλο και μεγαλύτερη αξία και σημασία.

4. Επιλογή μοντέλου (model selection): Στη φάση αυτή επιλέγεται ένα μοντέλο ανάλογα με τη φύση του προβλήματος και την εμπειρία του αναλυτή και τα αποτελέσματα της αξιολόγησης. Δε θα είναι απαραίτητα μονάδικό. Είναι σύνηθες να γίνεται σύγκριση μοντέλων πριν την τελική επιλογή του κατάλληλου. Συχνά μάλιστα επιλέγεται ένας συνδυασμός μοντέλων με τεχνικές που παρουσιάζονται συνοπτικά στην ενότητα 2.2.1.2.

5. Εκπαίδευση μοντέλου (training): Επιλέγουμε ένα υποσύνολο του dataset το οποίο εφαρμόζουμε (fit) πάνω στο μοντέλο ώστε να εκπαιδευτεί και να ρυθμίσει κατάλληλα τις εσωτερικές του παραμέτρους. Με τον όρο εσωτερικές παράμετροι αναφερόμαστε στις παραμέτρους του μοντέλου οι οποίες μαθαίνονται κατα τη διαδικασία της εκπαίδευσης, όπως είναι για παράδειγμα η κλίσης ενός μοντέλου απλής γραμμικής παλινδρόμησης που θα εξεταστεί σε επόμενες ενότητες. Η εκπαίδευση του μοντέλου γίνεται μέσω ελαχιστοποίησης μιας συνάρτησης σφάλματος(π.χ mse, mae, cost functions, loss functions).

6. Αξιολόγηση μοντέλου (model evaluation): Στο στάδιο αυτό γίνεται χρήση μετρικών προκειμένου να διαπιστώσουμε πόσο καλά λειτουργεί το μοντέλο που εκπαιδεύσαμε.
    - Όσον αφορά στη μη επιτηρούμενη μάθηση, η αξιολόγηση μοντέλου είναι μια σχετικά διαισθητική διαδικασία εφόσον δεν έχουμε κάποιο δομημένο μέτρο εκτίμησης της απόδοσης και των σφαλμάτων μας. Ωστόσο, για παράδειγμα σε μεθόδους συσταδοποίησης υφίστανται μέτρα ομοιότητας, ενδοσυσταδικής και διασυσταδικής απόστασης τα οποία δίνουν στον αναλυτη μια εικόνα των επιδόσεων του μοντέλου και των τιμών των παραμέτρων που πρέπει να τροποποιήσει.
    - Όσον αφορά στην επιτηρούμενη μάθηση (ενότητα), η αξιολόγηση γίνεται με μετρικές όπως είναι το μέσο τετραγωνικό σφάλμα και το μέσο απόλυτο σφάλμα για τεχνικές παλινδρόμησης, ο πίνακας σύγχυσης (confusion matrix), τα precision - recall και η καμπύλη ROC για μοντέλα ταξινόμησης τα οποία θα αναλύσουμε σε επόμενες ενότητες. Χρησιμοποιούνται επίσης τεχνικές όπως αξιολόγησης μοντέλων βλ. ενότητα) όπως αυτή της διασταυρωμένης επικύρωσης ή cross validation.

7. Τροποποίηση παραμέτρων του μοντέλου (parameter tuning): Στο στάδιο αυτό γίνεται τροποποίηση των εξωτερικών παραμέτρων του μοντέλου (hyperparameters). Με τον όρο hyperparameters αναφερόμαστε σε παραμέτρους, οι οποίες παίρνουν τιμή από τον αναλυτή πριν την εκπαίδευση του μοντέλου και δε "μαθαίνονται" κατά τη διάρκεια της μάθησης, ωστόσο επηρεάζουν τη διαδικασία της μάθησης. Σ' ένα νευρωνικό δίκτυο για παράδειγμα τέτοιες παράμετροι είναι ο αριθμός νευρώνων, ο ρυθμός μάθησης κ.α. τα οποία είναι αντικείμενα του κλάδου της βαθιάς μηχανικής μάθησης για τον οποίο δε γίνεται εκτενής θεωρητική μελέτη στα πλαίσια της διπλωματικής αυτής εργασίας. Χαρακτηριστική είναι η υποκλάση model_selection.GridSearchCV της βιβλιοθήκης scikit-learn [4] της Python που δίνει στον αναλυτή τη δυνατότητα να δοκιμάσει διαφορετικές τιμές εξωτερικών παραμέτρων επαναληπτικά πάνω σε ένα μοντέλο και επιλέγει το συνδυασμό με τα καλύτερα αποτελέσματα με μετρικές αξιολόγησης και διαδικασία επικύρωσης της επιλογής μας (π.χ. cross validation).

8. Πρόβλεψη (prediction): Στην περίπτωση της επιτηρούμενης μηχανικής μάθησης, σκοπός είναι να κάνουμε προβλέψεις ετικετών σε νέα μη σεσημασμένα δεδομένα. Πρόκειται για το στάδιο πρακτικής εφαρμογής του μοντέλου μας και το λόγο της κατασκευής του.

2.2 Κατηγορίες μηχανικής μάθησης

2.2.1 Επιτηρούμενη μηχανική μάθηση

Επιτηρούμενη μηχανική μάθηση ή Supervised Learning (SL) ονομάζεται η υποκατηγορία της μηχανικής μάθησης όπου η διαδικασία μάθησης βασίζεται σε ζεύγη εισόδου και εξόδου. Το dataset

αποτελείται από μία σειρά από χαρακτηριστικά εισόδου (features) τα οποία αποτελούν τις ανεξάρτητες μεταβλητές ή αλλιώς ένα διάνυσμα ανεξάρτητων μεταβλητών $\mathbf{X} = (x_{1,}...x_k)$ και μία ετικέτα (label) - έξοδος (output) που αποτελεί την εξαρτημένη μεταβλητή y. Στον πίνακα 2.1 βλέπουμε ένα χαρακτηριστικό παράδειγμα συνόλου δεδομένων ή dataset επιτηρούμενης μάθησης. Οι πρώτες 5 στήλες αποτελούν τα features ή χαρακτηριστικά. Η τελευταία είναι ετικέτα ή έξοδος. Κάθε γραμμή του dataset αποτελεί ένα instance, πρότυπο ή αντικείμενο. Η εξαρτημένη μεταβλητή παίρνει τιμές στο συνεχή χώρο (πρόβλημα παλινδρόμησης) είτε στο διακριτό χώρο (πρόβλημα ταξινόμησης).

Κατά την επιτηρούμενη μάθηση, αφού έχουμε περάσει από τα στάδια της συλλογής και προεπεξεργασίας δεδομένων, συνήθως γίνεται ο διαχωρισμός τους σε ένα σύνολο εκπαίδευσης ή training set, σε ένα σύνολο επικύρωσης ή validation set και σε ένα test set:

- Το **training set**, αποτελεί το υποσύνολο του dataset με το οποίο τροφοδοτείται το μοντέλο μηχανικής μάθησης ή το νευρωνικό δίκτυο προκειμένου να εκπαιδεύσει τις εσωτερικές παραμέτρους του (για παράδειγμα τα βάρη και η σταθεροί όροι ενός νευρωνικού δικτύου).
- Το **validation set** αποτελεί το υποσύνολο του dataset το οποίο χρησιμοποιείται για να την αξιολόγηση του μοντέλου και τον έλεγχο της επίδοσης του σε δεδομένα στα οποία δεν έχουν χρησιμοποιηθεί για την εκπαίδευση του ώστε γίνει ρύθμιση των εξωτερικών παραμέτρων του μοντέλου (hyperparameters) από τον αναλυτή.
- Το **test set** αποτελεί το υποσύνολο του dataset, το οποίο δε χρησιμοποιείται για την εκπαίδευση του αλγόριθμου, και θεωρητικά ούτε για τη ρύθμιση των εξωτερικών παραμέτρων του αλλά χρησιμεύει για την εφαρμογή και αξιολόγηση του μοντέλου σε νέα γι΄αυτό δεδομένα. Το test set ουσιαστικά προσομοιώνει νέες άγνωστες εισόδους για το μοντέλο ενώ παράλληλα η επιθυμητή έξοδος είναι γνωστή ωστέ να ελέγχεται η επιτυχία των προβλέψεων. Θεωρητικά, όπως αναφέρεται και στο [5] το test set θα έπρεπε να παραμένει "κρυφό" μέχρι τη διαμόρφωση του τελικού μοντέλου και απλώς να χρησιμοποιηθεί για τον έλεγχο των επιδόσεων του χωρίς περαιτέρω ρύθμιση των εξωτερικών του παραμέτρων. Ωστόσο, στην πράξη, το test set και το validation set ενίοτε συγχέονται, δηλαδή προσαρμόζουμε τις εξωτερικές παραμέτρους του μοντέλου έτσι ώστε να έχουμε καλά αποτελέσματα στο test set και στην πορεία ακολουθούμε γνωστές διαδικασίες επικύρωσης όπως το cross-validation που αναλύεται σε επόμενη ενότητα, για να διαπιστώσουμε κατά πόσο γενικεύονται τα αποτελέσματα μας. Αυτό είναι λογικό να συμβαίνει διότι δεν υπάρχει συνήθως η πολυτέλεια μεγάλου όγκου δεδομένων. Το test set είναι κατά κανόνα μικρότερο στο πλήθος από το training set. Οι αναλογίες τους κατα κανόνα κυμαίνονται από 50-50% εώς 90-10% ανάλογα με τη φύση του προβλήματος και των δεδομένων.

← Features →					Label
Position	**Experience**	**Skill**	**Country**	**City**	**Salary ($)**
Developer	0	1	USA	New York	103100
Developer	1	1	USA	New York	104900
Developer	2	1	USA	New York	106800
Developer	3	1	USA	New York	108700
Developer	4	1	USA	New York	110400
Developer	5	1	USA	New York	112300
Developer	6	1	USA	New York	114200
Developer	7	1	USA	New York	116100
Developer	8	1	USA	New York	117800
Developer	9	1	USA	New York	119700
Developer	10	1	USA	New York	121600

Πίνακας 2.1: Ένα παράδειγμα dataset επιτηρούμενης μάθησης. [1]

Στη συνέχεια, επιλέγεται ένας υποψήφιος αλγόριθμος επιτηρούμενης μάθησης ανάλογα με τη φύση του προβλήματος η διαδικασία επιτηρούμενης μάθησης, η οποία διαρθρώνεται ως εξής:

Έστω ένα dataset **A** το οποίο χωρίζουμε σε training και test sets A_1, A_2 αντίστοιχα. Το A_1 αποτελείται από i γραμμές με συγκεκριμένες τιμές των ανεξάρτητων μεταβλητών $\mathbf{X} = (x_{1,}...x_k)$ και τις αντίστοιχες τιμές y_i της εξαρτημένης μεταβλητής y. Υποθέτουμε πως οι **X**,y συνδέονται μέσω μιας άγνωστης συνάρτησης στόχου για την οποία f ξέρουμε μόνο τις ακριβείς τιμές τις πάνω στα δοθέντα y_i, και για την οποία ισχύει:

$$y = f(x_1, \ldots x_k) = f(x) \qquad \text{(Εξίσωση 2.1)}$$

Στόχος είναι να κάνουμε, λοιπόν μια γενικευμένη εκτίμηση - μοντέλο h της f:

$$\hat{y} = h(x_1, \ldots x_k) = h(x) \qquad \text{(Εξίσωση 2.2)}$$

τέτοια ώστε να ελαχιστοποιείται μια συνάρτηση σφάλματος:

$$E(h) = \sum_x error(h(x), f(x)) \ , \ x \in A_1 \qquad \text{(Εξίσωση 2.3)}$$

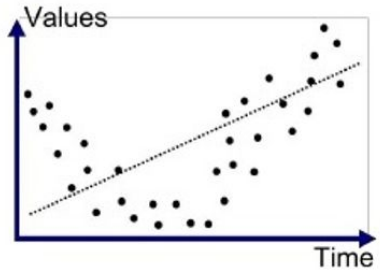
Η συνάρτηση σφάλματος (error function) που χρησιμοποιείται κατα την εκπαίδευση, ποικίλλει ως προς τη μορφή της ανάλογα με το είδος και τη φύση του προβλήματος που αντιμετωπίζουμε. Το μοντέλο h που κατασκευάζεται αξιολογείται ως προς τις επιδόσεις του πάνω στο validation set, όπως ήδη αναφέρθηκε.

[1] *[Πηγή:https://thenewstack.io/machine-learning-linear-regression-mere-mortals/]*

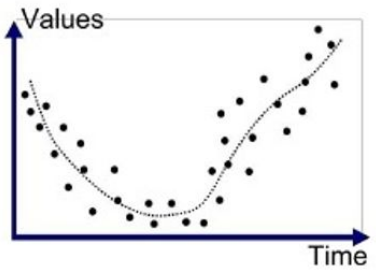
2.2.1.1 Bias variance tradeoff

Η επιτηρούμενη μηχανική μάθηση παρουσιάζει την εξής ιδιαιτερότητα: Πάντα αναζητείται η χρυσή τομή στην πολυπλοκότητα του μοντέλου που επιλέγουμε ανάλογα με τα δεδομένα του προβλήματος. Καλές επιδόσεις ενός μοντέλου στο training set συνδυαστικα με κακές επιδόσεις στο test set υποδηλώνουν overfitting. Αυτό σημαίνει ότι μοντέλο παλινδρόμησης ή ταξινόμησης μας παρουσιάζει υψηλή μεταβλητότητα (variance) και καταλήγει να είναι υπερβολικά προσαρμοσμένο πάνω στο training set ενσωματώνοντας μέχρι και το θόρυβο στα δεδομένα, όντας έτσι ανίκανο να κάνει προβλέψεις σε νέα, πραγματικά δεδομένα (βλ. Γράφημα 2.1.3, 3.1.3). Σ αυτή την περίπτωση τόσο οι παράμετροι του μοντέλου όσο και το ίδιο το μοντέλο τίθενται υπό αμφισβήτηση. Τέτοιο ζήτημα συναντάται συχνά σε πολυπαραμετρικά και ευέλικτα μοντέλα τα οποία μαθαίνουν πολλή λεπτομέρεια από τα δεδομένα (π.χ δέντρα αποφάσεων, πολυωνυμική παλινδρόμηση υψηλης τάξης).

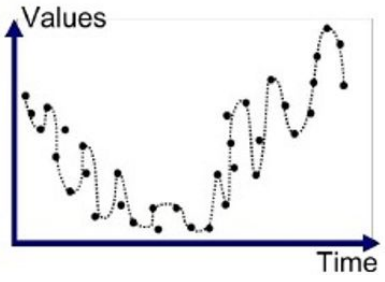
Απο την άλλη, κακές επιδόσεις στο training set ή μεγάλο bias, δηλαδή μεγάλες και συστηματικές αποκλίσεις πρόβλεψης από την αναμενόμενη τιμή, υποδηλώνουν underfitting. Αυτό σημαίνει ότι το μοντέλο μας είναι υπεραπλουστευμένο και κατα κάποιο τρόπο δύσκαμπτο. Κάτι τέτοιο για παράδειγμα συμβαίνει όταν προσπαθούμε να προβλέψουμε μη γραμμικά δεδομένα με ένα απλο μοντέλο γραμμικής παλινδρόμησης (βλ. Γράφηματα 2.1.1., 3.1.1) Στην περίπτωση αυτή, που είναι εύκολο να εντοπιστεί, απαιτείται να αντικαταστήσουμε το μοντέλο μας με ένα πιο σύνθετο και παραμετροποιήσιμο μοντέλο.



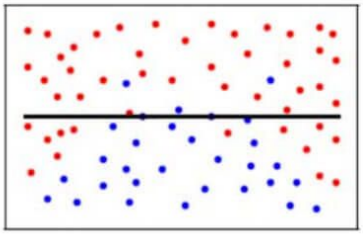
Εικόνα 2.1: Παρουσίαση ενός underfitted μοντέλου γραμμικής παλινδρόμησης.[2]



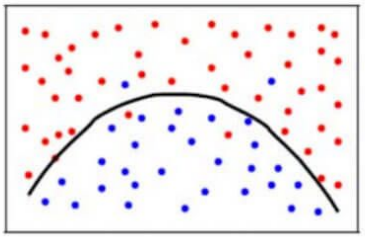
Εικόνα 2.2: Παρουσίαση ενός καλώς προσαρμοσμένου μοντέλου παλινδρόμησης.[3]



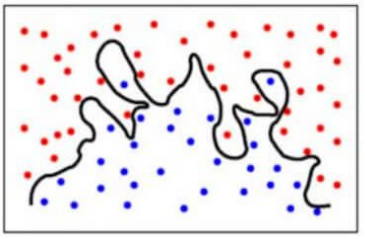
Εικόνα 2.3: Παρουσίαση ενός overfitted μοντέλου πολυωνυμικής παλινδρόμησης.[3]



Εικόνα 2.4: Παρουσίαση ενός underfitted μοντέλου ταξινόμησης.[3]



Εικόνα 2.5: Παρουσίαση ενός καλώς προσαρμοσμένου μοντέλου ταξινόμησης.[4]



Εικόνα 2.6: Παρουσίαση ενός overfitted μοντέλου ταξινόμησης.[4]

[2] *[Πηγή:https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76]*

[3] *[Πηγή εικόνων: https://tomrobertshaw.net/2015/12/introduction-to-machine-learning-with-naive-bayes/]*

2.2.1.2 Συνδυασμός μεθόδων (ensemble methods)

Όπως είδαμε στα στάδια του ML, κατα την επιλογή μοντέλου δεν είναι απαραίτητο πως η μέθοδος που θα χρησιμοποιήσουμε θα είναι μοναδική. Συχνά ο συνδυασμός μεθόδων βελτιώνει τα τελικά αποτελέσματα. Οι δύο κύριες τεχνικές συνδυασμού μεθόδων είναι οι bagging και boosting. Στις τεχνικές αυτές το κάθε μοντέλο του συνδυασμού εκπαιδεύεται ατομικά και το αποτέλεσμα της πρόβλεψης αποτελεί μια συνάρτηση των προβλέψεων των επιμέρους μοντέλων.

2.2.1.2.1 Bagging (bootstrap aggregating)

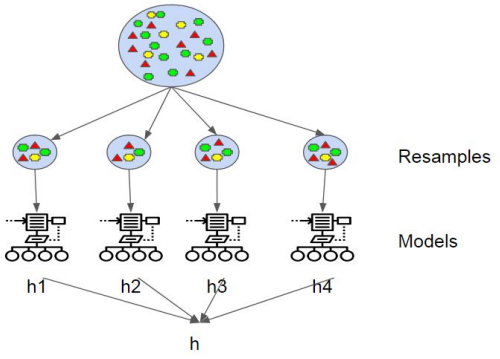
Το Bagging [6] προέρχεται από τη φράση bootstrap aggregating και αποτελεί μια συνδυαστική μέθοδο της κατηγορίας του bootstrapping που πρώτη φορά αναλύεται στο [7]. Στην τεχνική αυτή:

- Κάθε επιμέρους μοντέλο εκπαιδεύεται με training set ένα τυχαίο υποσύνολο του αρχικου training set, με την επιλογή στοιχείων για τη δόμηση του να γίνεται ομοιόμορφα και με αντικατάσταση όπως φαίνεται και στην εικόνα 2.7. (bootstrapping).
- Το τελικό αποτέλεσμα στην παλινδρόμηση ορίζεται ως ο μέσος όρος των εκτιμήσεων των επιμέρους μοντέλων ένω στην ταξινόμηση επιλέγεται η κλάση με τις περισσότερες ψήφους. (aggregating)

Το bagging προσφέρει τα εξής πλεονεκτήματα:

1. Περιορίζει το overfitting σε περίπτωση μοντέλων υψηλής μεταβλητότητας.
2. Περιορίζει το underfitting σε περίπτωση μοντέλων υψηλού bias (π.χ decision trees)
3. Μειώνει το θόρυβο χρησιμοποιώντας πολλές τυχαίες δειγματοληψίες.
4. Βοηθάει να χτίσουμε πιο ισχυρά μοντέλα σε μικρά dataset.

Κλασική τεχνική Bagging αποτελούν τα Random Forests που συναντάμε σε επόμενο κεφάλαιο.



Εικόνα 2.7: Η λογική επιλογής δεδομένων εκπαίδευσης και κατασκευής τελικού μοντέλου στο bagging [4]

2.2.1.2.2 Boosting

"Can a set of weak learners create a single strong one?" [8]

Η τεχνική του boosting απαντάει σε αυτό το ερώτημα.Το boosting περιορίζει το underfitting σε περίπτωση μοντέλων υψηλού bias (π.χ ρηχό δέντρο αποφάσεων) συνδυάζοντας τα και κατασκευάζοντας ένα πιο μεταβλητό - προσαρμοστικό μοντέλο. Αδύναμος αλγόριθμος μάθησης θεωρείται ένας αλγόριθμος ο οποίος πετυχαίνει αποτελέσματα οριακά καλύτερα από τυχαία, όπως ένα

[4] *[Πηγη:https://hackernoon.com/how-to-develop-a-robust-algorithm-c38e08f32201]*

δέντρο αποφάσεων ενός επιπέδου (decision stump). Το boosting βασίζεται επίσης στη μέθοδο του bootstrapping όπως είδαμε και για το bagging ωστόσο υφίστανται οι εξής διαφοροποιήσεις:

- ➢ Η εκπαίδευση στο boosting ολοκληρώνεται μετά από κάποιο αριθμό επαναλήψεων.
- ➢ Ο αλγόριθμος συγκρατεί ποια από τα επιμέρους datasets είχαν τα χειρότερα αποτελέσματα και τα αντιστοιχίζει σε μεγαλύτερα βάρη υπολογισμού για την επόμενη επανάληψη.
- ➢ Κατά την πραγματοποίηση της πρόβλεψης, ο αλγόριθμος, έχοντας κρατήσει αρχείο των επιδόσεων κάθε μοντέλου κατα τη διάρκεια της εκπαίδευσης, δίνει μεγαλύτερα βάρη στις μοντέλα με τα μικρότερα καταγεγραμμένα σφάλματα.

Στη βιβλιογραφία συναντάμε, ανάμεσα σε άλλες, τις παρακάτω εκδοχές του boosting:

- ❖ AdaBoost (**Ada**ptive **Boost**ing): Παρουσιάστηκε στο [9], του οποίου οι συγγραφείς βραβεύτηκαν το 2003 με βραβείο Gödel. Η λειτουργία του είναι να τροποποιεί κάθε φορά τα βάρη των δειγμάτων έτσι ώστε κάθε νέο αδύναμο μοντέλο που εκπαιδεύεται να λαμβάνει σοβαρά υπόψη τα λάθη των προηγούμενων. Τελικά συνδυάζονται οι αποφάσεις των επιμέρους μοντέλων ανάλογα με τις επιδόσεις τους. Χρησιμοποιεί κατα κανόνα decision stumps. Είναι, ωστόσο, αρκετά ευαίσθητος σε θόρυβο και outliers. Οι κλάσεις του στη βιβλιοθήκη scikit-learn της python python είναι οι εξης:

  Regression:
  class sklearn.ensemble.**AdaBoostRegressor**(*base_estimator=None*, *n_estimators=50*, *learning_rate=1.0*, *loss='linear'*, *random_state=None*)[5]

  Classification: *class* sklearn.ensemble.**AdaBoostClassifier**(*base_estimator=None*, *n_estimators=50*, *learning_rate=1.0*, *algorithm='SAMME.R'*, *random_state=None*)[6]

  Gradient Tree Boosting: Παρουσιάστηκε από τον Friedman στο μαζί με την εξέλιξη του Stochastic Gradient Boosting στα [10], [11] αντίστοιχα. Μοιάζει με την τεχνική Adaboost, ωστόσο εφαρμόζει λογική ελαχιστοποίησης τόσο πάνω σε μια συνάρτηση σφάλματος ή loss function, δηλαδή κάθε νέο μοντέλο εκπαιδεύεται πάνω στα σφάλματα πρόβλεψης των προηγούμενων με σκοπό την ελαχιστοποίηση τους. Το gradient boosting χρησιμοποιείται ευρέως σε προβλήματα anomaly detection. Οι κλάσεις του στη βιβλιοθήκη scikit-learn της python είναι οι εξης:

  Classification:
  class sklearn.ensemble.**GradientBoostingClassifier**(*loss='deviance'*, *learning_rate=0.1*, *n_estimators=100*, *subsample=1.0*, *criterion='friedman_mse'*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0* ,*max_depth=3*, *min_impurity_decrease=0.0*, *min_impurity_split=None*, *init=None*, *random_state=None*, *max_features=None*, *verbose=0*, *max_leaf_nodes=None*, *warm_start=False*, *presort='auto'*, *validation_fraction=0.1*, *n_iter_no_change=None*, *tol=0.0001*)[7]

[5] [Πηγή: https://github.com/scikit-learn/scikit-learn/blob/bac89c2/sklearn/ensemble/weight_boosting.py#L852]
[6] [Πηγή: https://github.com/scikit-learn/scikit-learn/blob/bac89c2/sklearn/ensemble/weight_boosting.py#L295]
[7] *[Πηγή: https://github.com/scikit-learn/scikit-learn/blob/bac89c2/sklearn/ensemble/gradient_boosting.py#L1684]*

Regression:
class sklearn.ensemble.**GradientBoostingRegressor**(*loss='ls'*, *learning_rate=0.1*, *n_estimators=100*, *subsample=1.0*, *criterion='friedman_mse'*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_depth=3*, *min_impurity_decrease=0.0*, *min_impurity_split=None*, *init=None*, *random_state=None*, *max_features=None*, *alpha=0.9*, *verbose=0*, *max_leaf_nodes=None*, *warm_start=False*, *presort='auto'*, *validation_fraction=0.1*, *n_iter_no_change=None*, *tol=0.0001*)[8]

- XGBoost: Αποτελεί την state-of-the-art τροποποίηση του gradient boosting. Είναι κατάλληλο για μεγάλα datasets σε καλές ταχύτητες. Επιπλέον, στην python δε χρειάζεται κανονικοποίηση δεδομένων (feature scaling), γεγονός που εξυπηρετεί πολύ τη διαρκή διαισθητική επαφή με τα δεδομένα του προβλήματος. Μια υλοποίηση του για ταξινόμηση, η οποία χρησιμοποιεί στην ξεχωριστή βιβλιοθήκη xgboost της python είναι η εξης:

```
from xgboost import XGBClassifier
classifier = XGBClassifier()
classifier.fit(X_train, y_train)
```

Είναι σημαντικό να αναφερθούν τα εξής για τις τεχνικές boosting, εν γένει:

1. Είναι αρκετά επιρρεπείς σε overfitting, όταν υπάρχει πολύς θόρυβος ειδικά για αλγόριθμους όπως ο Adaboost οι οποίοι λύνουν ένα convex πρόβλημα βελτιστοποίησης.
2. Η εκπαίδευση του μοντέλου είναι χρονοβόρα διότι εκτελείται σειριακά, πόσο μάλλον σε real-time πλατφόρμες όπου επιβάλλεται η παραλληλοποίηση.
3. Στο gradient boosting, σε σχέση με τα random forests, είναι δυσκολότερη η ρύθμιση εξωτερικών παραμέτρων γιατί συνήθως έχουν τρεις : αριθμό δέντρων, βάθος, ρυθμό μάθησης.

2.2.1.3 Η τεχνική cross validation για την αξιολόγηση και βελτιστοποίηση μοντέλων

Στάδιο της μηχανικής μάθησης όπως είδαμε αποτελεί η αξιολόγηση του μοντέλου. Η απλούστερη μορφή αξιολόγησης είναι, όπως είδαμε ,ο διαχωρισμός των δεδομένων σε training και test set με μία αναλογία της τάξης των δύο τρίτων αντίστοιχα. Το μοντέλο εκπαιδεύεται στο training set, αξιολογείται στο test set και στη συνέχεια πραγματοποιείται η ρύθμιση των εξωτερικών παραμέτρων του σταδιακά ώστε το μοντέλο να λειτουργει όσο καλύτερα γίνεται και στα δύο σύνολα.

Η παραπάνω μέθοδος, ωστόσο, είναι αρκετά απλοϊκή για να εξασφαλίσει ότι το μοντέλο δεν είναι overfitted πάνω στο training set ή ότι το test set που επιλέχτηκε δεν έτυχε να είναι αρκετά εύκολο τις προβλέψεις του μοντέλου. Το πρόβλημα αυτό έρχεται να λύσει η τεχνική της διασταυρωμένης επικύρωσης ή cross validation. Πιο χαρακτηριστική μορφή της είναι το k-folds cross validation το οποίο διαρθρώνεται ως εξής:

1. Ανακάτεψε τα δεδομένα, επίλεξε μια τιμή για το k και χώρισε το dataset σε k τμήματα ίδιου μεγέθους.
2. Για κάθε τμήμα κάνε τα εξής:
    a. Θεώρησε το σαν validation set και τα υπόλοιπα k-1 ως training set.
    b. Προσάρμοσε τα δεδομένα του training set στο μοντέλο και αξιολόγησε το μοντέλο στο validation set.

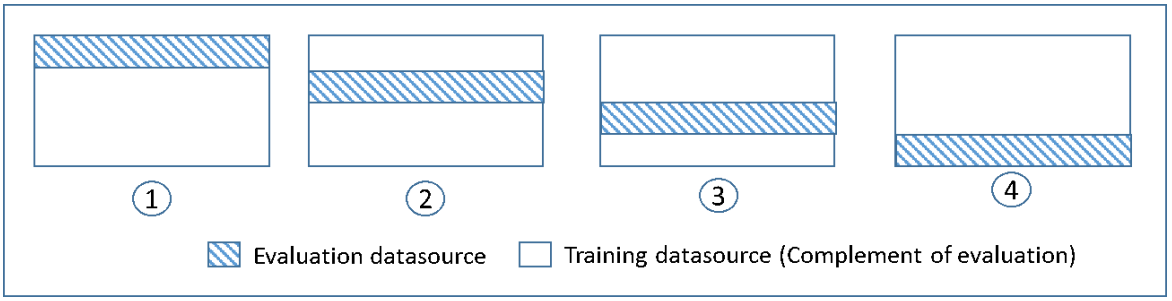
[8] *[Πηγή:https://github.com/scikit-learn/scikit-learn/blob/bac89c2/sklearn/ensemble/weight_boosting.py#L852]*

c. Κράτα μόνο το επιθυμητό αποτέλεσμα (score) της αξιολόγησης (για παράδειγμα μέσο τετραγωνικό σφάλμα), αγνόησε το μοντέλο και συνέχισε στο επόμενο τμήμα.

3. Αξιολόγησε το μοντέλο βάση των επιμέρους αποτελεσμάτων αξιολόγησης που έχεις συγκεντρωσει. (Στο βήμα αυτό συνήθως χρησιμοποιείται ο μέσος όρος των αποτελεσμάτων αξιολόγησης)

Με τον τρόπο αυτό μπορούν να βγουν πολύ πιο ασφαλή, συνεπή και γενικευμένα συμπεράσματα για την απόδοση ενός μοντέλου, καθώς είναι δοκιμασμένο πλέον σε διαφορετικές συνθήκες μάθησης και έχει εξασφαλιστεί πως η όποια απόδοση του δεν οφείλεται στο ότι τροφοδοτείται από ένα συγκεκριμένο συνδυασμό training και test sets.

Το cross-validation μπορεί να χρησιμοποιηθεί ακόμα και για την επιλογή μοντέλου συγκρίνοντας τα αποτελέσματα αξιολογήσεων για διαφορετικούς αλγόριθμους ML.



Εικόνα 2.8: Η τμηματοποίηση του dataset κατα τη διαδικασία 4 folds cross-validation.[9]

2.2.1.4 Εφαρμογές επιτηρούμενης μάθησης

Σε γενικές γραμμές η επιτηρούμενη μάθηση έχει το σκοπό της πρόβλεψης των ετικετών νέων δεδομένων μέσω της μελέτης των ήδη υπαρχόντων δεδομένων με ετικέτες. Χαρακτηριστικές εργασίες επιτηρούμενης μάθησης αποτελεί η παλινδρόμηση, η ταξινόμηση που θα εξεταστούν στα κεφάλαια 3,4 και η υποκατηγορία LDA (linear discriminant analysis) της κατηγορίας του dimensionality reduction. Παραδείγματα εφαρμογής τέτοιων εργασιών είναι τα εξής:

- Προβλήματα παλινδρόμησης όπως αυτό που εκφράζει το dataset του πίνακα 2.1, όπου γίνεται προσπάθεια πρόβλεψης του μισθού ενός υπαλλήλου με συγκεκριμένα χαρακτηριστικά, μέσω της μελέτης μιας συλλογής δεδομένων υπαλλήλων με αντίστοιχες τιμές στα χαρακτηριστικά αυτα και ετικέτες μισθού που τους αντιστοιχούν.
- Προβλήματα ταξινόμησης όπως αυτό που εκφράζει το dataset της εικόνας 3.1, όπου γίνεται προσπάθεια πρόβλεψης της διακριτής τιμής Ναι ή Όχι (εναλλακτικά 0 ή 1) για το αν κάποιος πελάτης με συγκεκριμένα χαρακτηριστικά θα αγοράσει ένα SUV όχημα μιας εταιρείας αυτοκινήτων, μέσω μελέτης μια συλλογής δεδομένων πελατών και μη της εταιρείας με αντίστοιχες τιμές στα χαρακτηριστικά αυτά και διακριτές ετικέτες αγοράς ή όχι που τους αντιστοιχούν.

2.2.2 Μη επιτηρούμενη μάθηση

Στην κατηγορία της μη επιτηρούμενης μηχανικής μάθησης εμπίπτουν προβλήματα των οποίων τα δεδομένα δεν αποτελούν ζεύγη εισόδου - εξόδου. Δεν υπάρχουν δηλαδή ετικέτες οι οποίες να υποδηλώνουν κάποια έξοδο σε κάθε instance. Αντιθέτως, γίνεται προσπάθεια να αντληθούν πληροφορίες ομοιότητας και ανομοιότητας, κρυμμένες δομές και μοτίβα στα πρότυπα - εισόδους. Η

[9] *[Πηγη: https://docs.aws.amazon.com/machine-learning/latest/dg/cross-validation.html]*

μη επιτηρούμενη μάθηση εξυπηρετεί κυρίως στην προεπεξεργασία και τον προσδιορισμό της δομής των δεδομένων και όχι στην πρόβλεψη όπως συμβαίνει στην περίπτωση της επιτηρούμενης μάθησης. Χαρακτηριστικές εργασίες μη επιτηρούμενης μάθησης αποτελούν η συσταδοποίηση (clustering) και η πλειονότητα των τεχνικών dimensionality reduction όπως η PCA (principal component analysis) στις οποίες γίνεται αναφορά σε επόμενα κεφάλαια.

2.2.3 Ημιεπιτηρούμενη μάθηση

Στην κατηγορία ημιεπιτηρούμενης μάθησης (SSL) εμπίπτουν προβλήματα των οποίων τα δεδομένα είναι μερικώς σεσημασμένα με ετικέτες εξόδου. Στο [12] αναλύονται ο χαρακτήρας του SSL οι διάφορες κατηγορίες σχετικών αλγόριθμων και μοντέλων ως εξής:

- Generative μοντέλα: Πρόκειται για μοντέλα που βασίζονται στην από κοινού συνάρτηση πιθανότητας της εξαρτημένης και ανεξάρτητης μεταβλητής. Θα μπορούσε κανείς να αντιμετωπίσει σα μια μορφή συσταδοποίησης με παραπάνω πληροφορίες η ταξινόμησης με πληροφορίες οριακής πυκνότητας πιθανότητας. Επιπλέον, μία ανάλυση για deep generative μοντέλα γίνεται στο [13].
- Μέθοδοι διαχωρισμού χαμηλής πυκνότητας στις οποίες συμπεριλαμβάνονται μοντέλα όπως το TSVM (transductive support vector machine), η ταξινόμηση με δυαδική γκαουσσιανή διαδικασία και προσεγγίσεις μεγιστοποίησης της εντροπίας.
- Μέθοδοι γράφων όπου τα δεδομένα αναπαρίστανται από κόμβους ενός γράφου ενώ οι ακμές του γράφου είναι σεσημασμένες με πιθανοτικά βάρη. Μέσα από αυτό το σύστημα επιτυγχάνεται η διάδοση ετικετών από τα σεσημασμένα στα μη σεσημασμένα δεδομένα με χρήση διακριτών μαρκοβιανών πεδίων, τυχαίων γκαουσιανών πεδίων, αλλα και βαθιών συνελικτικών δικτύων κ.τ.λ
- Μέθοδοι δύο βημάτων όπου αρχικά πραγματοποιείται μια συσταδοποίηση στο σύνολο των δεδομένων και στην πορεία μια ταξινόμηση στα σεσημασμένα δεδομένα. Οι μέθοδοι αυτές έχουν στενή σύνδεση με εκείνες των γράφων.

Η ανάγκη για SSL προκύπτει καθώς η σήμανση δεδομένων αποτελεί κατά κανόνα μία δύσκολη εργασία η οποία απαιτεί την έντονη συμβολή του ανθρώπινου παράγοντα πράγμα χρονοβόρο και κοστοβόρο. Χαρακτηριστικές εφαρμογές του SSL είναι η ταξινόμηση ακολουθιών πρωτεϊνών και η αναγνώριση ομιλίας.

2.2.4 Ενισχυτική μάθηση

Η ενισχυτική μάθηση - Reinforcement Learning (RL) αποτελεί ένα είδος μάθησης (κοινώς μια απεικόνιση καταστάσεων σε δράσεις) στο οποίο ο σκοπός είναι η μεγιστοποίηση ενός σήματος επιβράβευσης [14]. Απαιτείται, λοιπόν, η ύπαρξη ενός πράκτορα (agent) ο οποίος διέπεται από τα εξής θεμελιώδη χαρακτηριστικά:

- Έχει στόχο.
- Έχει αίσθηση του περιβάλλοντος του (π.χ μέσω αισθητήρων) ώστε να μπορεί να αντιληφθεί τη συνέπεια των πράξεων του στο περιβάλλον του και, κατα συνέπεια, στην εξυπηρέτηση του στόχου του.
- Δύναται να λάβει αποφάσεις και να δράσει αναλόγως με βάση τα παραπάνω..

Ο πράκτορας μεταβαίνει από κατάσταση σε κατάσταση μέσω λήψης αποφάσεων. Οι αποφάσεις αυτές επιβραβεύονται η τιμωρούνται ανάλογα με την επίδραση που έχουν στην επίτευξη του στόχου μέσω ενός αντίστοιχου σήματος. Η διαδικασία παλινδρομεί διαρκώς ανάμεσα στις έννοιες τις εκμετάλλευσης και της εξερεύνησης (exploitation και exploration dilemma όπως αναφέρεται στο

[14]). Συνεπώς, ο πράκτορας, κατά προσπάθεια μεγιστοποίησης της ανταμοιβής έχει αφενός συμφέρον να ακολουθεί μονοπάτια αποφάσεων τα οποία ακολούθησε στο παρελθόν και αποδείχθηκαν αποτελεσματικά σε όρους επιβράβευσης και αφετέρου για να ανακαλύψει τέτοια μονοπάτια οφείλει να επιλέγει δράσεις τις οποίες δεν έχει ξαναεπιλέξει στο παρελθόν.

Τελικα, η ενισχυτική μάθηση δεν ορίζεται από μεθόδους επίλυσης προβλημάτων όπως για παράδειγμα η επιτηρούμενη μάθηση, αλλά από την παροχή μιας πλήρους περιγραφής ενός προβλήματος σε ένα πράκτορα με τα προαναφερθέντα χαρακτηριστικά. Το πρόβλημα συνήθως ορίζεται από μία Μαρκοβιανή στοχαστική διαδικασία αποφάσεων, η οποία ανάγεται σε πρόβλημα βελτιστοποίησης γραμμικού ή και δυναμικού προγραμματισμού. Ο πράκτορας κινείται ανάμεσα σε ένα σύνολο καταστάσεων S με ένα σύνολο δράσεων A για κάθε κατάσταση. Κάθε του επιλογή επιβραβεύεται η ποινικοποιείται. Τελικός σκοπός είναι η εξαγωγή μιας πολιτικής βέλτιστων μεταβάσεων (δράσεων) από κατάσταση σε κατάσταση για τον πράκτορα. Περαιτέρω ανάλυση και μαθηματικές θεμελιώσεις βρίσκει κανείς στο [14], καθώς δε θα παρατεθούν στα πλαίσια της διπλωματικής αυτής .

2.2.4.1 Εφαρμογές ενισχυτικής μάθησης

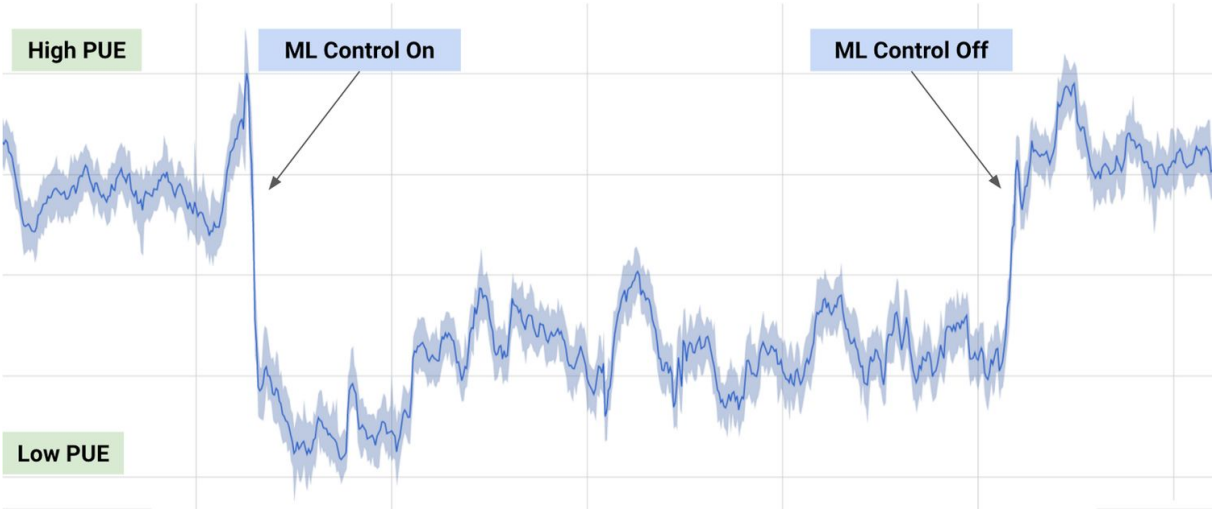
Χαρακτηριστικό παράδειγμα RL αποτελεί ένα ρομπότ το οποίο μαζεύει σκουπίδια και καλείται να αποφασίσει αν μπορεί να μεταβεί στον επόμενο χώρο για να συλλεξει νέα σκουπίδια ή πρέπει προηγουμένως να επισκεφτεί το σταθμό φόρτισης της μπαταρίας του. Η απόφαση αυτή (δράση) λαμβάνεται αρχικά βάσει της στάθμης της μπαταρίας του και βάσει του πόσο γρήγορα έχει βρεί σταθμό φόρτισης στο παρελθόν δεδομένης της τοποθεσίας του (αίσθηση του περιβάλλοντος), στα πλαίσια του στόχου να ολοκληρώσει τη συλλογη σκουπιδιών, ο οποίος απαιτεί να μην ξεμείνει από μπαταρία σε καμία περίπτωση. Γίνεται, λοιπόν, εύκολα αντιληπτό πως μια ενδεχόμενη απώλεια μπαταρίας συνεπάγεται μεγάλη ποινή στο σύστημα επιβράβευσης του πράκτορα.

Ένα δεύτερο παράδειγμα RL στο οποίο μπορεί κανείς να εντοπίσει το δίλημμα εκμετάλλευσης-εξερεύνησης που αναφέρθηκε παραπάνω είναι το multi armed bandit problem το οποίο μάλιστα έχει και πολλαπλές εφαρμογές σε προβλήματα του πραγματικού κόσμου:
Στην πιο απλή του μορφή μπορούμε να φανταστούμε μερικές φαινομενικά όμοιες μηχανές τυχερών παιχνιδιών, για παράδειγμα 5 μηχανές τύπου φρουτάκια. Προσπαθούμε να ανακαλύψουμε ποια είναι η πιο κερδοφόρα, όμως ταυτόχρονα ποντάρουμε χρήματα οπότε επιβάλλεται όλη αυτή η διαδικασία να γίνει με τα ελάχιστα δυνατά χαμένα χρήματα. Το συγκεκριμένο πρόβλημα μπορεί να αντιμετωπιστεί με στοιχειώδεις αλγόριθμους ενισχυτικής μάθησης όπως ο upper confidence bound, ο thompson sampling, ο softmax, ε-greedy κ.α των οποίων οι αποδόσεις ποικίλλουν ανάλογα με το είδος και τις παραμέτρους του προβλήματος . Σε κάθε περίπτωση γίνεται προσπάθεια σταδιακής εύρεσης της μηχανής με το μέγιστο αναμενόμενο χρησιμοποιώντας ταυτόχρονα όσο πιο πολύ γίνεται τις πιο κερδοφόρες μηχανές στην πορεία αυτη. Το multi-armed bandit problem βρίσκει εφαρμογή σε ζητήματα όπως αυτό της επιλογής της καλύτερης από μερικές διαφημιστικές καμπάνιες μιας εταιρείας δοκιμάζοντας την απήχηση όλων στον κόσμο (π.χ με βάση τα κλικ) και τελικά καταλήγοντας σε κατάργηση όλων πέραν της πιο αποτελεσματικής. Παρομοίως, εφαρμογή βρίσκει και σε κλινικές μελέτες για την επιλογή κατάλληλων θεραπειών σε ασθενείς όπως αναφέρεται στο [15].

Μία ακόμη πολύ κλασική περίπτωση εφαρμογής του RL είναι τα βιντεοπαιχνίδια. Ο πράκτορας δηλαδη εκπαιδεύεται στο να παίζει όλο και πιο αποτελεσματικά ένα ηλεκτρονικό παιχνίδι. Συνήθης αλγόριθμος στον κλάδο είναι αυτός του Q-learning, που θεμελιώνεται στο [16] ενώ πολύ πρόσφατο state-of-the-art αλγόριθμο αποτελεί ο Double DQN (Double Deep Q-Network) ο οποίος προτάθηκε στο [17] από την πρωτοπόρο εταιρεία Deepmind της Google και δοκιμάστηκε σε παιχνίδια

της γενιάς Atari 2600 και αποτελεί εξέλιξη του DQN[18]. Αλλοι γνωστοι αλγόριθμοι RL είναι οι SARSA (βλ.[19]) και DDPG.

Σημειώνεται πως η Google Deepmind, τροφοδοτώντας νευρωνικά δίκτυα με μετρήσεις αισθητήρων στα κεντρικά κτήρια πληροφοριών της Google (Google Data Centres) κατάφερε να μειώσει κατα 40% την κατανάλωση ενέργειας για την ψύξη των κτηρίων αυτών όπως φαίνεται και στο γράφημα της εικόνας 2.9. Στο γράφημα αναπαρίσταται ο λόγος της συνολικής κατανάλωσης ενέργειας προς την κατανάλωση του IT τμήματος του κτηρίου (PUE). Τρία νευρωνικά εκπαιδεύτηκαν ώστε να κάνουν πρόβλεψη του PUE, της θερμοκρασίας και της πίεσης εντός του κτηρίου υπολογιστών, ρυθμίζοντας αναλόγως το σύστημα ψύξης. Με ενεργοποιημένο τον έλεγχο από ML η εξοικονόμηση που παρατηρείται αντιστοιχεί σε μείωση 40% στην ψυκτική ισχύ που απαιτεί το κτήριο.



Εικόνα 2.9: Ο λόγος της συνολικής κατανάλωσης ενέργειας προς την κατανάλωση του IT τμήματος του κτηρίου (PUE).[10]

[10] *[Πηγή:https://deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-40/]*

3.Ταξινόμηση

3.1 Ταξινόμηση στη μηχανική μάθηση

Στη μηχανική μάθηση ταξινόμηση ονομάζεται η διαδικασία κατά την οποία ένας αλγόριθμος (ταξινομητής-classifier) εκπαιδεύεται πάνω σε δεδομένα τα οποία χαρακτηρίζονται από συγκεκριμένες ετικέτες, οι οποίες υποδεικνύουν την κλάση τους, και μαθαίνει, με αυτό τον τρόπο, να ταξινομεί νέα δεδομένα στις κλάσεις αυτές. Στην πραγματικότητα, η ταξινόμηση αποτελεί μια διαδικασία εκτίμησης μιας συνάρτησης στόχου f, όπως την είδαμε στην ενότητα 2.2.1, η οποία αντιστοιχίζει διανύσματα γνωρισμάτων (ανεξάρτητες μεταβλητές) εισόδου $\mathbf{X} = \{x_1,...,x_k\}$ σε διακριτή έξοδο η οποία παίρνει τιμές από ένα σύνολο $y = \{y_1,...,y_m\}$ όπου m ο αριθμός των κλάσεων, k ο αριθμός των γνωρισμάτων. Πρόκειται προφανώς για εργασία επιτηρούμενης μάθησης.

3.2 Είδη και παραδείγματα ταξινόμησης

Ένα παράδειγμα ταξινόμησης είναι η εξαγωγή απόφασης για τον αν κάποιος θα αγόραζε ένα συγκεκριμένο μοντέλο το οποίο παράγει μία εταιρία κρίνοντας από το εισόδημα του, το φύλλο και την ηλικία του. Η έξοδος που περιμένουμε είναι 0 ή 1, δηλαδή αν θα το αγοράσει ή όχι και το συγκεκριμένο αποτελεί πρόβλημα δυαδικής ταξινόμησης (binary classification) εφόσον η διακριτή έξοδος παίρνει δύο τιμές. Για παράδειγμα στην εικόνα 3.1 παρουσιάζεται η δομή ενός dataset δυαδικής παλινδρόμησης για την αγορά ενος αυτοκινήτου. Οι πρώτες 4 στήλες είναι τα γνωρίσματα των πελατών και η τελευταία είναι το αν αγόρασαν τελικά ή όχι το αυτοκίνητο. Η πρώτη στήλη προφανώς δε θα πρέπει να συμμετάσχει στην εξαγωγή μοντέλου και την πρόβλεψη, καθώς δε μπορεί να συσχετίζεται με την επιθυμητή έξοδο. Ένα τέτοιου είδους dataset μπορεί να βοηθήσει μια εταιρεία να κατανοήσει τα γνωρίσματα των εν δυνάμει πελατών της και να προχωρήσει σε πιο αποτελεσματικές και στοχευμένες διαφημίσεις ανάλογα με το αν εκείνοι έχουν τις προοπτικές να αγοράσουν κάποιο προϊόν της χωρίς έτσι να χρεώνεται για άκαρπες διαφημίσεις σε ομάδες που, εν γένει, δε δείχνουν ενδιαφέρον.

Αλλο παράδειγμα είναι ένα πρόβλημα εξαγωγής συμπεράσματος για τη ράτσα σκύλων που προβάλλεται σε μία φωτογραφία ενδέχεται να πρέπει να επιλέξουμε ανάμεσα σε παραπάνω κλάσεις όπως bulldog, golden retriever, labrador, pitbull. Σ αυτή την περίπτωση, η ταξινόμηση είναι πολλαπλών κλάσεων. Υπάρχουν και προβλήματα ταξινόμησης πολλαπλών ετικετών (multi label classification) όπου κάθε παρατήρηση χρήζει αντιστοίχισης σε παραπάνω από μία ετικέτα.

Index	User ID	Gender	Age	EstimatedSalary	Purchased
7	15694829	Female	32	150000	1
8	15600575	Male	25	33000	0
9	15727311	Female	35	65000	0
10	15570769	Female	26	80000	0
11	15606274	Female	26	52000	0
12	15746139	Male	20	86000	0
13	15704987	Male	32	18000	0
14	15628972	Male	18	82000	0
15	15697686	Male	29	80000	0
16	15733883	Male	47	25000	1
17	15617482	Male	45	26000	1
18	15704583	Male	46	28000	1

Εικόνα 3.1: Παράδειγμα δομής ενός dataset δυαδικής παλινδρόμησης για την αγορά ενός αυτοκινήτου.

3.3 Αλγόριθμοι και τεχνικές ταξινόμησης

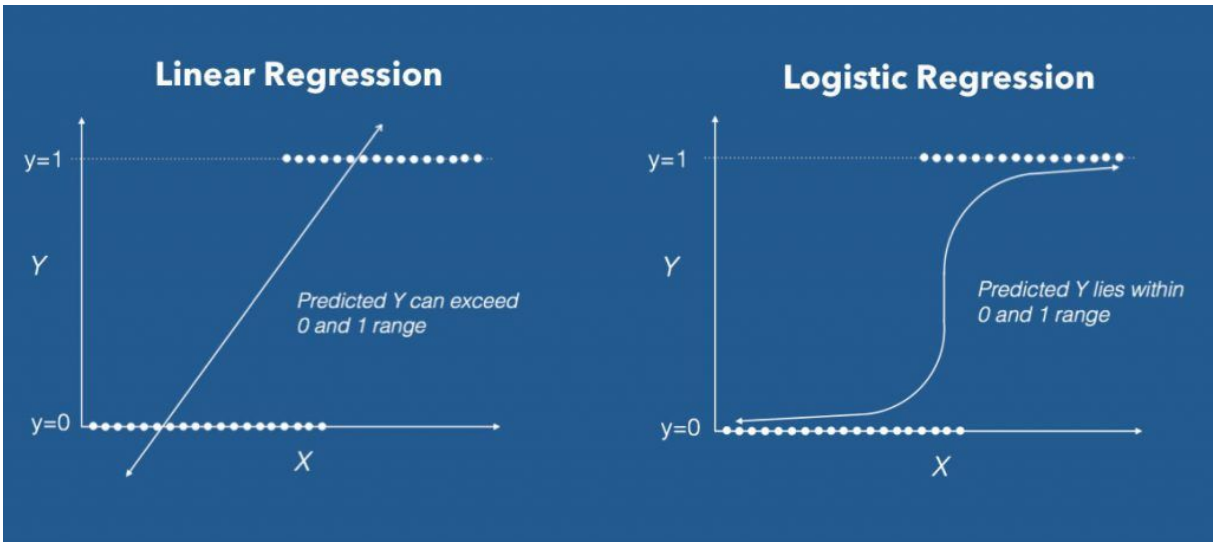
3.3.1 Λογιστική παλινδρόμηση (Logistic regression)

Η λογιστική παλινδρόμηση αποτελεί μια επέκταση παλινδρόμησης η οποία, ωστόσο, έχει εφαρμογή σε προβλήματα δυαδικής ταξινόμησης γι' αυτό το λόγο και παρατίθεται στην ενότητα αυτή. Η εξαρτημένη μεταβλητή παίρνει τις δυαδικές τιμές (0,1), ωστόσο κατασκευάζεται ένα μοντέλο γραμμικής παλινδρόμησης. Στη συνέχεια εφαρμόζεται ο μετασχηματισμός 3.1 στην εξαρτημένη μεταβλητή έτσι ώστε τελικά στον κατακόρυφο άξονα να έχουμε πλέον αναπαράσταση της πιθανότητας της κλάσης με τιμή 1 ως εξής:

$$Z_i = ln\left(\frac{P_i}{1-P_i}\right) = \alpha + \beta_1 x_1 + .. + \beta_n x_n \qquad \text{(Εξίσωση 3.1)}$$

$$P_i = E(y=1|x_i) = \frac{e^z}{1+e^z} = \frac{e^{\alpha+\beta_i x_i}}{1+e^{\alpha+\beta_i x_i}} \qquad \text{(Εξίσωση 3.2)}$$

Συνήθως τιμές από το κατώφλι πιθανότητας 0.5 και πάνω μεταφράζονται ως πρόβλεψη για την κλάση 1. Για τα δεδομένα του δυαδικού προβλήματος της ενότητας 3.2 αυτό μεταφράζεται σε πρόβλεψη αγοράς του αυτοκινήτου. Ωστόσο, είναι πολύ χρήσιμο σε ορισμένα προβλήματα το γεγονός ότι γνωρίζουμε τις πιθανότητες της ταξινόμησης και μπορούμε να έχουμε μια πιο ρεαλιστική οπτική της κατάστασης. Επιπλέον, είναι στο χέρι μας να κάνουμε το μοντέλο πιο αυστηρό η πιο χαλαρό, αυξομειώνοντας το επιθυμητό κατώφλι (threshold).



Εικόνα 3.2: Μια γραφική αναπαράσταση του μετασχηματισμού λογιστικής παλινδρόμησης. [11]

Η κλάση της βιβλιοθήκης scikit-learn που υλοποιεί τον αλγόριθμο σε python:
class sklearn.linear_model.**LogisticRegression**(*penalty='l2'*, *dual=False*, *tol=0.0001*, *C=1.0*, *fit_intercept=True*, *intercept_scaling=1*, *class_weight=None*, *random_state=None*, *solver='warn'*, *max_iter=100*, *multi_class='warn'*, *verbose=0*, *warm_start=False*, *n_jobs=None*)

3.3.2 Instance-based τεχνικές ταξινόμησης

Η κατηγορία αυτή τεχνικών μηχανικής μάθησης βασίζεται ατομικά σε κάθε, προς ταξινόμηση αντικείμενο, όπως υποδηλώνει και ο τίτλος της. Αυτό σημαίνει πως το κομμάτι της εκπαίδευσης απαιτεί ελάχιστο χρόνο και υπολογιστική ισχύ, ενώ όλες οι βασικές εργασίες γίνονται κατά τη διάρκεια της διαδικασίας ταξινόμησης εξού και ο χαρακτηρισμός lazy-learning algorithms του [20] .

3.3.2.1 Αλγόριθμος k πλησιέστερων γειτόνων - kNN

Αντιπροσωπευτικός αλγόριθμος είναι αυτός των k-πλησιέστερων γειτόνων ή k-nearest neighbors (kNN) οποίος απαιτεί την επιλογή ενός μέτρου απόστασης (π.χ. Ευκλείδια απόσταση, απόσταση Chebychev, σταθμισμένες αποστάσεις κ.α).

3.3.2.1.1 Βήματα αλγόριθμου

Έστω ότι έχουμε το dataset με τα δοθέντα διανύσματα γνωρισμάτων και τις ετικέτες. Επιλέγεται ο αριθμό k των κοντινότερων γειτόνων και το επιθυμητό μέτρο απόστασης. Κάθε νέο διάνυσμα Α κατηγοριοποιείται ως εξής:

1. Επίλεξε τους k κοντινότερους γείτονες του Α με βάση το μέτρο απόστασης
2. Από αυτούς μέτρα πόσοι ανήκουν σε κάθε κατηγορία.
3. Ανάθεσε στο νέο σημείο Α την ετικέτα της κατηγορίας στην οποία ανήκουν οι περισσότεροι εκ των γειτόνων.

3.3.2.1.2 Παρατηρήσεις kNN

Τα βασικά μειονεκτήματα αυτου του είδους αλγόριθμων, σύμφωνα με τα [21],[22], είναι τα εξής:

[11] *[Πηγή: https://www.machinelearningplus.com/machine-learning/logistic-regression-tutorial-examples-r/]*

- ❏ Οι μεγάλες απαιτήσεις σε χρόνο υπολογισμού και μνήμη κατα τη διαδικασία της ταξινόμησης
- ❏ Η απουσία γρήγορου συστηματικού τρόπου εύρεσης κατάλληλου k. Μπορεί να χρησιμοποιηθεί cross validation τεχνική, η οποία μαζί με άλλες κατατάσσεται σε ιδιαίτερα χρονοβόρες και υπολογιστικά απαιτητικές διαδικασίες.
- ❏ Η ευαισθησία στην επιλογή μέτρου απόστασης.

Συνεπώς καλό είναι να αποφεύγεται η χρήση του όταν τα δεδομένα είναι πολλά. Τα πλεονέκτημα του είναι η απλότητα του και η δυνατότητα του να λειτουργήσει για σύνορα κλάσεων με ακαθόριστους σχηματισμούς στο χώρο σε αντίθεση για παράδειγμα με τον αλγόριθμο ταξινόμησης Naive-Bayes.

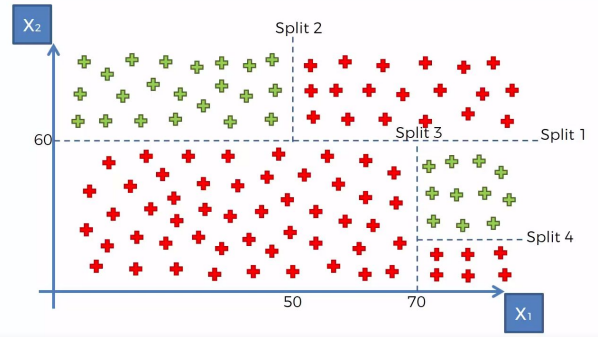
3.3.2.1.3 Κώδικας

Η κλάση της βιβλιοθήκης scikit-learn που υλοποιεί τον αλγόριθμο σε python:
class sklearn.neighbors.**KNeighborsClassifier***(n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=None, **kwargs)*

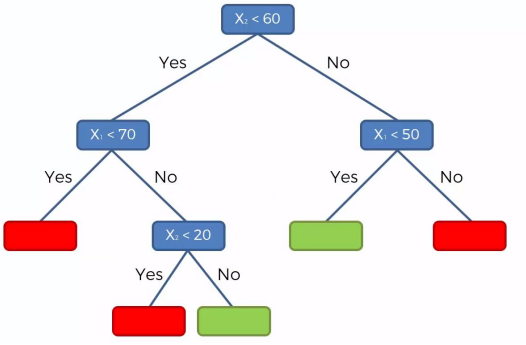
3.3.3 Τεχνικές βασισμένες σε λογικούς κανόνες.

3.3.3.1 Ταξινόμηση με δέντρα αποφάσεων

Τα δέντρα αποφάσεων αποτελούν την πιο χαρακτηριστική περίπτωση τεχνικών εκμάθησης κανόνων. Η λογική στην οποία βασίζονται είναι η εξής:
Αναζητείται το γνώρισμα (feature) το οποίο διαμερίζει με τον καλύτερο δυνατό τρόπο τα δεδομένα σύμφωνα με μετρικές όπως το gini index [23] η το κέρδος πληροφορίας [24] Το γνώρισμα αυτό αποτελεί τη ρίζα του δέντρου αποφάσεων και ταυτόχρονα ένα κόμβο απόφασης που οδηγεί σε διαφορετικές επιλογές ανάλογα με τις τιμές που παίρνει το γνώρισμα . Στη συνέχεια το δέντρο κατασκευάζεται με ανάλογο τρόπο, ορίζοντας έτσι περιοχές οι οποίες αντιστοιχούν σε κλάσεις (βλ. εικόνες 3.3, 3.4). Τα φύλλα του δέντρου πάντοτε ορίζουν μια περιοχή μιας συγκεκριμένης κλάσης.



Εικόνα 3.3: Διαχωρισμός περιοχών κλάσεων από ένα δέντρο αποφάσεων.



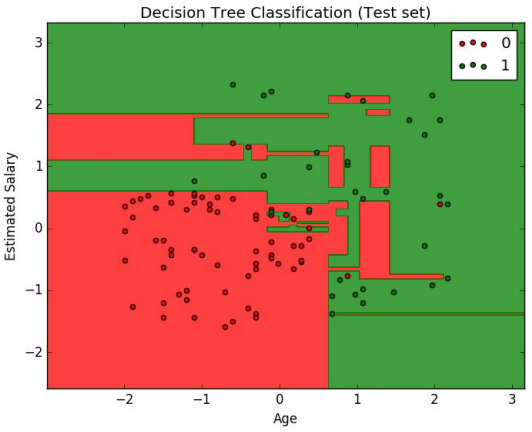
Εικόνα 3.4: Δομή του δέντρου αποφάσεων.

3.3.3.1.1 Παρατηρήσεις στα δέντρα αποφάσεων

Τα δέντρα αποφάσεων έχουν την κακή φήμη του overfitting ειδικά στην περίπτωση που είναι πλήρως αναπτυγμένα. Γι' αυτο στη βιβλιογραφία, υπάρχουν πολλές και διαφορετικές μέθοδοι "κλαδέματος" (pruning) με τις οποίες ναι μεν περιορίζουμε την απόδοση του δεντρου στο training set,

εντούτοις η απόδοση του σε νέα δεδομένα βελτιώνεται καθώς δεν είναι επακριβώς προσαρμοσμένη στο training set και σε ότι θόρυβο φέρει αυτο όπως φαίνεται στις εικόνες 3.5, 3.6. Σε γενικές γραμμές, πρόκειται για μια παρωχημένη μέθοδο η οποία μάλιστα είχε σταματήσει να έχει εφαρμογή μέχρι πρόσφατα οπότε και επανήλθε με αναβαθμίσεις όπως το gradient boosting και το random forest που θα δούμε παρακάτω κ.α.

Εικόνα 3.5: Η εκπαίδευση του αλγόριθμου δέντρου αποφάσεων πάνω σε πραγματικά δεδομένα.(έντονο overfitting)



Εικόνα 3.6: Η αξιολόγηση του αλγόριθμου δέντρου αποφάσεων στο test set.

3.3.3.1.2 Τυχαίο δάσος (Random Forest)

Η ταξινόμηση τυχαίου δάσους αποτελεί μία επέκταση των απλών δέντρων αποφάσεων στα πλαίσια του συνδυασμού μεθόδων (ensemble learning) και πιο συγκεκριμένα του bagging που είδαμε στην ενότητα 2.2.1.3.1. Ο αλγόριθμος τυχαίου δάσους διαρθρώνεται ως εξής:

1. Επίλεξε k τυχαία σημεία από το training set.
2. Κατασκεύασε το δέντρο απόφασης που αφορά στα k αυτά σημεία
3. Επίλεξε τον επιθυμητό αριθμό δέντρων αποφάσεων και επανάλαβε τα βήματα 1,2
4. Ένα νέο σημείο πληροφορίας, κατάταξε το στην κλάση που επιτάσσει η πλειοψηφία των δέντρων αποφάσεων.

Τα πλεονεκτήματα του αλγόριθμου αυτού είναι τα εξής:

- Προσφέρει έναν πολύ πιο ομαλό και αποτελεσματικό έλεγχο του bias-variance tradeoff σε σχέση με τα decision trees. Περιορίζει το overfitting που παρουσιάζει ένα πλήρως αναπτυγμένο δέντρο, ενώ από την άλλη παρουσιάζει μεγαλύτερη μεταβλητότητα από κάποιο ρηχό ή υποανάπτυκτο δέντρο.
- Πετυχαίνει μεγαλύτερη ευστοχία, εισάγοντας στην πρόβλεψη την αντικειμενικότητα παραπάνω δέντρων. Αν κάποιο δέντρο για παράδειγμα έχει “παρασυρθεί” από κάποιο outlier, το φαινόμενο εξομαλύνεται αφού η απόφαση του σταθμίζεται και με των άλλων δέντρων.
- Συνεχίζει να είναι επιρρεπής σε overfitting, ωστόσο αισθητά λιγότερο σε σχέση με μεθόδους όπως το Gradient Tree Boosting που είδαμε στην ενότητα 2.2.1 του SL.
- Τα random forests είναι εύκολο να παραλληλοποιηθούν σε real-time cloud πλατφόρμες κατά τη μάθηση, καθώς κάθε δέντρο εκπαιδεύεται ανεξάρτητα. Υπάρχουν και σχετικές πρόσφατες υλοποιήσεις όπως είναι αυτή του [25] σε Apache Spark.

- Ο ίδιος αλγόριθμος μπορεί να χρησιμοποιηθεί και σε προβλήματα παλινδρόμησης. Αντί για ψηφοφορία γίνεται χρήση μέσης τιμής των προβλέψεων κάθε δέντρου.

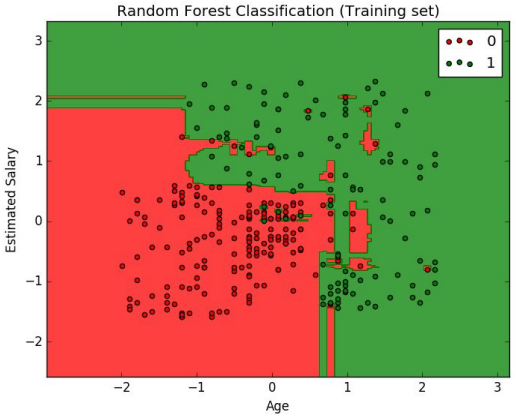
Τα μειονεκτήμα του αλγόριθμου Random Forest είναι τα εξής:

1. Είναι αισθητά πιο αργός κατά εκτέλεση του σε μεγάλα δεδομένα, όπου κατασκευάζονται πολλά δέντρα αποφάσεων.
2. Σε προβλήματα με κατηγορικές μεταβλητές πολλών επιπέδων τείνει να είναι biased προς αυτές, γεγονός που τον καθιστά αναποτελεσματικό.

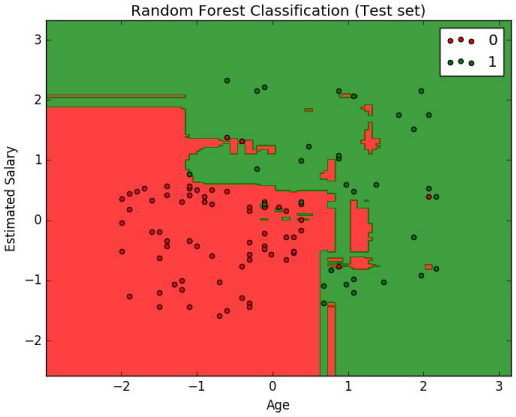
3.3.3.1.3 Συμπεράσματα για τα Random Forests

Χαρακτηριστικό παράδειγμα της αποτελεσματικότητας των Random Forests και απόδειξη ότι αποτελεί ένα state-of-the-art εργαλείο είναι η χρήση τους για αναγνώριση κίνησης στην πλατφόρμα βιντεοπαιχνιδιών Kinect της Microsoft με τον τρόπο που περιγράφεται στο [26].

Στις εικόνες 3.7, 3.8 βλέπουμε τα βελτιωμένα αποτελέσματα του αλγόριθμου Random Forest στα ίδια δεδομένα που είδαμε στις εικόνες 3.5, 3.6 της ενότητας 3.3.3.1.1**.** Είναι εμφανές ότι το overfitting είναι περιορισμένο σε σχέση με τα απλά δέντρα.



Εικόνα 3.7: Η εκπαίδευση του αλγόριθμου random forest πάνω σε πραγματικά δεδομένα. (μειωμένο overfitting)



Εικόνα 3.8: Η αξιολόγηση του αλγόριθμου δέντρου αποφάσεων στο test set.

3.3.3.1.4 Κώδικας

Η κλάση της βιβλιοθήκης scikit-learn που υλοποιεί τους αλγόριθμους σε python:

class sklearn.tree.**DecisionTreeClassifier**(*criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, class_weight=None, presort=False*)

class sklearn.ensemble.**RandomForestClassifier**(*n_estimators='warn', criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None*)

3.3.3.2 Μάθηση συνόλου κανόνων

Η τεχνική αυτή μάθησης διέπεται από προσπάθεια περιγραφής του training set με τη χρήση λογικών συμβόλων και κανόνων ταξινόμησης. Η περιγραφή κάθε κλάσης είναι δηλαδή της μορφής:

$$(X_1 \wedge X_2 \wedge \ldots \wedge X_n) \vee (X_n+1 \wedge X_{n+2} \wedge \ldots X_{2n}) \vee \ldots \vee (X_{(k-1)n+1} \wedge X_{(k-1)n+2} \wedge \ldots \wedge X_{kn})$$

Ο σκοπός είναι να κατασκευάσουμε μια περιγραφή των κλάσεων με τους λιγότερους δυνατούς κανόνες διότι το αν το πλήθος τους είναι πολύ μεγάλο, αυτό σημαίνει πως αλγόριθμος αυτό που κάνει είναι να προσπαθεί απλώς να απομνημονεύσει το training set [21]. Στην περίπτωση αυτή εγκυμονεί προφανώς ο κίνδυνος του overfitting. Σημειώνεται κάθε δέντρο αποφάσεων να εκφραστεί μέσω ενός συνόλου κανόνων από τη ρίζα προς κάθε φύλλο.

3.3.4 Μηχανές διανυσμάτων στήριξης (SVM)

Η μηχανή διανυσμάτων στήριξης αποτελεί μια από τις νεότερες μεθόδους ταξινόμησης στην οποία μάλιστα γίνεται πρώτη φορά αναφορά το 1995 στο [27]. Η μέθοδος αυτή βασίζεται στην έννοια του μέγιστου περιθωρίου ή maximum margin (συμβολισμός M). Η λογική του αλγόριθμου είναι προσδιοριστεί το υπερεπίπεδο το οποίο διαχωρίζει τις παρατηρήσεις στις δοθείσες κλάσεις με τρόπο ώστε η πιο κοντινή προς το υπερεπίπεδο παρατήρηση της μίας κλάσης να απέχει το μέγιστο δυνατόν από την αντίστοιχη παρατήρηση της "απέναντι" κλάσης. Στην εικόνα 3.8, εξετάζεται η λειτουργία του αλγόριθμου για την περίπτωση της ταξινόμησης δύο γραμμικά διαχωριζόμενων κλάσεων. Για την πραγματοποίηση του διαχωρισμού αυτού, απαιτούνται πολύπλοκα μαθηματικά τα οποία βασίζονται συνήθως στην ιδέα της ελαχιστοποίησης τετραγωνικής συνάρτησης (quadratic programming) υπό γραμμικούς περιορισμούς.

Συνοπτικά, για δύο κλάσεις διανυσμάτων $\mathbf{x}_i$,οι οποίες είναι γραμμικά διαχωριζόμενες, αποδεικνύεται ότι υπάρχει ένα ζεύγος (**w**,b) τέτοιο ώστε:

$$\boldsymbol{w}^T \cdot \boldsymbol{x}_i + b \geq 1, \quad \forall x_i \in A \qquad \text{(Εξίσωση 3.3)}$$

$$\boldsymbol{w}^T \cdot \boldsymbol{x}_i + b \leq -1, \quad \forall x_i \in B \qquad \text{(Εξίσωση 3.4)}$$

με το πρόσημο της παράστασης $\mathbf{w}^T\mathbf{x}_i$+ b να αποφασίζει για την ετικέτα κάθε νέου διανύσματος. Τα σημεία πληροφορίας $\mathbf{x}_i$ που ικανοποιούν την ισότητα αποτελούν τα διανύσματα στήριξης. Το παραπάνω ανάγεται στο εξής πρόβλημα ελαχιστοποίησης:

Συνάρτηση ελαχιστοποίησης:

$$f(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2 \qquad \text{(Εξίσωση 3.5)}$$

υπό τον περιορισμό

$$y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i + b\right) \geq 1 \qquad \text{(Εξίσωση 3.6)}$$

όπου y_i οι ετικέτες κλάσεων που εναλλάσσονται στις τιμές -1, 1 για τα στοιχεία του dataset.

3.3.4.1 Τροποποιήσεις και επεκτάσεις του αλγόριθμου

Όπως φαίνεται από τα παραπάνω, η φύση του αλγόριθμου παραπέμπει σε προβλήματα δυαδικής ταξινόμησης ωστόσο τα δεδομένα και οι περιορισμοί ελαχιστοποίησης μπορούν να τροποποιηθούν για την επίλυση προβλημάτων ταξινόμησης πολλαπλών κλάσεων, μια σύγκριση των οποίων μπορεί να δει κανείς αναλυτικά στο [28]
Επιπλέον, τα δεδομένα ποτέ δεν είναι πραγματικά γραμμικά διαχωριζόμενα σε κλάσεις, οπότε προκύπτουν στη βιβλιογραφία οι εξής τροποποιήσεις:

- Soft Margin: Στην περίπτωση θορύβου (βλ. εικόνα 3.8), γίνεται η χρήση soft margin η όπως προτείνεται στα [27], [29] και τροποποιεί τους περιορισμούς του προβλήματος ελαχιστοποίησης. Οι νέοι περιορισμοί επιτρέπουν να υπάρξουν μερικά σφάλματα ταξινόμησης ξ_i ,κατα τη μάθηση για τα δεδομένα - θόρυβο τα οποία δεν ικανοποιούν την υποτιθέμενη γραμμικότητα καθώς με την αυστηρή έννοια δε θα μπορούσε καν να βρεθεί το ζητούμενο υπερεπίπεδο. Αναζητείται δηλαδή ο καλύτερος συνδυασμός ελαχιστοποίησης των ξ_i, και του σφάλματος εκπαίδευσης του αλγόριθμους εισάγοντας παράγοντα κανονικοποίησης και χαλαρώνοντας τους περιορισμούς ως εξής:

$$f(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C \cdot \sum \xi_i \qquad \text{(Εξίσωση 3.7)}$$

$$y_i\left(\boldsymbol{w}^T \boldsymbol{x}_i + b\right) \geq 1 - \xi_i,\ \xi_i \geq 0 \qquad \text{(Εξίσωση 3.8)}$$

- Απεικόνιση σε χώρο περισσότερων διαστάσεων: Στην περίπτωση γενικευμένης μη γραμμικότητας του προβλήματος, μπορεί να πραγματοποιηθεί κατάλληλος μετασχηματισμός των δεδομένων προκειμένου γίνει επεξεργασία τους σε ένα χώρο περισσότερων διαστάσεων στον οποίο και είναι γραμμικά διαχωρίσιμα. Ωστόσο, η τεχνική αυτή δε συνίσταται καθώς όλοι οι υπολογισμοί του προβλήματος πραγματοποιούνται σε παραπάνω διαστάσεις από τις ήδη δοθείσες, γεγονός που καθιστά την επίλυση εξαντλητική ως προς τους επεξεργαστικούς πόρους.
- Kernel SVM: Ως λύση στο παραπάνω πρόβλημα, προτείνεται η χρήση των λεγόμενων συναρτήσεων πυρήνα ή kernel functions. Στην περίπτωση αυτή χρησιμοποιούμε μια η περισσότερες συναρτήσεις πυρήνα για να κατασκευάσουμε μη γραμμικά σύνορα για τις κλάσεις. Ουσιαστικά πρόκειται για ιδέα βασισμένη στην απεικόνιση σε χώρο παραπάνω διαστάσεων. Ωστόσο οι υπολογισμοί πραγματοποιούνται στον αρχικό αριθμό διαστάσεων μέσω χρήσης των συναρτήσεων πυρήνα (kernel trick), δεδομένου ότι είναι απαγορευτικό να προσδώσουμε πολυπλοκότητα παραπάνω διαστάσεων στους υπολογισμούς μας. Το βασικό μειονέκτημα είναι ο μεγάλος χρόνος που απαιτείται για την εκπαίδευση του αλγόριθμου αν και υπάρχουν αλγόριθμοι όπως ο αριθμητικός SMO [30] και βελτιώσεις του [31], που την επιταχύνουν. Οι πιο τυπικές συναρτήσεις πυρήνα είναι οι εξής:

  - Rbf ή γκαουσσιανός πυρήνας:

$$K(x, y) = e^{\frac{-\|x-y\|^2}{2\sigma^2}} \qquad \text{(Εξίσωση 3.9)}$$

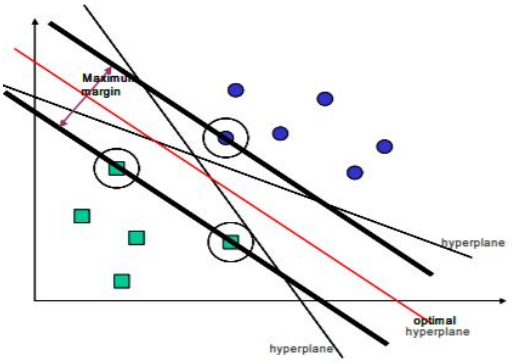
  - Σιγμοειδής πυρήνας:

$$K(x, y) = \tanh(kxy - \delta)^{p} \quad \text{(Εξίσωση 3.10)}$$

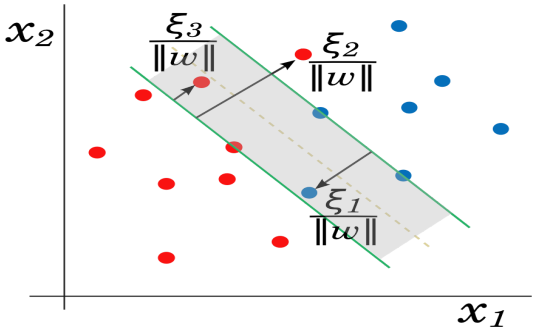
➢ Πολυωνυμικός πυρήνας:

$$K(x, y) = (x \cdot y + 1)^{p} \quad \text{(Εξίσωση 3.11)}$$

όπου x, τα διάνυσμα γνωρισμάτων του dataset μας και y μια βασική παράμετρος προς ορισμό του πυρήνα (π.χ κέντρο γκαουσσιανού πυρήνα).



Εικόνα 3.9: Η λογική διαχωρισμού δύο γραμμικά διαχωριζόμενων κλάσεων με χρήση SVM. Τα κυκλωμένα σημεία αποτελούν τα διανύσματα στήριξης.[12]

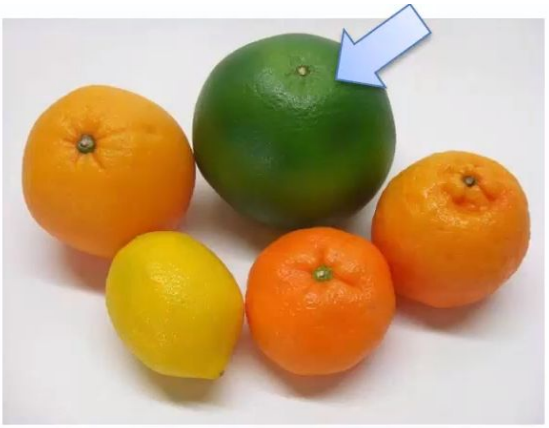


Εικόνα 3.10: Soft margin SVM.[13]

3.3.4.2 Ιδιαιτερότητες και εφαρμογές του SVM

Η λογική του αλγόριθμου διαφοροποιείται από τους κλασικούς αλγόριθμους ταξινόμησης καθώς ξεχνιέται υπο μία έννοια η λογική της μάθησης των πιο χαρακτηριστικών γνωρισμάτων κάθε κλάσης στο στάδιο της εκπαίδευσης. Αντιθέτως, γίνεται προσπάθεια να εντοπιστούν τα διανύσματα τα οποία είναι τα λιγότερο χαρακτηριστικά της κλάσης τους (διανύσματα στήριξης η support vectors) και να μεγιστοποιηθεί η απόσταση μεταξύ τους. Βλέποντας ένα παράδειγμα αναγνώρισης εικόνας με τη χρήση ταξινομητή SVM, υποθέτουμε ότι προσπαθούμε με χρήση του αλγόριθμου να διακρίνουμε μήλα από πορτοκάλια σε δοθείσες εικόνες. Ο αλγόριθμος εντοπίζει το πιο "μηλένιο" πορτοκάλι και το πιο "πορτοκαλένιο μήλο" όπως στις εικόνες 3.11, 3.12, προσπαθώντας να μεγιστοποιήσει την απόσταση μεταξύ τους έτσι ώστε να οριστεί το ζητούμενο υπερεπίπεδο που διαχωρίζει τις κλάσεις πορτοκαλιών και μήλων.

[12] *[Πηγη: [21]]*
[13] *[Πηγή: http://efavdb.com/svm-classification/]*



Εικόνα 3.11: Το πιο "μηλένιο" πορτοκάλι

Εικόνα 3.12: Το πιο "πορτοκαλένιο" μήλο

Κλασικά πεδία εφαρμογής του αλγόριθμου SVM είναι η αναγνώριση προσώπου, η ταξινόμηση εικόνων και η αναγνώριση γραφής. Επιπλέον χρησιμοποιείται ευρέως στην κατηγοριοποίηση κειμένου αλλά σε ζητήματα υπολογιστικής βιολογίας και ιατρικής, όπως η ταξινόμηση γονιδίων, η εύρεση ομόλογων πρωτεϊνών κ.α. Αξίζει να αναφέρουμε πρόσφατη έρευνα [32] από Έλληνες ερευνητές σε σχέση με το διαβήτη, στην οποία τονίζεται η προσφορά του SVM στον κλάδο.

3.3.4.3 Πλεονεκτήματα και μειονεκτήματα SVM

Τα πλεονεκτήματα του SVM είναι τα εξής:

- Τα αποτελέσματα του βασίζονται σε ένα κυρτό (convex) πρόβλημα ελαχιστοποίησης, το οποίο μάλιστα επιλύεται με αποδοτικές μεθόδους, και συνεπώς δεν υπάρχει κίνδυνος να πέσει σε τοπικά ελάχιστα.
- Αν τα δεδομένα είναι μη διαχωρίσιμα υπάρχει δυνατότητα χαλάρωσης των περιορισμών μέσω της παραμέτρου C που εξετάζεται και στην επόμενη ενότητα.
- Το kernel trick προσφέρει πολύτιμες πληροφορίες για το πρόβλημα μέσω μελέτης του πυρήνα.

Τα μειονεκτήματα του SVM είναι τα παρακάτω:

- Τα μοντέλα πυρήνων είναι αρκετά επιρρεπή σε overfitting και απαιτούν διεξοδική μελέτη.
- Ο αλγόριθμος από τη φύση του δεν εξάγει αποτελέσματα σε μορφή πιθανοτήτων. Υπάρχουν υλοποιήσεις όπως αυτή της scikit που παρουσιάζεται και στην επόμενη ενότητα οι οποίες δίνουν αυτή τη δυνατότητα, ωστόσο αυξάνει αισθητά το χρόνο υπολογισμου. Μέθοδοι όπως το Import Vector Machine [33] που βασίζεται στο Kernel Logistic Regression προτείνονται για τέτοιες ανάγκες επίλυσης, δηλαδή κυρίως σε προβλήματα multiclass classification.

3.3.4.4 Εργαλεία υλοποίησης SVM

Ο αλγόριθμος SVM είναι state-of-the-art σε επίπεδο "παραδοσιακών" μεθόδων machine learning και καλύπτεται από ένα σύνολο εργαλείων γλωσσών όπως η Java, το Matlab, η Python με τη scikit-learn. Η κλάση είναι η εξής:

class sklearn.svm.**SVC**(*C=1.0*, *kernel='rbf'*, *degree=3*, *gamma='auto_deprecated'*, *coef0=0.0*, *shrinking=True*, *probability=False*, *tol=0.001*, *cache_size=200*, *class_weight=None*, *verbose=False*, *max_iter=-1*, *decision_function_shape='ovr'*, *random_state=None*)[14]
Σημαντικό είναι να αναφέρουμε πως η παράμετρος C, χρησιμοποιείται για ποινικοποίηση των λανθασμένων ταξινομήσεων και όσο μεγαλύτερη είναι τόσο πιο "δύσκαμπτο" γίνεται το μοντέλο, ενώ αν τις δώσουμε πολύ μικρές τιμές εγκυμονεί ο κίνδυνος του overfitting.

3.3.5 Στατιστικές τεχνικές μάθησης

Στις τεχνικές αυτές χρησιμοποιούνται μοντέλα πιθανοτήτων. Το training set ως γνωστόν, αποτελείται instances ($\alpha_1,...,\alpha_n \mid v_j \in V$) δηλαδή ζεύγη διανυσμάτων γνωρισμάτων ($\alpha_1,...,\alpha_n$) και εξόδων - ετικετών αντίστοιχης κλάσης ή κλάσεων οι οποίες παίρνουν τιμές από ένα δεδομένο διακριτό σύνολο κλάσεων $V = \{v_1,...,v_k\}$. Τα instances αυτά είναι ξεχωριστά "στιγμιότυπα" ή τιμές των γνωρισμάτων ($A_1,...,A_n$) όπου A_i τα ονόματα των γνωρισμάτων (π.χ A_1 = ηλικία, A_2 = εισόδημα, α_1=25 ετών, α_2=20.000 $) μαζί με τις αντίστοιχες κλάσεις τους v_i ως έξοδο (π.χ αγόρασε το προϊόν η όχι). Οι στατιστικοί ταξινομητές επιτυγχάνουν, επεξεργαζόμενοι κάθε instance, να χτίσουν ένα πιθανοτικό μοντέλο το οποίο εξάγει μια πιθανότητα ενός νέου στοιχείου να ανήκει σε κάποια από τις δοθείσες κλάσεις. Συνήθως επιλέγεται η κλάση με τη μεγαλύτερη πιθανότητα. Πιο αναλυτικές πληροφορίες μπορούν να αναζητηθούν στα [21],[34].

3.3.5.1 Ταξινομητής Naive-Bayes

3.3.5.1.1 Περιγραφή αλγόριθμου

Ο ταξινομητής Naive - Bayes, ο οποίος θεμελιώθηκε στο [35], αποτελεί την πιο απλή μορφή στατιστικού ταξινομητή. Ανατρέχοντας σε όλα τα instances ($\alpha_1,...,\alpha_n \mid v_i$), μαθαίνει την πιθανότητα $P(\alpha_i \mid v_j)$ κάθε γνώρισμα A_i να παίρνει τιμές δεδομένης κάποιας ετικέτας κλάσης v_i. Στη συνέχεια χρησιμοποιεί το γνωστό νόμο του Bayes για να υπολογίσει την δεσμευμένη πιθανότητα ένα νέο διάνυσμα γνωρισμάτων να ανήκει σε κάθε μία από γνωστές κλάσεις και επιλέγεται συνήθως αυτή με τη μεγαλύτερη πιθανότητα. Ο υπολογισμός γίνεται ως εξής:

$$V_{nb} = argmax_{v_j \in V} P(v_j) \prod_i P(a_i \mid v_j) \qquad \text{(Εξίσωση 3.12)}$$

Η εκμάθηση των πιθανοτήτων $P(\alpha_i \mid v_j)$ γίνεται ανάλογα με τον τρόπο που επιλέγει ο αναλυτής. Οι πιο συνήθης υλοποιήσεις είναι η Gaussian Naive Bayes και η Multinomial Naive Bayes.

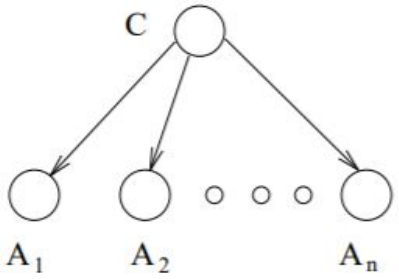
Στη διαδικασία αυτή, γίνεται η βασική και, κατα κανόνα, ψευδής παραδοχή ότι τα γνωρίσματα είναι εντελώς ανεξάρτητα μεταξύ τους, εξού και η ονοματοδοσία Naive δηλαδή αφελής. Μια πολύ απλή περίπτωση προβλήματος που καταδεικνύει το πρόβλημα αυτό είναι δύο γνωρίσματα να είναι η ηλικία και το εισόδημα των ατόμων όπως έχουμε ξαναδεί. Ο αλγόριθμος Naive-Bayes θεωρεί αυτα τα δύο γνωρίσματα στατιστικά ανεξάρτητα πράγμα το οποίο προφανώς απέχει πολύ απ' την πραγματικότητα. Ωστόσο, τα αποτελέσματα του αλγόριθμου συχνά εκπλήσσουν, ενώ ανταγωνίζεται σε ορισμένες κατηγορίες προβλημάτων, γνωστούς state-of-the-art αλγόριθμους οπως το SVM.
Ο αλγόριθμος Naive - Bayes:

- Διακρίνεται για την ταχύτητα του στο επίπεδο της μάθησης - training του μοντέλου.
- Ενδείκνυται όταν τα δεδομένα εκπαίδευσης είναι λίγα καθώς συγκλίνει σχετικά γρήγορα.

[14] *[Πηγή: https://github.com/scikit-learn/scikit-learn/blob/bac89c2/sklearn/svm/classes.py#L429]*

- ❖ Ενδείκνυται κυρίως για προβλήματα κατηγοριοποίησης κειμένου.
- ❖ Είναι ο πλέον κατάλληλος όταν ισχύει σε μεγάλο βαθμό η συνθήκη της ανεξαρτησίας.

Το μοντέλο Naive - Bayes, όπως φαίνεται στην εικόνα 3.13 αποτελεί μία υποπερίπτωση των Bayesian δικτύων που θα εξεταστούν στην επόμενη ενότητα.



Εικόνα 3.13: Η αναπαράσταση του μοντέλου Naive Bayes ως Bayesian δίκτυο.

3.3.5.1.2 Κώδικας

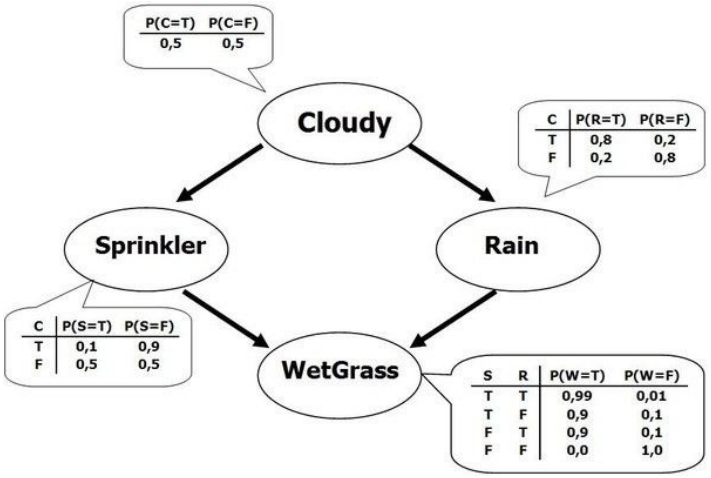
Η κλάση της βιβλιοθήκης scikit-learn που υλοποιεί τον αλγόριθμο Gaussian Naive Bayes σε python:

*class sklearn.*naive_bayes.***GaussianNB****(priors=None, var_smoothing=1e-09)*

3.3.5.2 Bayesian δίκτυα

3.3.5.2.1 Περιγραφή των Bayesian δικτύων

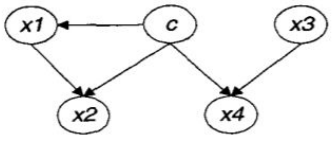
Ο αλγόριθμος Naive-Bayes είναι αρκετά αποτελεσματικός σε πολλές περιπτώσεις ωστόσο πάντοτε υπήρξε το ερωτηματικό εάν μπορεί να βελτιωθεί περαιτέρω η απόδοση του, εφόσον λειτουργεί με τη μη ρεαλιστική υπόθεση της ανεξαρτησίας.. Το πρόβλημα της συνθήκης ανεξαρτησίας λύνουν τα Bayesian δίκτυα τα οποία αποτελούν ακυκλικούς κατευθυνόμενους γράφους όπου κάθε κόμβος αναπαριστά μια τυχαία μεταβλητή - γνώρισμα η κλάση. Τα βέλη αναπαριστούν την εξάρτηση ή αλλιώς τη σχέση αιτίου-αιτιατού. Οι κόμβοι ικανοποιούν τη Μαρκοβιανή ιδιότητα, δηλαδή είναι ανεξάρτητοι των "προγόνων" δεδομένων των "γονέων" τους. Σε αντίθεση με το μοντέλο Naive - Bayes εδώ επιτρέπεται η σύνδεση μεταξύ των γνωρισμάτων A_i (βλ. εικόνα 3.13). Συναρτήσεις (scoring functions) όπως η MDL scoring function χρησιμοποιούνται ώστε να βρούμε την κατάλληλη δομή αλλα και τις παραμέτρους του Bayesian δικτύου έτσι ώστε να ικανοποιεί με τον καλύτερο τρόπο το δοθέν dataset. Ενα χαρακτηριστικό παράδειγμα Bayesian δικτύου είναι αυτό της εικόνας 3.14. Παρατηρούμε ότι με τη μαρκοβιανή ιδιότητα γίνεται η παραδοχή ότι αν παρατηρηθεί βροχή ή ενεργοποίηση του ποτίσματος τότε το βρεγμένο γρασίδι δεν εξαρτάται από την ύπαρξη ή όχι συννεφιάς.



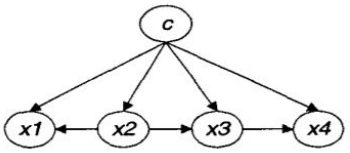
Εικόνα 3.14: Ένα χαρακτηριστικό Bayesian δίκτυο.[15]

Συνήθης υποκατηγορίες των Bayesian δικτύων που χρησιμοποιούνται ως ταξινομητές είναι, όπως αναλύει και το[34] οι εξής:

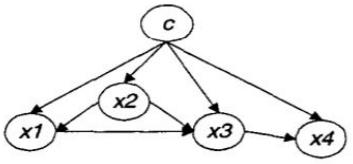
- General Bayesian Network (εικόνα 3.15)
- Tree augmented Naive Bayes - TAN (βλ. εικόνα 3.16)
- BN augmented Naive Bayes - BAN (εικόνα 3.17)



Εικόνα 3.15: General Bayesian Network



Εικόνα 3.16: Tree augmented Naive Bayes



Εικόνα 3.17: BN augmented Naive Bayes

3.3.5.2.2 Χρησιμότητα των Bayesian δικτύων

Η κύρια διαφορά των Bayesian δικτύων με τους υπόλοιπους αλγόριθμους classification είναι ότι προσφέρουν τη δυνατότητα να ανακαλύπτουμε σε ένα dataset τις από κοινού κατανομές $P(A_i , V_j)$ και όχι μόνο τις δεσμευμένες κατανομές $P(A_i \mid V_j)$. Πρόκειται για ένα πολύ πιο δύσκολο και χρονοβόρο υπολογισμό ο οποίος όμως προσφέρει τα εξής πλεονεκτήματα:

- ❖ Δίνει μια πολύ πιο συνολική εικόνα των εξαρτήσεων για το σύνολο των δεδομένων μας
- ❖ Φανερώνει σχέσεις αιτίου και αιτιατού στα δεδομένα
- ❖ Μειώνει την εξάρτηση από ελλείψεις δεδομένων (missing data) φαινόμενο το οποίο είναι για παράδειγμα πολύ σύνηθες σε ιατρικά δεδομένα.
- ❖ Ο υπολογισμός των από κοινού κατανομών γίνεται αισθητά ευκολότερος σε διακριτές μεταβλητές, όπου και η χρήση τους είναι πιο συνήθης.

[15] *[Πηγή:https://towardsdatascience.com/introduction-to-bayesian-networks-81031eeed94e]*

3.3.5.2.3 Εργαλεία

Η scikit δεν παρέχει κάποια κλάση για γενικευμένα Bayesian δίκτυα καθώς πρόκειται για ένα πολύ πιο εξειδικευμένο και πολύπλοκο αντικείμενο. Ωστόσο μπορεί κανείς να βρει υλοποιήσεις αυτων στο pomegranate[16], μια γνωστή βιβλιοθήκη του git με πιθανοτικά μοντέλα.

3.4 Μετρικές αξιολόγησης μοντέλων ταξινόμησης

Η επιλογή αλγόριθμου ταξινόμησης και στη συνέχεια η ρύθμιση των εξωτερικών παραμέτρων του επιλεχθέντος αλγόριθμο γίνονται με τη χρήση τεχνικών αξιολόγησης των μοντέλων πάνω στα validation sets όπως αυτή του cross-validation. Η τεχνικές αυτές, ωστόσο, πραγματοποιούν την αξιολόγηση χρησιμοποιώντας μια σειρά από μετρικές. Οι μετρικές αυτές είναι πολύ σημαντικό να χρησιμοποιηθούν και να ερμηνευθούν κατάλληλα από τον αναλυτή ο οποίος οφείλει και να γνωρίζει σε ποιες από αυτές πρέπει να δώσει βάση, δεδομένης της φύσης του προβλήματος που καλείται να λύσει.

3.4.1 Πίνακας σύγχυσης

Πριν αναφέρουμε οποιαδήποτε μετρική ταξινόμησης, είναι σημαντικό να οριστεί η έννοια της μήτρας σύγχυσης ή confusion matrix, από την ανάλυση της οποίας προκύπτουν εμμέσως, οι περισσότερες μετρικές αξιολόγησης ταξινομητών. Στον πίνακα 3.1 βλέπουμε ένα παράδειγμα confusion matrix στο περιβάλλον Spyder της python για δυαδική ταξινόμηση:

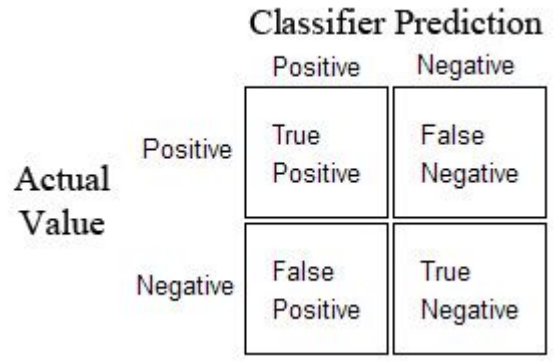
Index	Predicted class: 1	Predicted class: 0
Actual class: 1	64	4
Actual class: 0	4	28

Πίνακας 3.1: Παράδειγμα πίνακα σύγχυσης

Οι τιμές του confusion matrix 3.1 δίνουν τα αποτελέσματα της χρήσης ενός ταξινομητή για πρόβλεψη γνωστών δεδομένων και αντιπροσωπεύουν τα πλήθη των εξής: (όπως μπορεί να διακρίνει κανείς και στην εικόνα 3.18)

- True Positives (TP) = Δείγματα όπου ο ταξινομητής προέβλεψε ότι ανήκουν στην κλάση 1(Positive) και μάντεψε σωστά (TRUE).
- False Positives (FP) = Δείγματα όπου ο ταξινομητής προέβλεψε ότι ανήκουν στην κλάση 1(Positive) και μάντεψε λάθος (FALSE).
- False Negatives (FN) = Δείγματα όπου ο ταξινομητής προέβλεψε ότι ανήκουν στην κλάση 0 (Negative) και μάντεψε λάθος (FALSE).
- True Negatives (TN) = Δείγματα όπου ο ταξινομητής προέβλεψε ότι ανήκουν στην κλάση 0 (Negative) και μάντεψε σωστά (TRUE).

[16] *https://github.com/jmschrei/pomegranate*



Εικόνα 3.18: Η ερμηνεία των κελιών ενός confusion matrix.[17]

3.4.2 Ευστοχία

Η μετρική της ευστοχίας ή accuracy είναι η απλούστερη μετρική αξιολόγησης ενός μοντέλου ταξινόμησης. Μετά την εφαρμογή το μοντέλου στο εκάστοτε validation set, η μετρική της ευστοχίας για ταξινόμηση P θετικών δειγμάτων ετικέτας 1 και N αρνητικών δειγμάτων ετικέτας 0, δεδομένου και του confusion matrix, προκύπτει ως εξής:

$$acc = \frac{TP + TN}{N + P} \qquad \text{(Εξίσωση 3.13)}$$

Η μετρική αυτή είναι αρκετά απλοική και μπορεί να μας δώσει καλή εικόνα για το μοντέλο μόνο στην περίπτωση κλάσεων με ισάριθμα στοιχεία. Σε αντίθετη περίπτωση, αν φανταστούμε ένα σύνολο 1000 δειγμάτων δύο κλάσεων A,B και υποθέσουμε πως τα 980 δείγματα ανήκουν στην κλάση A και τα υπόλοιπα στην κλάση B τότε το απλούστατο μοντέλο το οποίο υποθέτει ότι όλα τα δείγματα ανήκουν στην κλάση A πετυχαίνει ποσοστό ευστοχίας 98% κάτι το οποίο φαινομενικά είναι κάλο, ωστόσο δε θα έλεγε κανείς το ίδιο αν για παράδειγμα τα δείγματα ήταν τραπεζικές συναλλαγές, η κλάση B τραπεζικές συναλλαγές απάτης, η κλάση A νόμιμες συναλλαγές και το μοντέλο δεχόταν ως είσοδο τα δεδομένα συναλλαγων με σκοπό να ξεχωρίσει τις νόμιμες συναλλαγές από τις απάτες.

3.4.3 Precision, Recall, F-measure, Specificity

Την αδυναμία της μετρικής ευστοχίας έρχονται να καλύψουν οι παρακάτω πιο εξειδικευμένες μετρικές, οι οποίες μας δίνουν μια πολύ πιο πλήρη και αντικειμενική εικόνα για την απόδοση του ταξινομητή που χρησιμοποιούμε:

1.

$$Precision = \frac{TP}{TP + FP} = \frac{correctly\ predicted\ positives}{total\ number\ of\ \ predicted\ positives} \qquad \text{(Εξίσωση 3.14)}$$

Δεδομένου ότι το μοντέλο σημαίνει συναγερμό απάτης, ποια είναι η πιθανότητα να υπάρχει πράγματι απάτη;

[17] *[Πηγή: https://www.researchgate.net/figure/Confusion-matrix-example_fig1_256418526]*

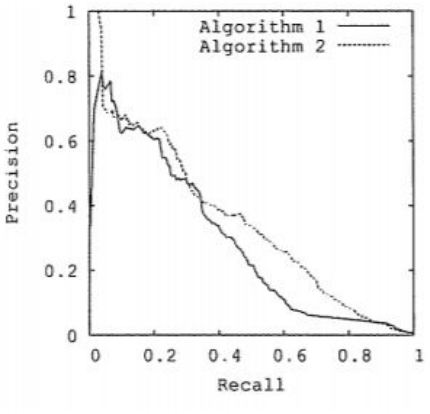
2.

$$Recall = Sensitivity = True\ Positive\ Rate = TPR = \frac{TP}{TP + FN} = \frac{TP}{N} = \frac{correctly\ predicted\ positives}{total\ number\ of\ \ actual\ positives}$$

(Εξίσωση 3.15)

Δεδομένου ότι υπάρχει απάτη με τη πιθανότητα θα επιτύχει το μοντέλο να την εντοπίσει; (επιθυμούμε μεγιστοποίηση αυτής της μετρικής στο συγκεκριμένο πρόβλημα)

Το διάγραμμα precision recall μπορεί να μας δώσει πληροφορίες για σύγκριση αλγόριθμων ταξινόμησης όπως τονίζεται στην εικόνα 3.19 και ορισμένες φορές πιο αποτελεσματικά απ' ότι το ROC curve που θα δούμε στην επόμενη ενότητα 3.4.4 και πιο συγκεκριμένα στην εικόνα 3.19 όπου συγκρίνονται οι ίδιοι αλγόριθμοι στο ίδιο πρόβλημα ταξινόμησης. Στην εικόνα 3.19 γίνεται εμφανής η υπεροχή του αλγόριθμου 2 έναντι του 1 στο χώρο precision, recall όπως παρουσιάζεται και στο [36].



Εικόνα 3.19: Παράδειγμα διαγράμματος precision-recall 2 αλγόριθμων ταξινόμησης

3.

$$F - measure = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$$

(Εξίσωση 3.16)

Η μετρική F-measure αποτελεί τον αρμονικό μέσο των precision-recall και ιδανικό για τον αναλυτή είναι να έχει τη μεγαλύτερη δυνατή τιμή στο διάστημα [0,1]. Χρησιμοποιείται ευρέως σε προβλήματα Natural Language Processing.

4.

$$Specif icity = True\ Negative\ Rate = TNR = \frac{TN}{TN + FP} = \frac{TN}{N} = \frac{correctly\ predicted\ negatives}{total\ number\ of\ \ actual\ negatives}$$

(Εξίσωση 3.17)

Δεδομένου ότι μια συναλλαγή είναι νόμιμη, ποιά η πιθανότητα το μοντέλο να το εντοπίσει και να μη σημάνει συναγερμό χωρίς λόγο;

5.

$$False\ Positive\ Rate = \frac{FP}{FP + TN} = 1 - Specif icity = \frac{incorrectly\ predicted\ positives}{total\ number\ of\ \ actual\ negatives}$$

(Εξίσωση 3.18)

Δεδομένου ότι μια συναλλαγή είναι νόμιμη, ποιά η πιθανότητα ο ταξινομητής να τη θεωρήσει ως απάτη και να σημάνει συναγερμό; Παρατηρούμε πως η μετρική αυτή παίρνει τις συμπληρωματικές του specificity, ωστόσο παρατίθεται λόγω της χρησιμότητας του στην επόμενη ενότητα που αφορά την καμπύλη ROC.

Οι παραπάνω μετρικές μας δίνουν μια σαφέστερη του μοντέλου απ΄ ότι αυτή της ευστοχίας και επιτρέπουν στον αναλυτή να δώσει διαφορετική βαρύτητα στις προβλέψεις του μοντέλου ανάλογα με τις ανάγκες του προβλήματος, όπως είδαμε και παραπάνω.

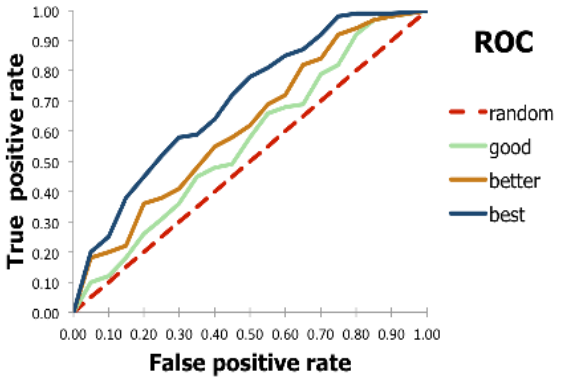
3.4.4 Καμπύλη ROC και εμβαδόν AUC

Για το ενδεχόμενο στο οποίο θέλουμε να κάνουμε μια γενική αξιολόγηση του μοντέλου χωρίς να δίνεται διαφορετική βαρύτητα στα πλήθη των False Positives ή False Negatives, όπως στην περίπτωση της τραπεζικής απάτης υπάρχει η μετρική AUC (Area Under Curve) [37] η οποία αποτελεί το εμβαδόν της γνωστής καμπύλης ROC (Receiver Operating Characteristic) με τον οριζόντιο άξονα.

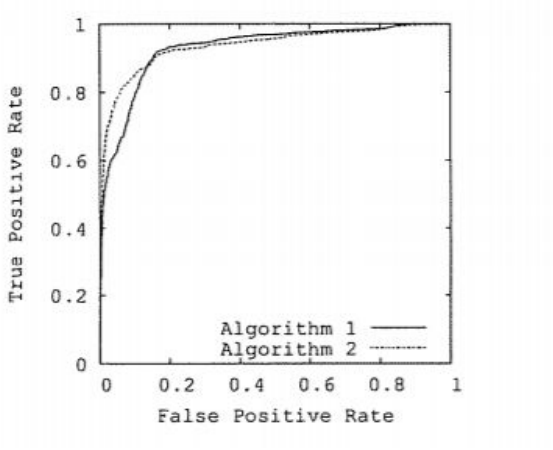
Η καμπύλη ROC κατασκευάζεται από τα ζεύγη TPR, FPR $\in$ [0,1] για όλα τα πιθανα κατώφλια πιθανότητας με την έννοια που εξετάστηκαν στην ενότητα 3.3.1 του Logistic Regression. Γίνεται λοιπόν εμφανές ότι είναι αναγκαία η χρήση ενός πιθανοτικού μοντέλου ταξινόμησης. Στην εικόνα 3.20 η διαγώνια γραμμή αναπαριστά τις επιδόσεις ενός μοντέλου το οποίο κάνει τυχαίες προβλέψεις. Το σημείο (0,0) αναπαριστά τις τιμές των δύο μετρικών για τιμή κατωφλίου 1, δηλαδή το μοντέλο προβλέπει πάντοτε 0 (Negative). Το σημείο (1,1) αναπαριστα τις τιμές των δύο μετρικών για τιμή κατωφλίου 0, δηλαδή όπου το μοντέλο προβλέπει για οποιοδήποτε δείγμα ετικέτα 1 (Positive)

Προφανώς, η χειρότερη δυνατή τιμή για το AUC είναι το 0,5 όπου η ROC ταυτίζεται με τη διαγώνιο και η ιδανική είναι το 1. Ωστόσο είναι σημαντικό να μη βασιζόμαστε αποκλειστικά στη μετρική AUC, καθώς μας δίνει συσσωρευτικές πληροφορίες για όλες τις τιμές κατωφλίου και όχι για τις συγκεκριμένες που μας ενδιαφέρουν. Αντιθέτως, η καμπύλη ROC δίνει περισσότερες πληροφορίες για τα επιμέρους διαστήματα κατωφλίων στα οποία δύο αλγόριθμοι ενδέχεται να έχουν διαφορετικές επιδόσεις. Για παράδειγμα, στην εικόνα 3.21 βλέπουμε τα ROC-curves των αλγόριθμων που παρουσιάστηκαν και στην εικόνα 3.19. Η υπεροχή του αλγόριθμου 2 είναι δεδομένη, ωστόσο η καμπύλη ROC δεν ενδείκνυται για να κάνει αυτού του είδους τη διάκριση.

Σε περίπτωση παραπάνω από δύο κλάσεων γίνεται αξιολόγηση των καμπυλών ROC όλων των κλάσεων ανα ζεύγη όπως παρουσιάζεται στο [38]. Περισσότερες λεπτομέρειες για την ROC μπορεί να αναζητήσει κανείς στο [39].



Εικόνα 3.20: Σύγκριση διάφορων καμπυλών ROC.[18]



Εικόνα 3.21: ROC-curve των αλγόριθμων της εικόνας 3.19.[19]

3.4.5 Λοιπές μετρικές αξιολόγησης μοντέλων ταξινόμησης

Ανατρέχοντας κανείς στο [40] θα συναντήσει πολλές ακόμη μεθόδους αξιολόγησης μοντέλων ταξινόμησης. Σημαντικό είναι να αναφέρουμε τη μετρική του cross-entropy (multiclass classification) που ανάγεται σε log loss για binary classification οι οποίες χρησιμοποιούνται ως μέτρα της αβεβαιότητας ενός πιθανοτικού μοντέλου στην πρόβλεψη κλάσης. Ενδεικτικά για κάθε πρόβλεψη ετικέτας δείγματος οι παραπάνω μετρικές υπολογίζονται ως εξης:

$$Log\ loss = -(y\log(\hat{y}) + (1-y)\log(1-\hat{y})),\ y=\{0,1\},\ \hat{y}\in[0,1] \qquad \text{(Εξίσωση 3.19)}$$

$$H(y,\hat{y}) = -\sum_i y_i \log\hat{y}_i,\quad y=\{0,1\},\ \hat{y}\in[0,1] \qquad \text{(Εξίσωση 3.20)}$$

Οι μετρικές αυτές αθροίζονται για όλες τις προβλέψεις του μοντέλου και έτσι κατασκευάζεται η τελική συνάρτηση σφάλματος ή loss function προς ελαχιστοποίηση. Επιπλέον, χρησιμοποιούνται ευρέως και στα νευρωνικά δίκτυα ταξινόμησης τα οποία εξετάζονται σε μετέπειτα κεφάλαια.

Χαρακτηριστική είναι και η hinge loss function που χρησιμοποιεί ο αλγόριθμος SVM:

$$\ell(y) = \max(0, 1 - y\cdot\hat{y}) \qquad \text{(Εξίσωση 3.20)}$$

Αντίστοιχα με την παλινδρόμηση μπορούν να χρησιμοποιηθούν και συναρτήσεις σφάλματος όπως τα mse, mae, ωστόσο δεν "ποινικοποιούν" με βέλτιστο τρόπο τις πιθανοτικές προβλέψεις και συνεπώς δεν απολαμβάνουν ευρείας χρήσης σε προβλήματα ταξινόμησης.

[18] *[Πηγή:https://docs.eyesopen.com/toolkits/cookbook/python/plotting/roc.html]*
[19] *[Πηγη:[36]]*

4. Παλινδρόμηση

4.1 Μια απαραίτητη αναδρομή σε γνώσεις στατιστικής

Έστω ένας κανονικά κατανεμημένος πληθυσμός ο οποίος έχει μέση τιμή μ και τυπική απόκλιση σ, και έχουμε ένα δείγμα n ατόμων από τον πληθυσμό. Τότε όλα τα άτομα X_i, ακολουθούν αντίστοιχη κατανομή:

$$X_1, \ldots, X_n \sim N(\mu, \sigma^2) \qquad \text{(Εξίσωση 4.1)}$$

Τότε, η μέση τιμή του δείγματος

$$\overline{X} = \frac{X_1 + \ldots + X_n}{n} \qquad \text{(Εξίσωση 4.2)}$$

έχει την ακόλουθη κατανομή:

$$\overline{X} \sim N(\mu, \sigma^2/n) \qquad \text{(Εξίσωση 4.3)}$$

Τα **στατιστικά σφάλματα** είναι τα εξής:

$$e_i = X_i - \mu \qquad \text{(Εξίσωση 4.4)}$$

Τα **residuals** ορίζονται ως εξής:

$$r_i = X_i - \overline{X} \qquad \text{(Εξίσωση 4.5)}$$

4.2 Γενικα για την παλινδρόμηση

Η παλινδρόμηση (regression) αποτελεί ένα σύνολο από μεθόδους για την εκτίμηση της εξάρτησης μεταξύ μιας βαθμωτής ή διανυσματικής ανεξάρτητης μεταβλητής **X** και μιας βαθμωτής εξαρτημένης μεταβλητής y. Στόχος, λοιπόν, είναι να προσδιοριστεί η μέση απόκριση που έχει η εξαρτημένη μεταβλητή στις επιμέρους μεταβολές της ανεξάρτητης μεταβλητης.
Διαισθητικά ένα μοντέλο παλινδρόμησης αποτελεί σύμφωνα με το [41] μέσο έκφρασης των εξής :

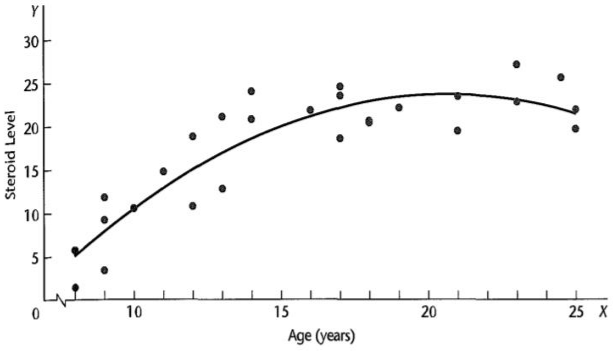
1. Μιας τάσης της εξαρτημένης μεταβλητής y να μεταβάλλεται με συστηματικό τρόπο σε σχέση με τις μεταβολές της ανεξάρτητης μεταβλητής.
2. Μιας διασποράς σημείων γύρω από την καμπύλη της στατιστικής σχέσης που συνδέει τις **X**, y.

Για παράδειγμα, το γράφημα της εικόνας 4.1 απεικονίζει μια καμπύλη παλινδρόμησης για την εκτίμηση της ποσότητας στεροειδών σε παιδιά και νέους άντρες ανάλογα με την ηλικία τους. Η καμπύλη παλινδρόμησης είναι μια συστηματική μεταβολή του Y ως προς το X ωστόσο οι πραγματικές παρατηρήσεις είναι διεσπαρμένες γύρω από την καμπύλη παλινδρόμησης.

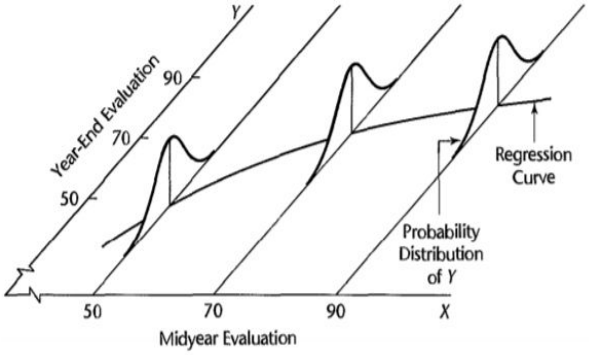
Τα παραπάνω ενσωματώνονται μεσα στο μοντέλο εισάγοντας τα παρακάτω δεδομένα:

1. Για κάθε τιμή του **X** το y ακολουθει μια κανονική κατανομή της οποίας το μοντέλο προβλέπει τη μέση τιμή.
2. Η μέσες τιμές των κανονικών αυτών κατανομών μεταβάλλονται με κάποιο συστηματικό τρόπο σε σχέση με τις μεταβολες της **X.**

Για παράδειγμα, το γράφημα της εικόνας 4.2 απεικονίζει μια καμπύλη παλινδρόμησης για την εκτίμηση της απόδοσης των υπαλλήλων σε αξιολόγηση στο τέλος της χρονιάς σε σχέση με την αξιολόγηση που απέσπασαν στο μέσο της χρονιάς. Η καμπύλη παλινδρόμησης αποτελεί το μοντέλο το οποίο προβλέπει τη μέση τιμή της εξαρτημένης μεταβλητής y. Ωστόσο οι πραγματικές τις τιμές ακολουθούν κανονική κατανομή γύρω από την αντίστοιχη προβλεπόμενη τιμή.



Εικόνα 4.1: Καμπύλη παλινδρόμησης για την εκτίμηση της ποσότητας στεροειδών σε παιδιά και νέους άντρες ανάλογα με την ηλικία τους.[20]



Εικόνα 4.2: Καμπύλη παλινδρόμησης για την εκτίμηση της απόδοσης των υπαλλήλων σε αξιολόγηση στο τέλος της χρονιάς σε σχέση με την αξιολόγηση που απέσπασαν στο μέσο της χρονιάς.[12]

4.3 Παλινδρόμηση στη μηχανική μάθηση

Η παλινδρόμηση αποτελεί, ως γνωστό, μία εργασία επιτηρούμενης μάθησης. Επανερχόμαστε στο dataset A της ενότητας 2.2.1 της επιτηρούμενης μάθησης και στις εξισώσεις που εμφανίστηκαν εκεί, θεωρώντας συνεχές σύνολο τιμών εξόδου, όπως επιτάσσει ο χαρακτήρας της παλινδρόμησης. Συνεπώς, οι εκτιμήσεις που προκύπτουν από το μοντέλο h συνδέονται με τις πραγματικές παρατηρήσεις με τον εξής τρόπο όπως καταδεικνύεται και στο [41]:

$$\widehat{y_i} = h(x_1, \ldots x_k \mid y_i) + r_i = y_i + r_i \qquad \text{(Εξίσωση 4.6)}$$

όπου:
r_i: τα residuals όπως ορίζονται στην εξίσωση 1.5.
$\widehat{y_i}$: Η εκτίμηση του y_i η οποία ουσιαστικά αποτελεί μια εκτίμηση της μέσης τιμής της κανονικής κατανομής την οποία ακολουθεί.

Ο τρόπος εξαγωγής της εκτιμήτριας συνάρτησης εξαρτάται από το είδος του μοντέλου που χρησιμοποιούμε. Επιπλέον, η διαδικασία όπως είδαμε βασίζεται στην ελαχιστοποίηση μιας συνάρτησης σφάλματος, όπως είδαμε στην εξίσωση 2.3, η οποία είναι της μορφής:

$$E(h) = \sum_x error(h(x), f(x)) \ , \ x \in A_1 \qquad \text{(Εξίσωση 4.7)}$$

[20] *[Πηγή :[41]]*

Στο σημείο αυτό είναι χρήσιμο να οριστεί και το άθροισμα τετραγώνων των residuals καθώς, κατα κανόνα, αποτελεί τη συνάρτηση ελαχιστοποίησης ή μέρος αυτής κατα την εκπαίδευση αλγόριθμων παλινδρόμησης:

$$RSS = Residual\ Sum\ of\ Squares = \sum_i r_i^2 = \sum_i \left(y_i - \hat{y}_i \right)^2 \quad \text{(Εξίσωση 4.8)}$$

4.4 Μετρικές αξιολόγησης μοντέλων παλινδρόμησης

Οι πιο συνήθεις μετρικές αξιολόγησης των μοντέλων παλινδρόμησης είναι το μέσο τετραγωνικό σφάλμα (MSE), το μέσο απόλυτο σφάλμα (MAE), το R^2, και το explained variance, κάποια από τα οποία αναλύονται σε βάθος σε επόμενες ενότητες. Τα σφάλματα αυτά εμπεριέχονται στην κλάση metrics της βιβλιοθήκης scikit-learn της python.

$$MSE = \frac{\sum_{i=1}^{N} \left(y_i - \hat{y}_i \right)^2}{N} \quad \text{(Εξίσωση 4.9)}$$

$$MAE = \frac{\sum_{i=1}^{N} \left(y_i - \hat{y}_i \right)}{N} \quad \text{(Εξίσωση 4.10)}$$

4.5 Μοντέλα και αλγόριθμοι παλινδρόμησης

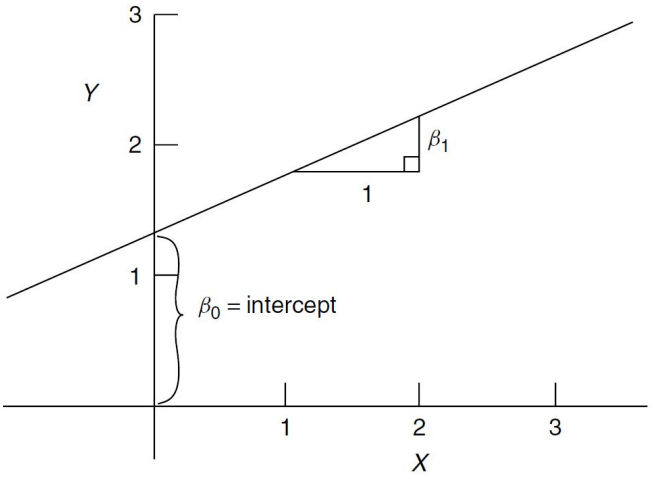
Στη βιβλιογραφία εξετάζονται πολλα διαφορετικά μοντέλα παλινδρόμησης όπως η απλή γραμμική παλινδρόμηση, η πολλαπλή γραμμική παλινδρόμηση, η πολυωνυμική παλινδρόμηση, η παλινδρόμηση με διάνυσμα στήριξης (SVR: support vector regression), παλινδρόμηση δέντρων αποφάσεων, η παλινδρόμηση random forest και η λογιστική παλινδρόμηση η οποία έχει εφαρμογή κυρίως σε προβλήματα ταξινόμησης. Στη συνέχεια θα εμβαθύνουμε στα μοντέλα παλινδρόμησης.

4.5.1 Απλή γραμμική παλινδρόμηση

Η γραμμική παλινδρόμηση αποτελεί την απλούστερη μορφή παλινδρόμησης. Ουσιαστικά γίνεται προσπάθεια να προσεγγίσουμε τη συνάρτηση στόχο f μέσω μιας γραμμικής συνάρτησης h:

$$y = h(X) = \beta_1 X + \beta o \quad \text{(Εξίσωση 4.11)}$$

Η αναζήτηση της συνάρτησης h ανάγεται στον προσδιορισμό των παραμέτρων της $\boldsymbol{\beta_0}$ (κλίση - slope) και $\boldsymbol{\beta_1}$ (τεταγμένη τομής με τον κατακόρυφο άξονα - intercept) μεγέθη τα οποία διευκρινίζονται στο γράφημα της εικόνας 4.3. Τα μεγέθη αυτά χαρακτηρίζουν τη σχέση της εξαρτημένης μεταβλητής με καθεμία απο τις μεταβλητές πρόβλεψης ξεχωριστά ενώ οι τιμές που παίρνουν είναι ανεξάρτητες για κάθε μεταβλητή πρόβλεψης. Στην απλή γραμμική παλινδρόμηση τα β_1,β_0 είναι βαθμωτα μεγέθη ενώ το **X** αποτελεί την ανεξάρτητη μεταβλητή.



Εικόνα 4.3: Slope και intercept της γραμμικής παλινδρόμησης. [21]

4.5.1.1 Παραδοχές της απλής γραμμικής παλινδρόμησης

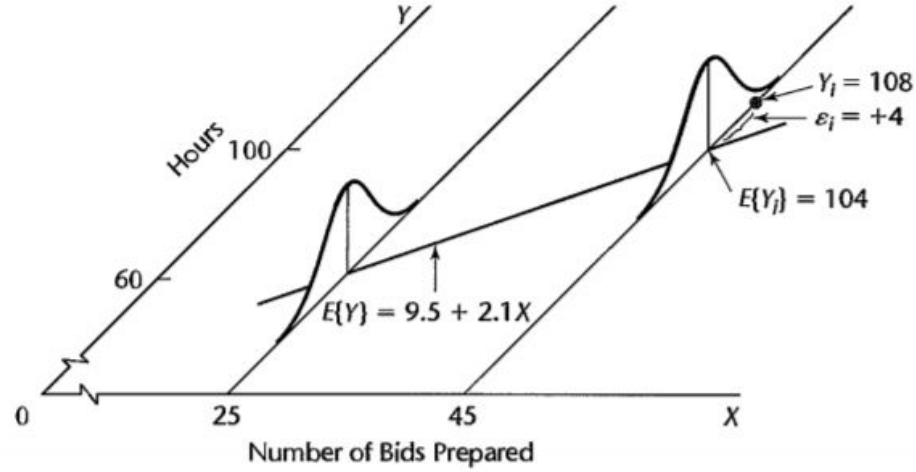
Η γραμμική παλινδρόμηση αποτελεί χρήσιμο εργαλείο της μηχανικής μάθησης, ωστόσο είναι συγκεκριμένες οι προϋποθέσεις τις οποίες πρέπει να πληρούν τα δεδομένα του προβλήματος (dataset) ώστε να έχουμε έγκυρα αποτελέσματα. Οι προϋποθέσεις αυτές είναι οι εξής:

- Γραμμική εξάρτηση εξαρτημένης - ανεξάρτητης μεταβλητής. Είναι επίσης πολύ σημαντικό να προσέξουμε τα απομακρυσμένα σημεία (outliers) τα οποία δηλαδη καταστρατηγούν κατα πολύ τη γραμμική εξάρτηση διότι επηρεάζουν κατα πολύ τα αποτελέσματα της γραμμικής παλινδρόμησης. (βλ. εικόνα 4.7)
- Σταθερή διακύμανση (Homoscedasticity): Αυτό σημαίνει πως το σφάλμα της εξαρτημένης μεταβλητής παρουσιάζει σταθερή διακύμανση ως προς την αναμενόμενη τιμή για όλες τις τιμές της εξαρτημένης μεταβλητής. Εναλλακτικά σημαίνει όλα πως τα σφάλματα πρόβλεψης - residuals έχουν σταθερή διακύμανση.
- Οι μεταβλητές του προβλήματος να ακολουθούν κανονικές κατανομές όπως φαίνεται στην εικόνα 4.4. Εξετάζεται με ιστόγραμμα η Q-Q γράφημα. Αν δεν ικανοποιείται μπορούμε να εφαρμόσουμε ένα μη γραμμικό μετασχηματισμό (π.χ λογαριθμικό)
- Απουσία αυτοσυσχέτισης μεταξύ των δεδομένων. Αυτοσυσχέτιση υπάρχει όταν τα residuals εξαρτώνται από προηγούμενες τιμές τους δηλαδή y(x+1) εξαρτάται από y(x) (π.χ χρονοσειρές όπως οι τιμές μετοχων, ή η θερμοκρασίες μιας πόλης κ.α)

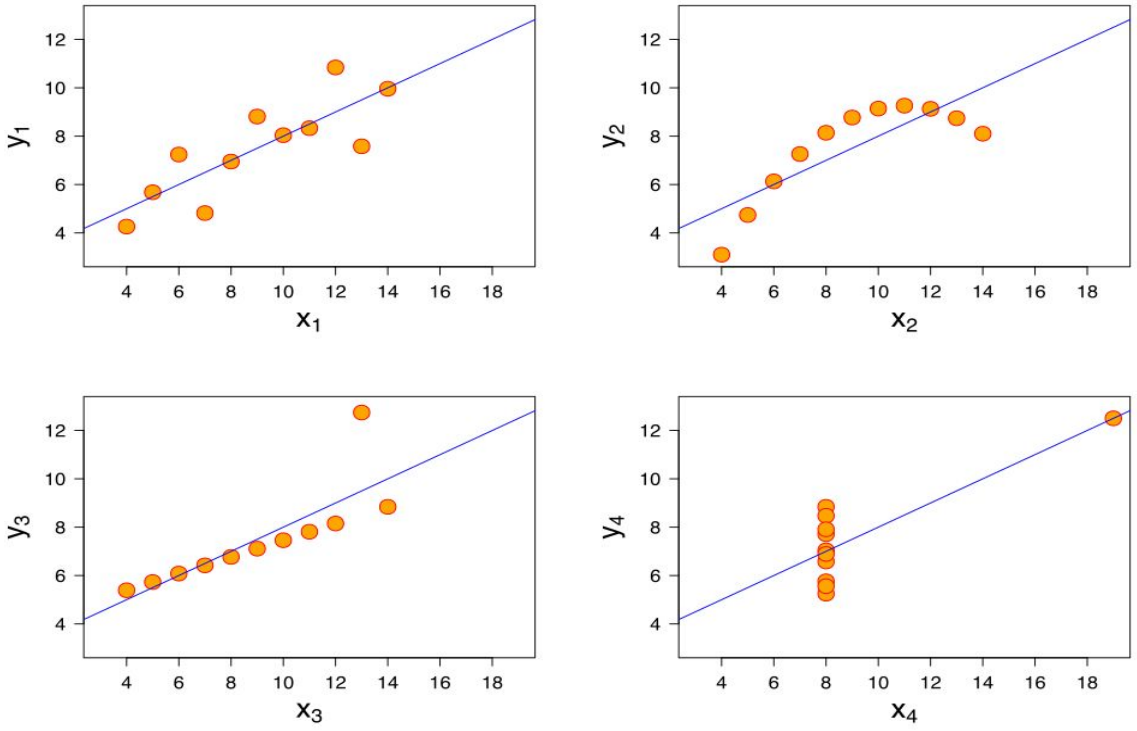
Προφανώς δεν είναι εφικτό να βρεθεί ένα πραγματικό dataset το οποίο να πληροί αυστηρά όλες τις παραπάνω προϋποθέσεις. Γι' αυτο ακριβώς και γίνεται αναφορά σε παραδοχές. Γίνεται λοιπόν η παραδοχή ότι ισχύουν και αυτό σημαίνει ότι το dataset πρέπει να παρουσιάζει ιδιότητες όσο το δυνατόν κοντινότερες στις προϋποθέσεις αυτές . Σ' αυτήν την περίπτωση οι εκτιμήσεις που θα γίνουν θα είναι ικανοποιητικές αλλιώς χρειαζόμαστε διαφορετικό μοντέλο παλινδρόμησης.

Στις εικόνες 4.5-4.9 παρουσιάζονται 4 γραφήματα τα οποία αποτελούν το κουαρτέτο του Anscombe [43]. Βλέπουμε 4 διαφορετικές ομάδες παρατηρήσεων με ιδιάζουσα χαρακτηριστικά οι οποίες προσεγγίζονται από την ίδια γραμμή γραμμικής παλινδρόμησης.

[21] *[Πηγή: [42]]*



Εικόνα 4.4: Ευθεία γραμμικής παλινδρόμησης για την εκτίμηση της διάρκειας της προετοιμασίας προσφορών που λαμβάνει ένας σύμβουλος ηλεκτρικής ενέργειας σε σχέση με τον αριθμό προσφορών που έχει λάβει από τους πελάτες του.[22]



Εικόνα 4.5: Παρατηρήσεις που πληρούν τις προϋποθέσεις γραμμικής παλινδρόμησης

Εικόνα 4.6: Μη γραμμικές παρατηρήσεις. Δεν ακολουθούν κανονική κατανομή.

Εικόνα 4.7: Παρατηρήσεις με γραμμική συμπεριφορά αλλα με παρουσία ενός outlier που επηρεάζει έντονα την εκτίμηση

Εικόνα 4.8: Παρατηρήσεις με ασυσχέτιστες εξαρτημένη - ανεξάρτητη μεταβλητή. Ένα outlier δίνει την εντύπωση γραμμικής εξάρτησης χρησιμοποιώντας μοντέλο γραμμικής παλινδρόμησης.

[22] *Πηγή: [41]*

4.5.1.2 Μέθοδος ελαχίστων τετραγώνων

Η εκτίμηση της ευθείας της απλής γραμμικής παλινδρόμησης γίνεται με τη γνωστή μέθοδο των ελαχίστων τετραγώνων η οποία μπορεί να επεκταθεί και στην σταθμισμένη και γενικευμένη μορφή της. Η μέθοδος αυτή ουσιαστικά αποτελεί μια συστηματική ελαχιστοποίηση του μέσου τετραγωνικού σφάλματος(mse). Γίνεται, λοιπόν, αναζήτηση των παραμέτρων β_0, β_1 της ευθείας $y = \beta_1 X + \beta_{o,}$ έτσι ώστε να ελαχιστοποιείται το άθροισμα των τετραγώνων των residuals δηλαδή η συνάρτηση:

$$RSS(\beta_1, \beta_0) = \sum_i r_i^2 = \sum_i (y_i - \hat{y}_i)^2 \qquad \text{(Εξίσωση 4.12)}$$

4.5.1.3 Αξιολόγηση του μοντέλου

Η αξιολόγηση του μοντέλου απλής γραμμικής παλινδρόμησης γίνεται με χρήση μετρικών όπως το R-squared, διαγραμμάτων residuals, και στατιστικών tests (F,t) όπως θα δούμε στη συνέχεια.

4.5.1.3.1 R-squared

Μετά την εφαρμογή της μεθόδου ελαχίστων τετραγώνων είναι απαραίτητο να ελέγξουμε την ποιότητα της εφαρμογής (Goodness of Fit) του μοντέλου στα δεδομένα του προβλήματος. Η κύρια μετρική που χρησιμοποιείται είναι το R-squared ή R^2. Το R^2 αποτελεί την εξής ποσότητα:

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\textit{Residual Sum of Squares}}{\textit{Total Sum of Squares}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} = \frac{\sum_i (\hat{y}_i - \bar{y})^2}{\sum_i (y_i - \bar{y})^2} = \frac{\text{Διακύμανση που εξηγείται απο το μοντέλο}}{\text{Συνολική Διακύμανση}}$$

(Εξίσωση 4.13)

Το R^2 μπορεί να πάρει τιμές στο διάστημα [0,1] και σε γενικές γραμμές όσο μεγαλύτερο είναι, είναι μια καλή ένδειξη ότι το μοντέλο είναι καλό, χωρίς όμως να είναι αρκετή. Απαιτείται και η εξέταση των γραφικών των residuals (residual plots). Επίσης υπάρχουν συγκεκριμένα datasets που αφορούν για παράδειγμα ζητήματα ανθρώπινης συμπεριφοράς όπου το R^2 παρουσιάζει, κατά κανόνα, τιμές μικρότερες του 50% χωρίς να σηματοδοτεί πρόβλημα στο μοντέλο. $R^2 < 0$ υποδηλώνει πολυ κακή εφαρμογή και πως η μέση τιμή των παρατηρήσεων ερμηνεύει τα δεδομένα καλύτερα από το μοντέλο που χρησιμοποιήθηκε.

4.5.1.3.2 P-value

Τα δύο τεστ των επόμενων ενοτήτων βασίζονται στην ιδέα της στατιστικής σημαντικότητας. Και στα δυο τεστ γίνεται αναφορά στις δύο παρακάτω υποθέσεις:

- Μηδενική Υπόθεση (Null Hypothesis) H_0: $y = \beta_o \Leftrightarrow \beta_1 = 0$
  Η μηδενική υπόθεση εκφράζει την περίπτωση όπου η εξαρτημένη μεταβλητή είναι εντελώς ασυσχέτιστη με την ανεξάρτητη μεταβλητή εφόσον η πρόβλεψη της είναι ίση με το σταθερό όρο ο οποίος εκφράζει τη μέση τιμή των παρατηρήσεων.
- Εναλλακτική υπόθεση (Alternate Hypothesis) H_A: $y = \beta_1 X + \beta_o$

Στατιστικά μιλώντας, θεωρούμε ένα δείγμα n παρατηρήσεων που είναι τα δεδομένα μας και ικανοποιούν την H_A, και προσπαθούμε να εξετάσουμε ποια είναι η πιθανότητα αυτά τα δεδομένα να προέρχονται από ένα πληθυσμό ο οποίος ικανοποιεί την $H_{0.}$

4.5.1.3.3 F-test

Το F-test υπολογίζει μία παράμετρο f βασισμένη στις διασπορά των δεδομένων:

$$f = \frac{TSS - RSS}{\widetilde{\sigma}^2}, \qquad \text{(Εξίσωση 4.14)}$$

όπου $\widetilde{\sigma}^2$ *η διασπορά των παρατηρήσεων.*

Υπολογίζεται η μετρική P-value με αντιστοίχιση της τιμής της παραμέτρου f πανω σε μια f-κατανομή ή κατανομή Fisher–Snedecor (1,df), όπου df οι βαθμοί ελευθερίας. Στην απλή γραμμική παλινδρόμηση η βαθμοί ελευθερίας είναι n-2.

4.5.1.3.4 T-test

Το t-test υπολογίζει μία παράμετρο t βασισμένη στη μέση τιμή της κλίσης των δεδομένων:

$$t = \frac{\beta_1}{s_{\widehat{\beta 1}}}, \qquad \text{(Εξίσωση 4.15)}$$

όπου $s_{\widehat{\beta 1}} = \frac{\sqrt{\frac{\sum_i (y_i - \widehat{y}_i)^2}{df}}}{\sqrt{\sum_i (x_i - \widehat{x}_i)^2}}$*, df οι βαθμοί ελευθερίας. Στην απλή γραμμική παλινδρόμηση df = 2.*

Υπολογίζεται η μετρική P-value με αντιστοίχιση της τιμής της παραμέτρου t πανω σε μια t-κατανομή ή κατανομή Student βαθμού n.

Αν το P-value κάποιου εκ των τεστ αρκετά μικρο, δηλάδή κάτω από ένα κατώφλι, το οποίο συνήθως είναι στο διάστημα [0,05-0,1] τότε μπορούμε με ασφάλεια να απορρίψουμε τη μηδενική υπόθεση και έχουμε αποδείξει τη στατιστική σημαντικότητα της ανεξάρτητης μεταβλητής.

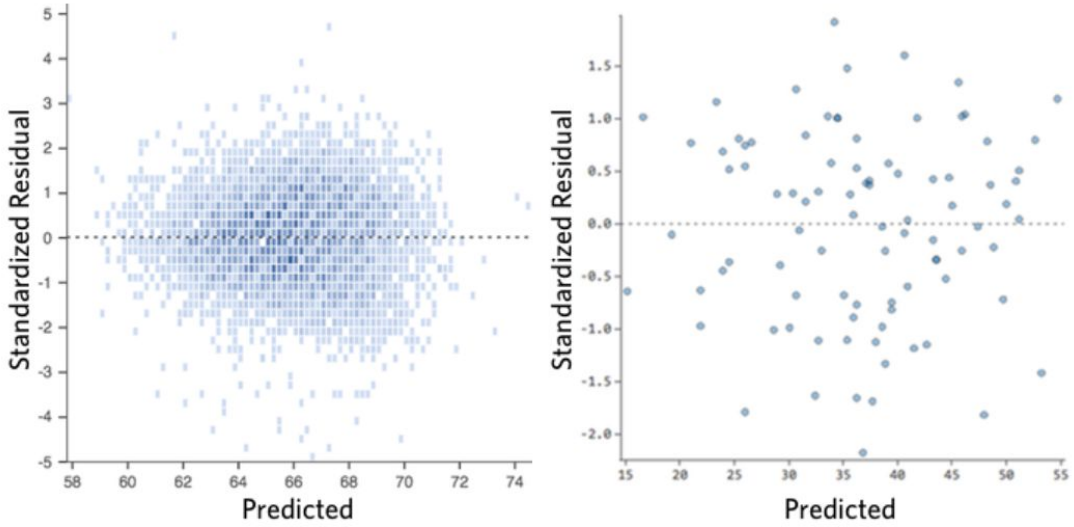
4.5.1.3.5 Γραφικές παραστάσεις των residuals γύρω από την αναμενόμενη τιμή τους (residual plot)

Στην απλή γραμμική παλινδρόμηση είναι απαραίτητο να γίνει έλεγχος του residual plot πριν αποφασιστεί η αποτελεσματικότητα του μοντέλου. Το γράφημα των residuals ενός καλού μοντέλου παλινδρόμησης διέπεται από τα εξής χαρακτηριστικά: (βλ. εικόνα 4.9)

- Συμμετρική κατανομή με μία τάση συγκέντρωσης γύρω από το κέντρο του διαγράμματος.
- Μικρές σχετικά αποκλίσεις από την οριζόντια γραμμή αναμενόμενων τιμών
- Απουσία συγκεκριμένων μοτίβων στη διάταξη τους.

Στην περίπτωση που παρατηρηθεί κάτι διαφορετικό από τα παραπάνω, το μοντέλο γραμμικής παλινδρόμησης ενδέχεται να παρουσιάζει ζητήματα όπως τα εξής:

- ❑ Ετεροσκεδαστικότητα που πιθανώς να υποδηλώνει απουσία κάποιας ανεξάρτητης μεταβλητής).
- ❑ Μη γραμμικότητα των δεδομένων, που συνεπάγεται ανάγκη για άλλο μη γραμμικό μοντέλο
- ❑ Παρουσία έντονου θορύβου (outliers), που καθιστά το μοντέλο μας biased. Πιθανώς, απαιτείται κάποιο τροποποιημένο μοντέλο λιγότερο ευαίσθητο στο θόρυβο (π.χ robust linear regression)



Εικόνα 4.9: Δύο περιπτώσεις όπου τα residuals είναι φυσιολογικά κατανεμημένα όπως θα περιμέναμε σε ένα καλό μοντέλο γραμμικής παλινδρόμησης.[23]

4.5.2 Πολλαπλή γραμμική παλινδρόμηση

Στην πολλαπλή γραμμική παλινδρόμηση το **X** είναι διάνυσμα k ανεξάρτητων μεταβλητών, $\boldsymbol{\beta}_1$ διάνυσμα k βαθμωτών σταθερών, β_0 σταθερός όρος. Πρόκειται λοιπόν για μια επέκταση κατα την οποία έχουμε k διαφορετικές και ανεξάρτητες εξισώσεις γραμμικής παλινδρόμησης να ισχύουν ταυτόχρονα για ένα σύνολο παρατηρήσεων, συνεπώς η Εξίσωση 4.1 επεκτείνεται ως εξής:

$$y = h(x) = \beta X + \varepsilon \qquad \text{(Εξίσωση 4.16)}$$

όπου:

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad X = \begin{pmatrix} \mathbf{x}_1^{\mathsf{T}} \\ \mathbf{x}_2^{\mathsf{T}} \\ \vdots \\ \mathbf{x}_n^{\mathsf{T}} \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

και n το μέγεθος του δείγματος παρατηρήσεων, p το πλήθος των ανεξάρτητων μεταβλητών. Εναλλακτικά γράφεται ως εξής:

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p \qquad \text{(Εξίσωση 4.17)}$$

4.5.2.1 Παραδοχές της πολλαπλής γραμμικής παλινδρόμησης

Στις προηγούμενες παραδοχές της απλής γραμμικής παλινδρόμησης (ενότητα 4.5.1.1) κάνουμε επέκταση τους για κάθε ανεξάρτητη μεταβλητή και τη σχέση της με την εξαρτημένη μεταβλητή. Επιπλέον, στην πολλαπλή γραμμική παλινδρόμηση γίνεται μία ακόμα παραδοχή:

- Ανεξαρτησία μεταξύ των μεταβλητών πρόβλεψης - Lack of Multicollinearity. Υπάρχουν διάφορα εργαλεία για να την εξετάσουμε (π.χ Condition Index, Variance Inflation Factor,

[23] *[Πηγή:http://docs.statwing.com/interpreting-residual-plots-to-improve-your-regression/#y-unbalanced-header]*

Tolerance, Correlation matrix). Δεν είναι, δηλαδή, δυνατόν να χρησιμοποιήσουμε μια ανεξάρτητη μεταβλητή για να προβλέψουμε μία άλλη.

4.5.2.2 Αξιολόγηση του μοντέλου

Στην πολλαπλή γραμμική παλινδρόμηση οι ανεξάρτητες μεταβλητές, έχουν γραμμική σχέση με την εξαρτημένη μεταβλητή. Πρόκειται για μία "υπέρθεση" ανεξάρτητων γραμμικών παλινδρομήσεων μεταξύ καθε ανεξάρτητης και της εξαρτημένης μεταβλητής. Οι ίδιο έλεγχοι της απλής γραμμικής παλινδρόμησης, πραγματοποιούνται και εδώ με ορισμένες τροποποιήσεις. Ωστόσο αυτή τη φορά υπεισέρχεται στο πρόβλημα μία ακόμα πρόκληση, η επιλογή ανεξάρτητων μεταβλητών που είναι οι καταλληλότερες και που τελικά θα συμμετάσχουν στην κατασκευή του μοντέλου και στην πρόβλεψη της εξαρτημένης μεταβλητής ανάλογα με το επίπεδο σημαντικότητας τους.

4.5.2.2.1 Adjusted R-squared

Η μετρική αυτή αποτελεί την εξής τροποποίηση του R^2, που μελετήθηκε στην ενότητα 4.5.1.3.1 Το απλό R^2 συνεχώς αυξάνεται όταν προσθέτουμε μεταβλητές στο πρόβλημα, οπότε γίνεται εισαγωγή του τροποποιημένου R^2 το οποίο ποινικοποιεί την προσθήκη νέων ανεξάρτητων μεταβλητών στο πρόβλημα και διατυπώνεται ως εξής:

$$R^2{}_{adj} = \frac{\left(1 - R^2\right)(n-1)}{n-k-1} \qquad \text{(Εξίσωση 4.18)}$$

4.5.2.2.2 Στατιστικά τεστ και επιλογή ανεξάρτητων μεταβλητών

Στην πολλαπλή γραμμική παλινδρόμηση τα στατιστικά τεστ είναι μείζονος σημασίας για την επιλογή των επιθυμητών απο τις διαθέσιμες ανεξάρτητες μεταβλητές και την κατασκευή του τελικού μοντέλου. Με τον υπολογισμό του P-value κάθε μεταβλητής υπολογίζεται η συγκριτική σημαντικότητα των ανεξάρτητων μεταβλητών και είτε εισάγουμε βήμα-βήμα νέες μεταβλητές στο μοντέλο (τεχνική forward selection), είτε να αφαιρούμε βήμα-βήμα τις πιο ασήμαντες μεταβλητές η οποίες και δε βελτιώνουν το μοντέλο (τεχνική backward elimination), είτε και τα δυο συνδυαστικά (τεχνική bidirectional elimination). Οι τεχνικές αυτές εντάσσονται στην ευρύτερη μεθοδολογία του stepwise regression η οποία αναλύεται στο [44].

4.5.3 Πολυωνυμική γραμμική παλινδρόμηση

Η πολυωνυμική γραμμική παλινδρόμηση αποτελεί, επι της ουσίας, μια ειδική περίπτωση της πολλαπλής γραμμικής παλινδρόμησης όπου:

$$\boldsymbol{X} = (x_1, x_{12} \ldots, x_{1k})^T \qquad \text{(Εξίσωση 4.19)}$$

Εξυπηρετεί όταν οι παρατηρήσεις υπό μελέτη εμφανίζουν κάποια μη γραμμικότητα (βλ. εικόνα 4.6) και συνεισφέρει σε μια καλύτερη προσαρμογή του μοντέλου στα δεδομένα εκπαίδευσης. Εντούτοις, η πολυωνυμική παλινδρόμηση είναι πολύ επιρρεπής σε overfitting, πόσο μάλλον όταν το k παίρνει μεγάλες τιμές.

4.5.4 Ridge regression, Lasso regression, ElasticNet regression

Χαρακτηριστικό είναι, όπως ήδη αναφέραμε στην πολυωνυμική παλινδρόμηση, το πως η πολυπλοκότητα του μοντέλου προσαρμόζεται αφενός καλύτερα στα δεδομένα, αφετέρου μετα από ένα σημείο επιφέρει ανεπιθύμητο overfitting. Επιπλέον, κατά κανόνα, η αύξηση της πολυπλοκότητας ενός πολυωνυμικού μοντέλου επιφέρει ταυτόχρονα αύξηση των απόλυτων τιμών των συντελεστών των ανεξάρτητων μεταβλητών, και κατα συνέπεια της βαρύτητας κάθε ανεξάρτητης μεταβλητής στο μοντέλο. Το πρόβλημα αυτό έρχονται να λύσουν οι αλγόριθμοι Ridge Regression [45] και Lasso Regression [46] μέσω της τεχνικής του L1, L2 regularization. Το regularization τροποποιεί τη συνάρτηση ελαχιστοποίησης E(h) ως εξής:

Ridge Regression:

$$E(h) = RSS + L2(coeff) = \sum_i \left(y_i - \hat{y}_i\right)^2 + a_1 \sum_j \left(coeff_j\right)^2$$

(Εξίσωση 4.20)

Lasso Regression:

$$E(h) = RSS + L1(coeff) = \sum_i \left(y_i - \hat{y}_i\right)^2 + a_2 \sum_j \left|coeff_j\right|$$

(Εξίσωση 4.21)

ElasticNet regression:

$$E(h) = RSS + L2(coeff) = \sum_i \left(y_i - \hat{y}_i\right)^2 + a_1 \sum_j \left(coeff_j\right)^2 + a_2 \sum_j \left|coeff_j\right|$$

(Εξίσωση 4.22)

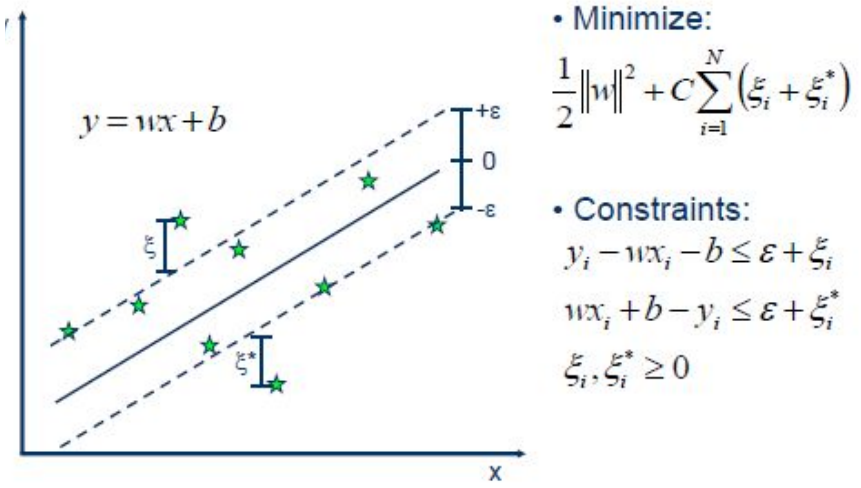
όπου:
$coeff_j$ ο j-οστός συντελεστής του πολυωνυμικού μοντέλου,
α_1,α_2 οι συντελεστές κανονικοποίησης L1,L2 επιπέδου 1,2 αντίστοιχα (regularization factors)

Οι παραπάνω τεχνικές:

- Ελέγχουν το overfitting της πολυωνυμικής αλλά και το underfitting της γραμμικής παλινδρόμησης κρατώντας τις εξαρτήσεις μεταξύ ανεξάρτητων και εξαρτημένων μεταβλητών υπό έλεγχο. Οι παράμετροι alpha (α_1,α_2) αποτελούν τις εξωτερικές παραμέτρους (υπερπαραμέτρους) των μοντέλων και συνεισφέρουν στην επιθυμητή προσαρμογή του στα δεδομένα.
- Βασικό πλεονέκτημα των παραπάνω τεχνικών, εν γένει, είναι ότι απαντούν αποτελεσματικά στο πρόβλημα του multicollinearity, δηλαδή της αλληλεξάρτησης μεταξύ εξαρτημένων μεταβλητών. Κυρίως όσον αφορά στο Ridge Regression, έχει πολύ ενδιαφέρον η έρευνα που έχει γίνει προς αυτή την κατεύθυνση για μεθόδους προσδιορισμού της παραμέτρου ridge δηλαδή της α_1. Μια γενικη εικόνα της έρευνας αυτής μπορεί κανείς να αποκτήσει στο [47].
- Επιπλέον, πλεονέκτημα συγκεκριμένα της τεχνικής Lasso είναι ότι πραγματοποιεί και κάποιου είδους feature selection καθώς λόγω του απόλυτου χαρακτήρα της έχει την τάση να μηδενίζει εντελώς αρκετούς πολυωνυμικούς συντελεστές. Σχετική ανάλυση γίνεται στα [48], [49].

4.5.5 Παλινδρόμηση με διανύσματα υποστήριξης - SVR

Δεδομένης της ενότητας 3.3.4.3 των μηχανών διανυσμάτων υποστήριξης (SVM) για ταξινόμηση, ο αλγόριθμος SVR αποτελεί μία επέκταση αυτού για παλινδρόμηση με τις εξής τροποποιήσεις στους περιορισμούς και τη συνάρτηση ελαχιστοποίησης:



Εικόνα 4.10: Αλγόριθμος SVR

4.5.6 Παλινδρόμηση με δέντρα αποφάσεων - τυχαίου δάσους

Καλό είναι να επισκεφθεί πρώτα κάποιος την ενότητα 3.3.3.1 της ταξινόμησης με δέντρα αποφάσεων., ώστε να υπάρχει γνώση όλων των λεπτομερειών σχετικα με τους ταξινομητές δέντρων αποφάσεων και τυχαίου δάσους. Αρκεί να αναφέρουμε πως η παλινδρόμηση ακολουθεί την ίδια ακριβώς λογική, ωστόσο κάθε περιοχή που στην ταξινόμηση αντιπροσωπεύει μια κλάση στην περίπτωση της παλινδρόμησης αντιπροσωπεύει μια τιμή και η τιμή αυτή είναι ο μέσος όρος των στοιχείων εκπαίδευσης τα οποία ανήκουν στην περιοχή αυτή.

Στην περίπτωση του τυχαίου δάσους, η οποία είναι αποτέλεσμα bagging δέντρων αποφάσεων, για κάθε νέο σημείο που θέλουμε να προβλέψουμε επιλέγουμε το μέσο όρο των τιμών που δίνουν τα δέντρα που συνιστούν το δάσος.

4.5.7 Robust regression

Είναι γνωστό το πόσο ευαίσθητες είναι οι μέθοδοι παλινδρόμησης, λόγω της ελαχιστοποίηση μέσου τετραγωνικού σφάλματος, σε outliers. Γι' αυτό έχουν διεκπεραιωθεί πολυάριθμες μελέτες σε μοντέλα, τα οποία λαμβάνουν σοβαρά υπόψη το γεγονός αυτό ώστε να εξομαλύνουν τις διαταραχές που προκαλούν τα outliers. Το πρόβλημα που καλούνται να λύσουν οι αλγόριθμοι αυτοί γίνεται εμφανές στην εικόνα 4.7 της ενότητας 4.5.1. Χαρακτηριστικά είναι τα μοντέλα στιβαρής παλινδρόμησης η Robust Regression τα οποία αναλύονται με λεπτομέρεια στο [50].

4.5.8 MARS regression (Multivariate Adaptive Regression Splines)

Η μέθοδος αυτή θεμελιώθηκε στο [51]. Αποτελεί μια γενικευμένη μη παραμετρική μέθοδο παλινδρόμησης η οποία κάνει αυτόματη μοντελοποίηση της μη γραμμικότητας και των συσχετίσεων μεταξύ των μεταβλητών. Η μέθοδος αποτελεί υπέρθεση επιμέρους μοντέλων γραμμικής παλινδρόμησης με χρήση συναρτήσεων Hinge, δηλαδή συναρτήσεων της μορφής max(0, x-constant)

ή max(0,constant-x). Το μοντέλο διαμορφώνεται, κατα την εκπαίδευση ως εξής χωρίς την αρχικοποίηση εξωτερικών παραμέτρων:

$$\widehat{f}(x) = h(x) = \sum_{i=1}^{k} c_i B_i(x) \qquad \text{(Εξίσωση 4.23)}$$

όπου B_i σταθερά ή συνάρτηση Hinge ή γινόμενο συναρτήσεων Hinge.

Η διαδικασία μάθησης διακρίνεται στο forward και backward passes με τη σειρά. Το πρώτο προσαρμόζεται στο training set, συνήθως καταλήγοντας σε πλήρες overfitting. Το δεύτερο έρχεται να διαπιστώσει, ποιο υποσύνολο του τελικού μοντέλου του forward pass δίνει το βέλτιστο bias-variance tradeoff, μέσω μιας γενικευμένης διαδικασίας cross-validation.
Η τεχνικη MARS έχει τα εξής πλεονεκτήματα:

- Παρουσιάζει πολύ καλύτερη προσαρμογή (fit) από τα γραμμικά μοντέλα.
- Δε χρειάζεται αρχικοποίηση εξωτερικών παραμέτρων.
- Λειτουργει πολύ καλύτερα από τις αντίστοιχες αναδρομικές μεθόδους όπως τα δέντρα αποφάσεων σε προβλήματα παλινδρόμησης.
- Ενδεχομένως να παρουσιάζει χειρότερη προσαρμογή από τεχνικές όπως το gradient boosting ωστόσο ο αλγόριθμος είναι αρκετά γρηγορότερος στην εκτέλεση.
- Κρατάει σε καλά επίπεδα το bias-variance tradeoff. Αφενός μπορεί να γίνει αρκετά προσαρμοστικός σε μη γραμμικότητες, αφετερου η φύση των συναρτήσεων βάσης Hinge είναι γραμμική με μεγάλο bias, οπότε υπάρχει μια ισορροπία.
- Η αξιολόγηση του είναι σχετικά απλή και γίνεται μόνο με cross-validation.
- Δε χρειάζεται ιδιαίτερο preprocessing

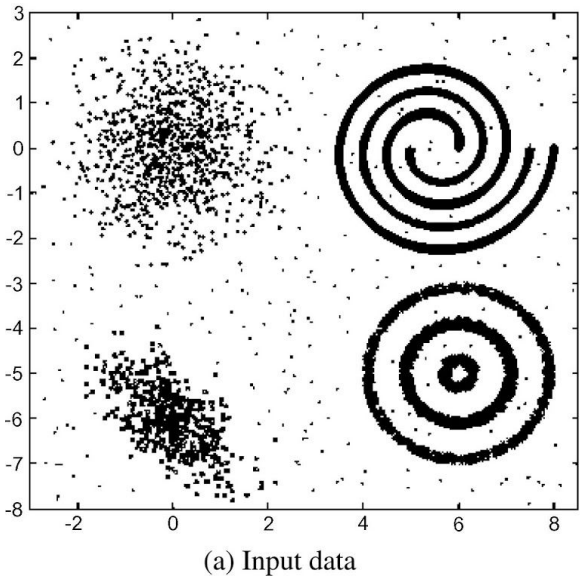
5. Συσταδοποίηση

5.1 Γενικα για τη συσταδοποίηση

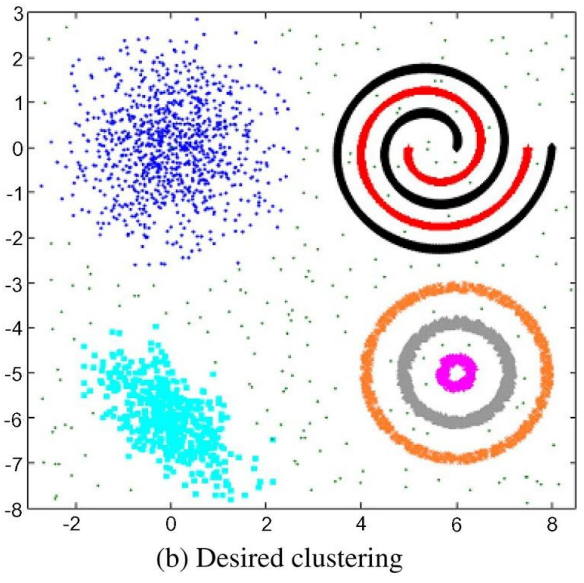
Ένας τυπικός ορισμός της συσταδοποίησης (clustering) είναι ο εξής:
Συσταδοποίηση καλείται η εργασία καταμερισμού ενός ετερογενούς πληθυσμού σε ένα σύνολο περισσοτέρων ετερογενών συστάδων (clusters), με την ιδιότητα τα στοιχεία της κάθε συστάδας να είναι ομοιότερα ανα μεταξύ τους απ' ότι με εκείνα των άλλων συστάδων [52],[53]. Η ομοιότητα, βεβαια, είναι μια σχετική έννοια και εξαρτάται κάθε φορά απο τη φύση του προβλήματος που καλούμαστε να λύσουμε. Τα σημεία πληροφορίας αναφέρονται συχνά στη βιβλιογραφία και ως διανύσματα ή πρότυπα.

Η συσταδοποίηση αποτελεί μια εργασία μη επιτηρούμενης μάθησης καθώς οι ομάδες δεν είναι γνωστές εκ των προτέρων. Αυτο έρχεται σε αντίθεση με τον επιτηρούμενο χαρακτήρα της ταξινόμησης(classification) η οποία έχει, εκ των προτέρων, γνώση σεσημασμένων δεδομένων (labeled data) με τα οποία εκπαιδεύεται ένα μοντέλο ωστε να κατηγοριοποιεί τα νέα δεδομένα με ετικέτες από ένα προκαθορισμένο σύνολο κλάσεων.

Επιπλέον η συσταδοποίηση δεν είναι μια αυτοματοποιημένη διαδικασία, με εκ των προτέρων εκπαιδευμένα μοντέλα που εφαρμόζονται πάνω σε δεδομένα. Απεναντίας πρόκειται για μια επαναληπτική διαδικασία σταδιακής "ανακάλυψης γνώσης" και βελτιστοποίησης που εμπεριέχει δοκιμές και αποτυχίες. Πολλές φορές χρειάζεται τροποποίηση της προεπεξεργασίας δεδομένων (data preprocessing), αλλαγή των παραμέτρων του μοντέλου και πιθανώς δοκιμή διαφορετικών αλγόριθμων συσταδοποίησης για να φτάσουμε στο επιθυμητό αποτέλεσμα.Στις εικόνες 5.1, 5.2 βλέπουμε ένα ενδεικτικό παράδειγμα συσταδοποίησης



Εικόνα 5.1: Ένα γράφημα σημείων πληροφορίας.[24]



Εικόνα 5.2: Το αποτέλεσμα της συσταδοποίησης των σημείων πληροφορίας. [16]

[24] *Πηγή: [54]*

5.2 Μαθηματικοί ορισμοί συσταδοποίησης

Δεδομένου ενός συνόλου διανυσμάτων $X = \{x_1, x_2, x_3 \ldots x_n\}$ *ζητούνται m σύνολα-συστάδες* $C_1, C_2 \ldots C_m$, *με* $m << n$ *έτσι ώστε το* C_i *να περιέχει στοιχεία για κάθε* $i = 1,2,3 \ldots m$ *και οι m ομάδες αποτελούν διαμέριση του συνόλου X.*

Ο ορισμός αυτός αναφέρεται στην αυστηρή συσταδοποίηση διότι κάθε διάνυσμα ανήκει σε μία και μόνο ομάδα. Εναλλακτικά μπορεί να οριστεί η ασαφής συσταδοποίηση:

Κάνοντας χρήση των ασαφών συνόλων μπορούμε να ορίσουμε m συναρτήσεις συμμετοχής $u_j: X \rightarrow [0,1]$ *για* $j = 1,2,\ldots m$. *Κάθε συνάρτηση* u_j *αντιστοιχίζει κάθε* x_i *του συνόλου X στην πιθανότητα που έχει να ανήκει στην ομάδα j.*

Ετσι τελικά κάθε στοιχείο του X καταλήγει να αντιστοιχίζεται στις πιθανότητες να ανήκει σε κάθε υπαρκτή συσταδα.

5.3 Έννοια της συστάδας και μοντέλα συσταδοποίησης

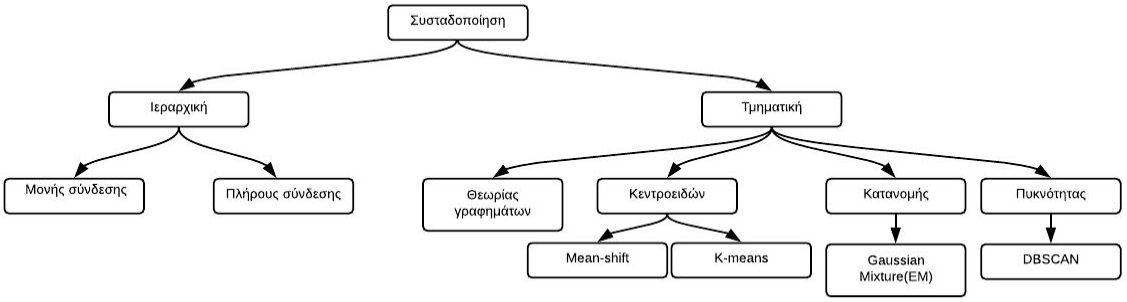
Η έννοια της συστάδας, η οποία θεωρείται δεδομένη στον παραπάνω μαθηματικό ορισμό, δε μπορεί να οριστεί επακριβώς πέρα από την ευρεία έννοια της "ομάδας" που δώσαμε παραπάνω. Γι αυτό ακριβώς το λόγο υπάρχουν πολλά διαφορετικά μοντέλα συσταδοποίησης και αντίστοιχοι αλγόριθμοι που χρησιμοποιούνται. Η έννοια της συστάδας αποκτά διαφορετικές ιδιότητες ανάλογα με τον αλγόριθμο από τον οποίο προκύπτει, επομένως η κατανόηση της απαιτεί την εξέταση των μοντέλων αυτών. Τα κυριότερα, λοιπον, μοντέλα είναι τα εξής:

Μοντέλα ιεραρχικής συσταδοποίησης: (Χαρακτηρίζονται από χαμηλές πολυπλοκότητες ενώ αποδίδουν καλά σε ισοτροπικούς σχηματισμούς συστάδων)

- Μοντέλα σύνδεσης(linkage): Διακρίνονται σε μονής και πλήρους σύνδεσης. Στη μονή σύνδεση η ομοιότητα μεταξύ 2 συστάδων υπολογίζεται βάσει της ελάχιστης απόστασης μεταξύ των σημείων τους ενώ στην πλήρη βάσει της μέγιστης[55], ωστόσο μπορεί να χρησιμοποιηθεί και η μέση απόσταση ή και η απόσταση κεντροειδών.

Μοντέλα τμηματικής συσταδοποίησης: (Χαρακτηρίζονται από υψηλότερες πολυπλοκότητες ενώ αποδίδουν καλά σε ανισότροπους, αλυσοειδείς και ομόκεντρους σχηματισμούς συστάδων[56])

- Μοντέλα κεντροειδών: Παράδειγμα αλγόριθμου είναι ο k-means όπου κάθε cluster εκπροσωπείται από το κέντρο βάρους (κεντροειδές) των στοιχείων του.
- Μοντέλα κατανομής: Οι συστάδες μοντελοποιούνται βασεις στατιστικων κατανομων όπως για παράδειγμα το μοντέλο μίξης κανονικών κατανομών (Gaussian mixture model) που χρησιμοποιεί τον αλγόριθμο Expectation-Maximization.
- Μοντέλα πυκνότητας: Οι συστάδες που δημιουργούνται αποτελούν συνδεδεμένες περιοχές πυκνωμάτων σημείων πληροφορίας. Χαρακτηριστικο παράδειγμα αποτελεί ο αλγόριθμος DBSCAN.



Εικόνα 5.3: Διάγραμμα της βασικής κατηγοριοποίησης των τεχνικών συσταδοποίησης.[25]

Συμπληρωματικά στην ταξινόμηση της εικόνας 5.3, υπάρχουν επιπλέον και οι παρακάτω προσεγγίσεις στα μοντέλα συσταδοποίησης όπως αναφέρεται στο[55]:

Συσσωρευτική (Agglomerative) - Διαιρετική (Divisive)

Η συσσωρευτική προσέγγιση ξεκινάει αρχικοποιώντας κάθε σημείο πληροφορίας ως ξεχωριστή συστάδα. Όσο εκτελούνται τα επαναληπτικά βήματα συσταδοποίησης τείνει να κατηγοριοποιήσει όλα τα διανύσματα σε μια συστάδα. Το αντίστροφο συμβαίνει στη διαιρετική προσέγγιση.

Αυστηρή (Hard) - Ασαφής (Fuzzy)

Στην ασαφή προσέγγιση ένα σημείο πληροφορίας μπορεί να ανήκει σε παραπάνω από μία συστάδα σε αντίθεση με την αυστηρή που μπορεί να ανήκει μόνο σε μία.

Ντετερμινιστικη - Στοχαστική

Αυτές οι προσεγγίσεις αναφέρονται κυρίως σε μοντέλα που επιδιώκουν την ελαχιστοποίηση ενός τετραγωνικού σφάλματος. Η ελαχιστοποίηση αυτή μπορεί να γίνει είτε με παραδοσιακές μεθόδους, είτε με στοχαστική βελτίστου μέσα σε όλες τις πιθανές συσταδοποιήσεις.

5.4 Στάδια της συσταδοποίησης

Εν γένει, η διαδικασία της συσταδοποίησης υλοποιείται στα παρακάτω στάδια όπως αναφέρεται στα [52],[55]:

- *Επιλογή χαρακτηριστικών γνωρισμάτων (feature selection)*

Ο στόχος είναι να επιλεγούν τα καταλληλότερα γνωρίσματα στα οποία πρόκειται να εφαρμοστεί η συσταδοποίηση ώστε να επιτυγχάνεται η βέλτιστη ομοιογένεια σε κάθε συστάδα. Συνεπώς, η προεπεξεργασία των δεδομένων πριν την εφαρμογή της διαδικασίας συσταδοποίησης κρίνεται απαραίτητη.

- *Επιλογή κατάλληλου αλγόριθμου συσταδοποίησης.*

Σε αυτό το στάδιο γίνεται η επιλογή ενός αλγόριθμου που θα οδηγήσει σε ένα καλό σχήμα συσταδοποίησης για ένα σύνολο δεδομένων. Για τη επιλογή του αλγόριθμου χρησιμοποιείται το μέτρο γειτνίασης και το κριτήριο συσταδοποίησης τα οποία ορίζουν απόλυτα τον αλγόριθμο, καθώς

[25] *Πηγή: [55, p. 275]*

επίσης και η δυνατότητά του να καθορίσει ένα σχήμα συσταδοποίησης που να προσαρμόζεται στο συγκεκριμένο σύνολο δεδομένων.

1. Το *μέτρο γειτνίασης (proximity measure)* αναφέρεται στην ομοιότητα δύο αντικειμένων (δηλαδή διανύσματα γνωρισμάτων). Η επιλογή των γνωρισμάτων πρέπει να γίνεται με τρόπο ώστε η συμβολή τους να είναι ανάλογη κατά τον υπολογισμό του μέτρου γειτνίασης και να μην υπερισχύει το ένα έναντι του άλλου.
2. Το *κριτήριο συσταδοποίησης (clustering criterion)* εκφράζεται βάσει μιας συνάρτησης κόστους, απόστασης ή κάποιου άλλου συνόλου κανόνων. Είναι σημαντικό να γνωρίζουμε τον τύπο των συστάδων που θα προκύψουν στο σύνολο δεδομένων, για να απιλέκουμε το κατάλληλο κριτήριο που θα ταιριάζει στο σύνολο δεδομένων και θα έχει ως αποτέλεσμα μία επιτυχημένη τμηματοποίηση.

- *Μείωση του όγκου δεδομένων(προαιρετικά).*

Στο στάδιο αυτό μπορούμε να κρατήσουμε ορισμένα αντιπροσωπευτικά συσταδοποιημένα δεδομένα για λόγους απλότητας ταχύτητας και ευκολίας. Ένα κλασικό παράδειγμα αντιπροσωπευτικών διανυσμάτων συσταδοποίησης αποτελούν τα κεντροειδή των συστάδων τα οποία από μόνα τους σε πολλά προβλήματα μας προσφέρουν πολλή γνώση με λίγη πληροφορία [55], [57].

- *Επικύρωση αποτελεσμάτων*.

Σε αυτή τη φάση αξιολογούνται τα αποτελέσματα του αλγόριθμου συσταδοποίησης σύμφωνα με κατάλληλα κριτήρια ορθότητας συσταδοποίησης και τεχνικές. Παράδειγμα ενός τέτοιου κριτηρίου είναι η σύγκριση των αποτελεσμάτων της ανάλυσης με κάποια ήδη γνωστά αποτελέσματα ή η σύγκριση των αποτελεσμάτων δύο διαφορετικών συσταδοποιήσεων. Η ποιότητα της συσταδοποίησης εξαρτάται από την ομοιότητα(δηλαδή μεγάλη ομοιότητα εντός της συστάδας - μικρή ομοιότητα μεταξύ των συστάδων) και την μέθοδο συσταδοποίησης.

- *Ερμηνεία των αποτελεσμάτων*.

Σ αυτό το στάδιο ο αναλυτής καλείται να εξάγει γνώση από τις παραχθείσες συστάδες, συνδυάζοντας κι άλλα στοιχεία με σκοπό τη βέλτιστη δυνατή επίλυση του προβλήματος.

5.5 Επιλογή κατάλληλου αριθμού συστάδων.

Σε αρκετούς αλγόριθμους clustering οπως είναι και ο k-means και ο gaussian mixture, ο αναλυτής καλείται να δώσει σαν είσοδο στον αλγόριθμο τον αριθμό k τον επιθυμητών συστάδων [54]. Το γεγονός αυτός χαρακτηρίζεται σα ‘‘δίκοπο μαχαίρι’’, καθώς όταν έχουμε διαισθητική επαφή με το τι θέλουμε να πετύχουμε μπορούμε να το κάνουμε πιο εύκολα, ωστόσο συχνά είναι πολύ περιοριστικό να πρέπει εξαρχής να επιλέξουμε τον αριθμό συστάδων. Για την επιλογή του κατάλληλου αριθμού συστάδων υπάρχουν ποικίλες μέθοδοι. Οι μέθοδοι αυτές διακρίνονται σε εμπειρικές(π.χ rule of thumb), γραφικές(π.χ elbow method, silhouette method), στατιστικές (π.χ gap statistic), θεωρίας πληροφορίας, μηχανικής μάθησης (cross-validation). Στη συνέχεια κάνουμε μια συνοπτική αναφορά στις πιο σημαντικές.

5.5.1 Εμπειρικός κανόνας (Rule of Thumb)

Ο εμπειρικός κανόνας αποτελεί μια πολύ πρόχειρη και διαισθητική μέθοδο η οποία δεν είναι ασφαλές να εφαρμοστεί σε προβλήματα clustering χωρίς την κατάλληλη εποπτεία του προβλήματος από τον αναλυτή, διότι θα οδηγήσει σε μη ικανοποιητικά αποτελέσματα. Ο κανόνας είναι ο εξής:

$$k = \sqrt{\frac{n}{2}} \qquad \text{(Εξίσωση 5.1)}$$

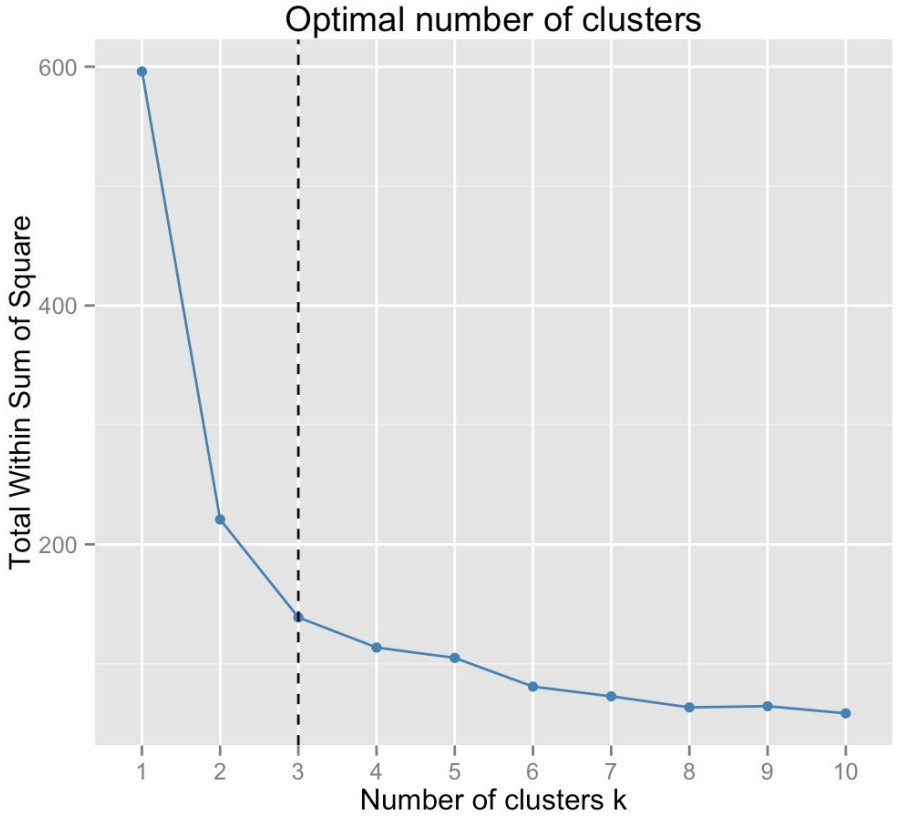
όπου n το πλήθος των σημείων πληροφορίας προς συσταδοποίηση.

5.5.2 Μέθοδος του "αγκώνα" (Elbow method)

Η μέθοδος αυτή είναι η πιο συνήθης και αυτη που χρησιμοποιήθηκε και κατά την εκπόνηση της εργασίας. Χρησιμοποιεί τη μετρική WSS(within-cluster sum of squares). Δεδομένου ενός συνόλου παρατηρήσεων $X = \{x_1,...,x_n\}$κάποιου αριθμού m από συστάδες C_i με κέντρα $K_{i,}$ όπου i ο αύξων αριθμός της κάθε συστάδας, η μαθηματική διατύπωση της μετρικής είναι η εξής:

$$` WSS = \sum_{i=1}^{m} \sum_{x_j \in C_i} \left\| x_j - \mathrm{K}_i \right\|^2 \qquad \text{(Εξίσωση 5.2)}$$

Η μετρική αυτή εκφράζει το άθροισμα των εντός συστάδας τετραγωνικών αποστάσεων για όλες τις συστάδες. Κατ' επέκταση είναι ένα μέτρο του πόσο κοντινά η όμοια είναι τα διανύσματα τα οποία ανήκουν στην ίδια συστάδα το οποίο στο τέλος αθροίζεται για όλες τις συστάδες μαζί. Στόχος μας ιδανικά είναι η ελαχιστοποίηση της απόστασης αυτής συναρτήσει του αριθμού k των συστάδων που θέτουμε. Το γράφημα WSS-k ωστόσο ακολουθεί φθίνουσα πορεία οπότε εμείς επιλέγουμε το ελάχιστο δυνατό k από το οποίο και μετά παρατηρείται πιο αργή πτώση της μετρικής. Αναζητούμε λοιπόν το k σημείου "αγκώνα" όπως βλέπουμε στην εικόνα 5.2. Ψάχνουμε έτσι, γραφικα, τη "χρυσή τομή" για μια ικανοποιητική συσταδοποίηση. Ωστόσο το σημείο του αγκώνα ενδέχεται να μην είναι πάντα εμφανές, γεγονός που αποτελεί προφανή αδυναμία της μεθόδου και που απαιτεί τη σωστή κρίση του αναλυτή ανάλογα με τη φύση του προβλήματος. Εν προκειμένω, θα μπορουσε να έχει επιλεγεί και η τιμή k = 4 ανάλογα με το πρόβλημα.



Εικόνα 5.4. Προσδιορισμό του σημείου αγκώνα, το οποίο αντιστοιχεί στον κατάλληλο αριθμό συστάδων.[26]

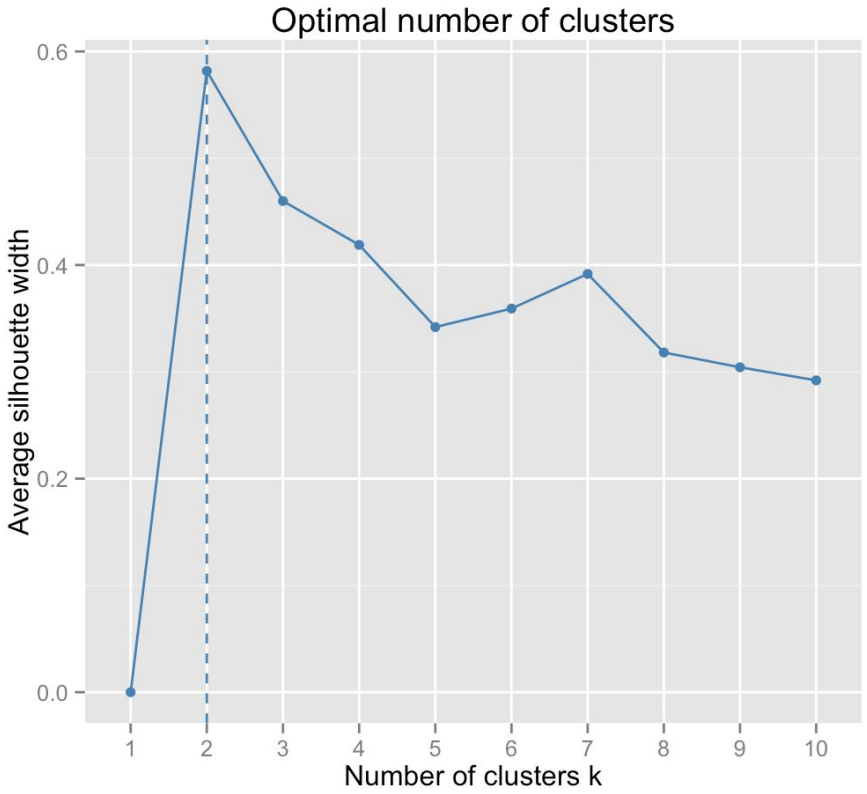
5.5.3 Μέθοδος μέσου εύρους σιλουέτας

Η μέθοδος αυτή χρησιμοποιεί τη μετρική του εύρους σιλουέτας. Έχουμε ως δεδομένα ένα σύνολο παρατηρήσεων $X = \{x_1,...,x_n\}$ και κάποιο αριθμό m από συστάδες C_j με κέντρα K_j, όπου j ο αύξων αριθμός της κάθε συστάδας. Θέτουμε α(i) να είναι η μέση απόσταση του x_i από τα σημεία της συστάδας $C_a \ni x_i$ και b(i) να είναι η ελάχιστη εκ των μέσων αποστάσεων του στοιχείου x_i από τα στοιχεία έκαστης συστάδας C_j , για i τέτοια ώστε $x_i \notin C_j$. Η μαθηματική διατύπωση της μετρικής είναι η εξής:

$$s(i) = \frac{b(i) - a(i)}{max\{b(i), a(i)\}}, -1 \leq s(i) \leq 1 \qquad \text{(Εξίσωση 5.3)}$$

Παρατηρούμε ότι μια αποτελεσματική συσταδοποίηση επιτάσσει μικρές τιμές α(i) (ενδοσυσταδική ομοιότητα - intracluster similarity) και μεγάλες τιμές β(i) (διασυσταδική ανομοιότητα - intercluster dissimilarity). Το γράφημα του εύρους σιλουέτας συναρτήσει του αριθμού των συστάδων μας δίνει τη δυνατότητα να προσδιορίσουμε των επιθυμητό αριθμό συστάδων απο το σημείο στο οποίο παρουσιάζει μέγιστο (βλ. εικόνα 5.3).

[26] Πηγή: http://www.sthda.com/english/wiki/print.php?id=239



Εικόνα 5.5. Γράφημα προσδιορισμού σημείου αγκώνα, το οποίο αντιστοιχεί στον κατάλληλο αριθμό συστάδων.[27]

5.6 Κυριότεροι αλγόριθμοι συσταδοποίησης

5.6.1 Αλγόριθμος k-means

5.6.1.1 Περιγραφή αλγόριθμου

Ο αλγόριθμος k-means, ο οποίος πρωτοδημοσιεύτηκε στο [58], αποτελεί έναν αλγόριθμο τμηματικής συσταδοποίησης και στην απλή του μορφή απαιτεί αρχικοποίηση k τυχαίων σημείων, τα οποία ονομάζονται κεντροειδή της συστάδας και αντιπροσωπεύουν το κέντρο βάρους της συστάδας. Ο αριθμός k ορίζει πόσες συστάδες θέλουμε να δημιουργηθούν από τον αλγόριθμο. Είναι πολύ σημαντικό να έχουμε υπόψη πως η ανάγκη ορισμού του k όπως αναφέρθηκε και παραπάνω είναι μια παράμετρος η οποία είναι συχνα βολική, ωστόσο υπάρχουν περιπτώσεις στις οποίες δυσκολεύει πολύ το έργο του αναλυτή και για το λόγο αυτό έχουν προταθεί ποικίλες εναλλακτικές υλοποιήσεις του αλγόριθμου οι οποίες κάνουν εκτίμηση του k, όπως ο X-means [59].

Τα δύο βασικότερα στάδια του αλγόριθμου τα οποία και εκτελούνται επαναληπτικά αφορούν πρώτον την ανάθεση σε κάποια συστάδα και δεύτερον στον επαναπροσδιορισμό και τη μετατόπιση του κεντροειδούς κάθε συστάδας. Ο k-means ενδείκνυται κυρίως για συσταδοποίηση ισοτροπικών σχηματισμών συστάδων. Αναλύεται σε βάθος καθώς έγινε χρήση του στο πρόβλημα συσταδοποίησης που αντιμετωπίστηκε κατά την εκπόνηση της εργασίας.

[27] *Πηγή*: http://www.sthda.com/english/wiki/print.php?id=239

Βήματα αλγόριθμου:

Δεδομένου ενός συνόλου παρατηρήσεων A (dataset) το οποίο ανήκει σε έναν διανυσματικό χώρο $\Omega = \mathbb{R}^d$, ο αλγόριθμος k-means διαρθρώνεται στα εξής βήματα:

1. Επίλεξε τον επιθυμητό αριθμό k συστάδων.
2. Επίλεξε k αυθαίρετα διανύσματα (κεντροειδή): $C = \{c_1, c_2, ...c_k\} \subseteq \Omega$.
3. Με βάση ένα μέτρο απόστασης, για κάθε $i \in \{1, \ldots, k\}$, θέσε τη συσταδα C_i να είναι τα διανύσματα του **A** τα οποία είναι κοντινότερα στο κέντρο της c_i απ' ότι σε οποιοδήποτε άλλο κέντρο c_j, $i \neq j$.
4. Για κάθε $i \in \{1, \ldots, k\}$ υπολόγισε το νέο κεντροειδές c_i κάθε συστάδας ως το κέντρο βάρους των στοιχείων της.
5. Αν δεν πληρείται κανένα κριτήριο σύγκλισης επίστρεψε στο βήμα 3 ειδάλλως η διαδικασία ολοκληρώθηκε.

Επεξηγήσεις:

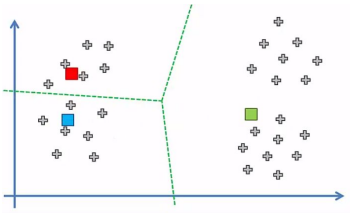
Μαθηματικά το βήμα 4 ισοδυναμεί με την ομαδοποίηση των παρατηρήσεων βάσει του διαγράμματος Voronoi των κεντροειδών του εκάστοτε επαναληπτικού βήματος

Το απλούστερο κριτήριο σύγκλισης είναι στο τρέχων βήμα να μην έχουμε καμία ανακατάταξη, ωστόσο συνήθως δε φτάνουμε σε αυτό το σημείο αλλα τερματίζουμε νωρίτερα τον αλγόριθμο με κάποιο από τα κριτήρια που εξετάζονται στην ενότητα 5.5.

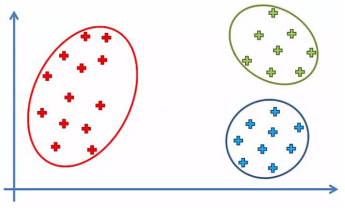
5.6.1.2 Παγίδα τυχαίας αρχικοποίησης κεντροειδών - αλγόριθμος k-means++

Η τυχαία αρχικοποίηση των κεντροειδών του αλγόριθμου k-means κάποιες φορές δημιουργεί προβλήματα ειδικά αν είναι εντελώς αυθαίρετη και ο προγραμματιστής δεν έχει μια εποπτεία του χώρου διανυσμάτων. Στην εικόνα 5.4, όπου έχουμε ένα σύνολο δεδομένων δύο γνωρισμάτων θα μπορούσαμε με μία επιλογή αρχικών κεντροειδών όπως της εικόνας 5.5 να καταλήξουμε στη συσταδοποίηση της εικόνας 5.6, αποτέλεσμα το οποίο διαισθητικά και μόνο δεν είναι ικανοποιητικό. Απεναντίας με μία αρχικοποίηση σαν αυτή της εικόνας 5.7, θα καταλήγαμε στο επιθυμητό αποτέλεσμα της εικόνας 5.8.

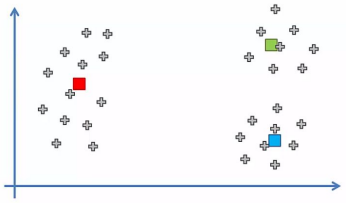
Εικόνα 5.6



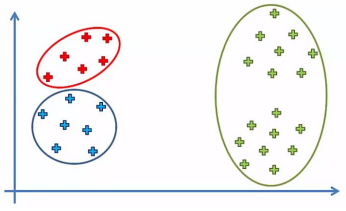
Εικόνα 5.7



Εικόνα 5.8



Εικόνα 5.9



Εικόνα 5.10

Είναι λοιπόν εμφανές πως η αρχικοποίηση των κεντροειδών στον αλγόριθμο k-means επηρεάζει το τελικό αποτέλεσμα συσταδοποίησης, πράγμα ανεπιθύμητο. Το πρόβλημα αυτό ήρθε να λύσει ο αλγόριθμος k-means++ ο οποίος προτάθηκε στο [60] και αποτελεί μία επέκταση,

λογαριθμικής πολυπλοκότητας, η οποία ορίζει ποια θα είναι τα αρχικά κέντρα με τα οποία θα εκκινήσει ο παραδοσιακός k-means. Έχουμε ως δεδομένα ένα σύνολο παρατηρήσεων A (dataset) το οποίο ανήκει σε έναν διανυσματικό χώρο $\Omega = \mathbb{R}^d$ και έστω D(x) η ελάχιστη απόσταση ενός διανύσματος x από το κοντινότερο κεντροειδές. Ο αλγόριθμος k-means++ διαρθρώνεται, στα εξής βήματα:

1. Επίλεξε τον επιθυμητό αριθμό k συστάδων.
2. Επίλεξε τυχαία και ισοπίθανα απο τον A το πρώτο κεντροειδές $c_1 \in A$.
3. Επίλεξε το επόμενο κεντροειδές $c_1 \in A$ με πιθανότητα: $\frac{D(c_i)^2}{\sum_{x \in \Omega} D(x)^2}$.
4. Επανάλαβε το βήμα 3 μέχρις ότου να δημιουργηθούν k κεντροειδή.

<u>Επεξήγηση:</u> Στο βήμα 3 ευνοούμε για να επιλεγούν τα σημεία του dataset τα οποία απέχουν πιο πολύ από τα ήδη επιλεγμένα.

5.6.1.3 Κώδικας αλγόριθμου

Η κλάση της βιβλιοθήκης scikit-learn που υλοποιεί τον αλγόριθμο σε python:
class sklearn.cluster.**KMeans**(*n_clusters=8*, *init='k-means++'*, *n_init=10*, *max_iter=300*, *tol=0.0001*, *precompute_distances='auto'*, *verbose=0*, *random_state=None*, *copy_x=True*, *n_jobs=1*, *algorithm='auto'*)

5.6.2 Αλγόριθμοι ιεραρχικής συσταδοποίησης

5.6.2.1 Περιγραφή αλγόριθμων

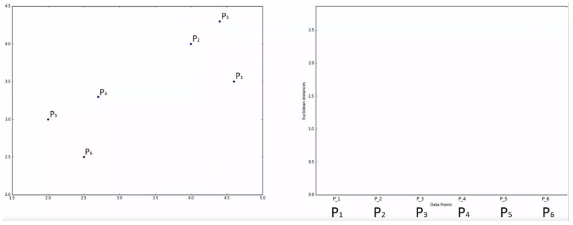
Οι αλγόριθμοι ιεραρχικής συσταδοποίησης χαρακτηρίζονται από διαδοχική αύξηση (διαιρετική προσέγγιση) ή μείωση (συσσωρευτική προσέγγιση) του αριθμού των συστάδων. Επιπλέον η διάταξη της αυξομείωσης αυτής καθορίζεται από ένα μέτρο διασυσταδικής απόστασης - ανομοιότητας το οποίο διαφέρει ανάλογα με το αν πρόκειται για μοντέλο μονής η πλήρους σύνδεσης. Στη συνέχεια θα εξετάσουμε ένα αλγόριθμο ιεραρχικής συσταδοποίησης, συσσωρευτικής προσέγγισης. Οι υπολοιπες προσεγγίσεις διαρθρώνονται με ανάλογο τρόπο.

Βήματα αλγόριθμου συσσωρευτικής προσέγγισης:
Τα βήματα του αλγόριθμου ιεραρχικής συσταδοποίησης, συσσωρευτικής προσέγγισης είναι τα εξης:

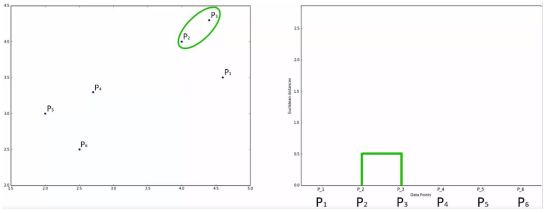
1. Τοποθέτησε κάθε διάνυσμα σε μία ξεχωριστή σημειακή συστάδα.
2. Κατασκεύασε διάγραμμα συστάδων-ανομοιότητας (δενδρόγραμμα).
3. Συγχώνευσε τις δύο κοντινότερες μεταξύ τους συστάδες (σύμφωνα με το επιλεχθέν μέτρο διασυσταδικής απόστασης)
4. Φέρε ευθύγραμμο τμήμα παράλληλο στον οριζόντιο άξονα που εκτείνεται μεταξύ των σημείων των δύο συστάδων που συγχωνεύθηκαν στο βήμα 3 και βρίσκεται σε ύψος τόσο όση η ανομοιότητα των δύο συστάδων. Ένωσε τα άκρα του με κατακόρυφες γραμμές με τον οριζόντιο άξονα
5. Επανάλαβε τα βήματα 3-4 μέχρις ότου όλα τα αρχικά σημεία-συστάδες του οριζόντιου άξονα του δενδρογράμματος να συνδέονται μεταξύ τους μέσω ενός πλήρως διασυνδεδεμένου γράφου. Εναλλακτικά μέχρι να καταλήξεις σε μία και μοναδική συστάδα.

Σημείωση: Όταν έχει πλέον σχηματιστεί το δενδρόγραμμα, επιλέγουμε το κατώφλι ανομοιότητας της επιλογής μας, το οποίο αυτόματα ορίζει και έναν αριθμό από συστάδες στο δενδρόγραμμα. Συχνά εξυπηρετεί να επιλέξουμε το κατώφλι αυτό κατα μήκος της πιο μακριάς κατακόρυφης γραμμής του δενδρογράμματος, μετρώντας το μήκος τους από πάνω πρός τα κάτω και μέχρι το σημείο στο οποίο τέμνουν οποιαδήποτε οριζόντια γραμμή η νοητή προέκταση της.

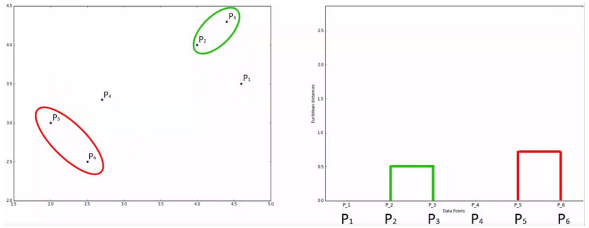
Στις εικόνες 5.11, 5.12, 5.13, 5.14, 5.15, 5.16 βλέπουμε ένα παράδειγμα εκτέλεσης του αλγόριθμου μαζι με την παράλληλη εκτέλεση του δενδρογράμματος. Στην εικόνα 5.16 φαίνεται και ο προτεινόμενος εντοπισμός του κατάλληλου αριθμού των συστάδων.



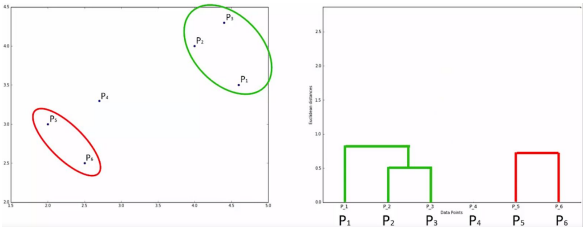
Εικόνα 5.11. Αρχική κατάσταση: Κάθε σημείο αποτελεί μια συστάδα



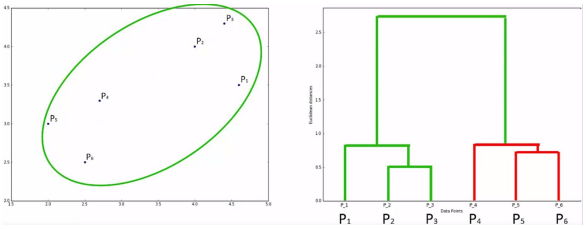
Εικόνα 5.12. Συγχώνευση 2 πιο κοντινών σημείων-συστάδων. Ενημέρωση δενδρογράμματος.



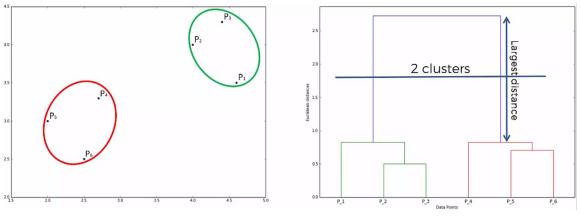
Εικόνα 5.13: Συγχώνευση των 2 επόμενων πιο κοντινών συστάδων. Ενημέρωση δενδρογραμματος



Εικόνα 5.14. Συνέχεια διαδικασίας.



Εικόνα 5.15. Τελικό βήμα. Όλα τα σημεία βρίσκονται σε μια συστάδα. Στο δενδρόγραμμα είναι πλέον διασυνδεδεμένα.



Εικόνα 5.16. Στο στάδιο αυτό επιλέγουμε τον κατάλληλο αριθμό συστάδων.

5.6.2.2 Κώδικας αλγόριθμων

Η κλάση της βιβλιοθήκης scikit-learn που υλοποιεί τον συσσωρευτικό εκ των ιεραρχικών αλγόριθμων σε python:

class sklearn.cluster.**AgglomerativeClustering**(*n_clusters=2*, *affinity='euclidean'*, *memory=None*, *connectivity=None*, *compute_full_tree='auto'*, *linkage='ward'*, *pooling_func=<function mean>*)

5.6.3 Αλγόριθμος DBSCAN

5.6.3.1 Περιγραφή αλγόριθμου

Ο αλγόριθμος DBSCAN(density based algorithm for discovering clusters) θεμελιώθηκε στο [61] και βασίζεται στην πυκνότητα των σημείων πληροφορίας. Έχει τα εξής πλεονεκτήματα:

- Δε χρειάζεται ορισμό του αριθμού των συστάδων.
- Δεν επηρεάζεται απο τους σχηματισμούς των συστάδων στο χώρο συνεπώς μπορεί να ανιχνεύσει συστάδες με αυθαίρετα σχήματα, αποστάσεις και κατανομές στο χωρο.
- Είναι αρκετά αποτελεσματικός σε datasets με έντονο θόρυβο και απομακρυσμένα σημεία (outliers)

Ωστόσο απαιτεί αρχικοποίηση δύο βασικών παραμέτρων που καθορίζουν, το αποτέλεσμα συσταδοποίησης για δεδομένο dataset. Επιπλέον, οι τιμές που παίρνουν, για να υπάρξουν ικανοποιητικά αποτελέσματα, αποτελούν αντικείμενο διερεύνησης. Οι παράμετροι αυτές είναι οι εξής:

ε: Είναι η μέγιστη απόσταση μεταξύ δύο σημείων ώστε αυτά να θεωρούνται *"γειτονικά"*.
minPoints: Είναι ο ελάχιστος αριθμός σημείων που απαιτούνται για να συσταθεί μια "*πυκνή περιοχή"*

Για την εκτέλεση του κατηγοριοποιούμε τα σημεία πληροφορίας σε *σημεία πυρήνα, σημεία άμεσης εμβέλειας, σημεία έμμεσης εμβέλειας* και *σημεία περιφέρειας*, όπως θα δούμε στη συνέχεια:

- Ένα σημείο p είναι *σημείο πυρήνα* όταν έχει τουλάχιστον *minPts γειτονικά σημεία*. Τα σημεία αυτα ονομάζονται *σημεία άμεσης εμβέλειας* του p.
- Ένα σημείο w είναι σημείο έμμεσης εμβέλειας του p όταν υπάρχει μονοπάτι σημείων πυρήνα αμεσης εμβέλειας(p →...→ w) που να τα συνδέει.
- Ένα σημείο το οποίο δεν είναι γειτονικό με κάποιο άλλο ονομάζεται *απομακρυσμένο σημείο (outlier)*.

Η συσταδοποίηση, λοιπόν, διαρθρώνεται ως εξής:

- Η κάθε συστάδα διαμορφώνεται από ένα σημείο πυρήνα και όλα τα σημεία άμεσης και έμμεσης εμβέλειας του.
- Κάθε συστάδα περιέχει τουλάχιστον ένα σημείο πυρήνα.
- Σημεία που δεν είναι πυρήνα αλλα ούτε σημεία περιφέρειας εντάσσονται σε συστάδες και συνιστούν το "σύνορο" τους .
- Τα εξωτερικά σημεία δεν ανήκουν σε καμία συστάδα.

Στη βιβλιογραφία συναντώνται ποικίλες παραλλαγές του αλγόριθμου, όπως GDBSCAN, HDBSCAN αλλά και η ιεραρχική επέκταση του OPTICS.

5.6.3.2 Κώδικας αλγόριθμου

Η κλάση της βιβλιοθήκης scikit-learn που υλοποιεί τον αλγόριθμο σε python:
class sklearn.cluster.**DBSCAN**(*eps=0.5*, *min_samples=5*, *metric='euclidean'*, *metric_params=None*, *algorithm='auto'*, *leaf_size=30*, *p=None*, *n_jobs=1*)

5.6.4 Αλγόριθμος ολίσθησης μέσου (mean-shift)

5.6.4.1 Περιγραφή αλγόριθμου

Ο αλγόριθμος ολίσθησης μέσου είναι ένας αλγόριθμος κεντροειδών. Η διαδικασία ολίσθησης μέσου παρουσιάστηκε για πρώτη φορά το 1975 [62]. Έχει το μεγάλο πλεονέκτημα ότι δε χρειάζεται αρχικοποίηση του αριθμού των συστάδων αλλα τον προσδιορίζει μόνος του. Ο αλγόριθμος ξεκινά είτε με όλα τα σημεία πληροφορίας ως σημεία εκκίνησης (είτε με ένα πυκνό πλέγμα σημείων ανάμεσα τους) ενώ χρησιμοποιεί και μια συνάρτηση πυρήνα (kernel function), την ίδια για καθένα από αυτά. Η συναρτήσεις πυρήνα ποικίλλουν ως προς το είδος τους (π.χ γκαουσιανη, τριγωνικη κ.τ.λ) ενώ χαρακτηρίζονται και από ένα εύρος (bandwidth). Η πιο συνήθης συνάρτηση πυρήνα είναι η γκαουσιανη ενώ η τιμή εύρους τίθεται από τον αναλυτή. Αυτό είναι και το αντάλλαγμα της μη απαίτησης γνώσης του αριθμού συστάδων απο το μοντέλο. Η επιλογή υπέρμετρα μεγάλου εύρους θα οδηγήσει σε μία μόνο συστάδα. Αντίστοιχα, αν το εύρος είναι πολύ μικρό θα έχουμε πάρα πολλές συστάδες στο τέλος, πράγμα επίσης ανεπιθύμητο, οπότε χρειάζεται προσοχή και εμπειρία από τον αναλυτή. Ο ρόλος συνάρτησης πυρήνα και εύρους είναι να ορίσουν μια περιοχή-παράθυρο γύρω από κάθε επιλεχθέν σημείο μέσα στην οποία θα υπολογιστεί ένας σταθμισμένος μέσος των σημείων τα οποία περιέχει.

Βήματα αλγόριθμου:

Δεδομένων των παραπάνω ο αλγόριθμος ολίσθησης μέσου διαρθρώνεται ως εξής:

1. Θέσε όλα τα σημεία πληροφορίας ως σημεία εκκίνησης και αποθήκευσε το στιγμιότυπο.
2. Υπολόγισε για το κάθε σημείο την επόμενη του θέση βάσει της συνάρτησης πυρήνα.
3. Αν κάποια παράθυρα συμπίπτουν κράτα μόνο εκείνο που έχει τα πιο πολλα σημεία.
4. Επανάλαβε τα 2,3 μέχρις ότου ο επανυπολογισμός να μην αλλάζει τα αποτελέσματα αισθητά(κριτήριο σύγκλισης).
5. Τα σημεία που έχουν απομείνει αποτελούν τα κεντροειδή των τελικών συστάδων οι οποίες σχηματίζονται με ένα διάγραμμα Voronoi των κεντροειδών πάνω στο αρχικό στιγμιότυπο που αποθηκεύτηκε στο βήμα 1.

5.6.4.2 Κώδικας αλγόριθμου

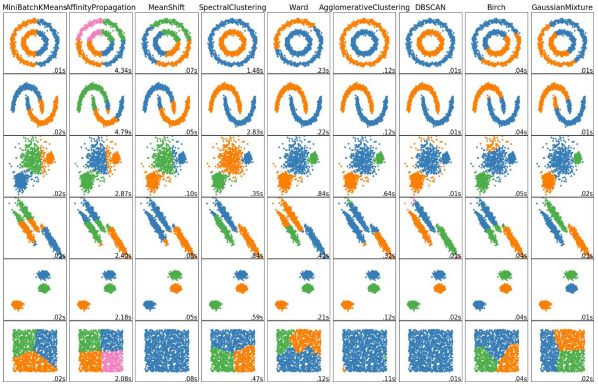
Η κλάση της βιβλιοθήκης scikit-learn που υλοποιεί τον αλγόριθμο σε python:
class sklearn.cluster.**MeanShift**(*bandwidth=None*, *seeds=None*, *bin_seeding=False*, *min_bin_freq=1*, *cluster_all=True*, *n_jobs=1*)

5.7 Εφαρμογές συσταδοποίησης

Η συσταδοποίηση έχει εφαρμογές σε κλάδους όπως η βιολογία, η ιατρική, οι κοινωνικές επιστήμες, το marketing, τα συστήματα διοίκησης αλλα και σε προβλήματα χωρικής ανάλυσης που είναι και η περίπτωση μας. Για παράδειγμα μια εταιρεία μπορεί να κατηγοριοποιεί τους πελάτες τις σε διαφορετικές συστάδες ανάλογα με τις καταναλωτικές τους συνήθειες ώστε να προσαρμόζει προϊόντα και διαφημίσεις πάνω τους. Σε επίπεδο χωρικής ανάλυσης, μπορεί μια εταιρεία παροχής cloud να συσταδοποιεί χωρικά και χρονικά τους πελάτες τις στον παγκόσμιο χάρτη έτσι ώστε να εγκαταστήσει τους server της σε στρατηγικά σημεία, να βελτιστοποιεί και να επιταχύνει τις παροχές της προς αυτους.

5.8 Τελικά συμπεράσματα

Η επιλογή αλγόριθμου συσταδοποίησης είναι πάντοτε συνάρτηση της φύσης του προβλήματος και αντικείμενο έρευνας, ενασχόλησης και εμπειρίας του αναλυτή. Κάθε αλγόριθμος έχει τα πλεονεκτήματα και τα μειονεκτήματα του. Οι παρακάτω εικόνες είναι χαρακτηριστικές ως προς τις αποδόσεις και τις περιπτώσεις χρήσης των αλγόριθμων συσταδοποίησης.



Εικόνα 5.17: Οι χρονικές επιδόσεις και τα αποτελέσματα συσταδοποίησης διαφορετικών αλγόριθμων που περιέχει η βιβλιοθήκη scikit-learn της python.[28]

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n_samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium `n_samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters, linkage type, distance	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large `n_samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes	Distances between nearest points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large `n_clusters` and `n_samples`	Large dataset, outlier removal, data reduction.	Euclidean distance between points

From scikit website

Πίνακας 5.1: Σύγκριση των αλγόριθμων συσταδοποίησης ως προς τις εφαρμογές για τις οποίες είναι κατάλληλοι.

[28] *Πηγή:* http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html

6. Μείωση διαστατικότητας

Το σύνολο μεθόδων Μείωσης διαστατικότητας ή Dimensionality reduction αποτελει μια υποκατηγορία του ML, η οποία εμπεριέχει τόσο supervised όσο και unsupervised μεθόδους. Σ' αυτό το σημείο είναι σημαντικό να αναφέρουμε την "κατάρα της διαστατικότητας" (Dimensionality curse). Γενικά και ανεξάρτητα από το μοντέλο του ταξινομητή, η απόδοση αυξάνεται όσο αυξάνεται το πλήθος και η ποιότητα των δεδομένων και όσο μειώνεται η διαστατικότητα. Αντίστροφα, τα προβλήματα δυσκολεύουν όσο η διαστατικότητα αυξάνεται και τα δείγματα δεν επαρκουν για να καλύψουν όλες τις κατηγορίες του προβλήματος. . Επίσης, όπως είδαμε σε προηγούμενες ενότητες, η πλειονότητα των αλγόριθμων λειτουργεί με την παραδοχή ότι οι ανεξάρτητες μεταβλητές είναι ανεξάρτητες μεταξύ τους. Ωστόσο, είδαμε ότι αυτό σπάνια συμβαίνει και συχνά προκύπτει το πρόβλημα του multicollinearity. Σκοπός λοιπόν των μεθόδων dimensionality reduction είναι η μείωση του πλήθους των εξαρτημένων μεταβλητών και άρα της διαστατικότητας του προβλήματος μας. Αυτό επιτυγχάνεται μέσω μιας διαδικασίας γνωστής και ως feature extraction ή εξαγωγή κύριων χαρακτηριστικών κατά την οποία γίνεται ταυτόχρονα σύνθεση και επιλογή των κυριότερων features ενός dataset. Οι τρόποι ποικίλλουν ανάλογα με τη μέθοδο που χρησιμοποιείται, όπως θα δούμε παρακάτω.

6.1 Principal Component Analysis (PCA)

6.1.1 Περιγραφή της μεθόδου

Η τεχνική PCA αποτελεί μια μη επιτηρούμενη τεχνική μείωσης διαστάσεων. Και μη επιτηρούμενη χαρακτηρίζεται για τον λόγο ότι δε λαμβάνει υπόψη της την ανεξάρτητη μεταβλητή. Ας υποθέσουμε, όπως γίνεται και στο [63], πως έχουμε ένα dataset το οποίο αποτελείται από κάποια instances ενός διανύσματος **x** p τυχαίων μεταβλητών X_i, οι οποίες σε όρους μηχανικής μάθησης αποτελούν τις ανεξάρτητες μεταβλητές του προβλήματος:

$$\boldsymbol{x} = \left(X_1, X_2, \ldots, X_p \right) \qquad \text{(Εξίσωση 6.1)}$$

Η τεχνική PCA αναζητά τα διανύσματα (principal components ή κύριες συνιστώσες) της μορφής

$$\boldsymbol{a}_i \boldsymbol{x} = a_{i1} X_1 + a_{i2} X_2 + \ldots a_{ip} X_p \qquad \text{(Εξίσωση 6.2)}$$

τα οποία να παρουσιάζουν 2 κύρια χαρακτηριστικά:

1. Να είναι όλα ασυσχέτιστα ή γραμμικώς ανεξάρτητα μεταξύ τους.
2. Το πρώτο εξ αυτών $\boldsymbol{\alpha}_1$ να παρουσιάζει τη μέγιστη διασπορά και όσο μεγαλώνει το i αυτή να παρουσιάζει φθίνουσα πορεία.

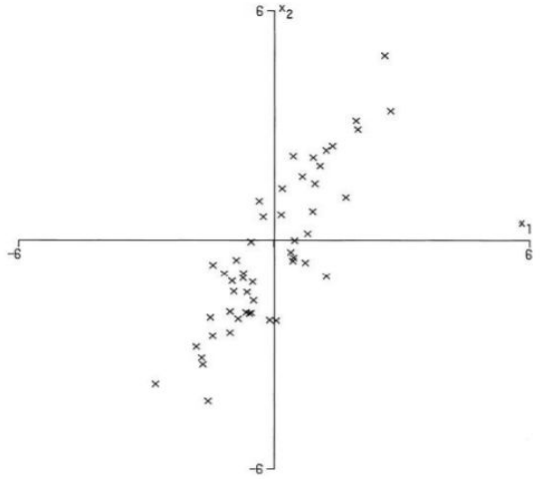
Συνοπτικά, η μέθοδος διαρθρώνεται ως εξης:

- Κανονικοποίηση των δεδομένων βάσει διακύμανσης της κάθε μεταβλητης (standardization)
- Κατασκευή του πίνακα συνδιακύμανσης ή συσχέτισης των δεδομένων
- Υπολογισμό ιδιοδιανυσμάτων και ιδιοτιμών.
- Επιλογή του επιθυμητού αριθμου ιδιοδιανυσμάτων (βάσεις του νέου υπόχωρου), ξεκινώντας από αυτά με τις μεγαλύτερες ιδιοτιμές.
- Προβολή του dataset στο νέο υπόχωρο που ορίζουν τα επιλεχθέντα ιδιοδιανύσματα.

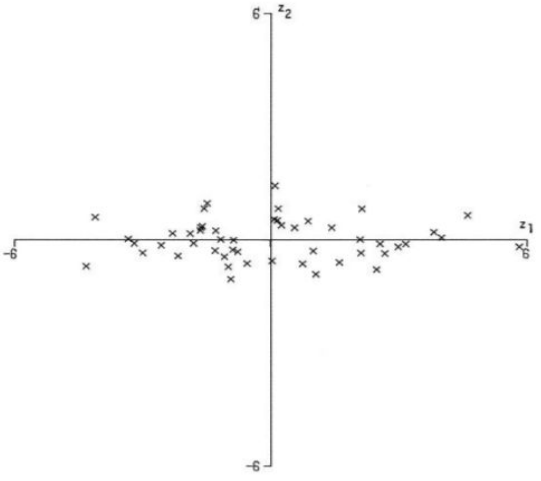
- ❖ Επιλογή με τη σειρά του αριθμό από principal components των οποίων η συσσωρευμένη διασπορά (cumulative explained variance) ξεπερνά ένα επιθυμητό κατώφλι ποσοστού επί της συνολικής.

Τελικά, η μέθοδος PCA προσπαθεί να "ανακαλύψει" στο dataset, νέες ανεξάρτητες μεταβλητές, οι οποίες αποτελούν γραμμικό συνδυασμό των προηγούμενων, και οι οποίες να "ερμηνεύουν" στο μέγιστο δυνατό βαθμό τη διασπορά που παρουσιάζει το αρχικό σύνολο ανεξάρτητων μεταβλητών. Με αυτό τον τρόπο καταπολεμούνται μια σειρά από προβλήματα τα οποία αντιμετωπίζονται κατά την επίλυση ενός προβλήματος μηχανικής μάθησης όπως είναι αυτό του multicollinearity, το οποίο έχει αναφερθεί πολλές φορές σε προηγούμενες ενότητες, και το μεγάλο πλήθος μεταβλητών, το οποίο αυξάνει την πολυπλοκότητα του προβλήματος και περιορίζει τη διαισθητική επαφή του αναλυτή με τα δεδομένα. Συχνά μάλιστα επιδιώκεται ο περιορισμός των ανεξάρτητων μεταβλητών σε δύο (εφόσον ερμηνεύουν ένα σεβαστό ποσοστό της διασποράς των δεδομένων) ώστε να υπάρχει δυνατότητα γραφικής αναπαράστασης των αποτελεσμάτων. Από την άλλη πλευρά είναι απαραίτητο να αναφέρουμε πως οι εναπομείνασες μεταβλητές (principal components) αποτελούν ένα υβρίδιο όλων των προηγούμενων οπότε συνήθως είναι δύσκολο να βρεθεί κάποια διαισθητική ερμηνεία γι' αυτές.

Χαρακτηριστικό είναι το παράδειγμα δύο έντονα συσχετισμένων μεταβλητών x_1,x_2 (εικόνα 6.1) οι οποίες παρουσιάζουν διασπορές $\sigma_1 < \sigma_2$. Στη συνέχεια, με μετασχηματισμό τους σε ένα χώρο (z_1,z_2) παρουσιάζουν τη συμπεριφορά του γραφήματος της εικόνας 6.2, όπου γίνεται εμφανές ότι πλέον είναι ασυσχέτιστες αλλά και ταυτόχρονα η μεταβλητή z_1 κουβαλάει πλέον το μεγαλύτερο ποσοστό της διασποράς του προηγούμενου χώρου.



Εικόνα 6.1: Τα δείγματα από δύο έντονα συσχετισμένες τυχαίες μεταβλητές.[29]



Εικόνα 6.2: Τα δείγματα από το μετασχηματισμό με PCA των δύο μεταβλητών της εικόνας 6.1.[21]

6.1.2 Επεκτάσεις της PCA

Στη βιβλιογραφία υπάρχουν μια σειρά από παραλλαγές της PCA όπως είναι η:

[29] *[Πηγη:[63]]*

- Sparse PCA: Η κλασική PCA συνήθως καταλήγει σε ένα χώρο του οποίου οι μεταβλητές, παρ' ότι λιγότερες, αποτελούν γραμμικούς συνδυασμούς όλων των αρχικών μεταβλητών. Η Sparse PCA αναζητά γραμμικούς συνδυασμούς που να μην περιέχουν καθόλου κάποιες από τις αρχικές μεταβλητές. Λεπτομέρειες μπορεί κανείς να αναζητήσει στο [64].
- L1-PCA: Ανήκει στην κατηγορία Robust τεχνικών. Εισάγει ένα περιορισμό επιπέδου 1 (L1), ωστε να είναι "ανθεκτική" σε outliers [65].
- Kernel PCA: Η τεχνική PCA απευθύνεται σε γραμμικά διαχωρίσιμα. Ωστόσο όπως είδαμε και στην περίπτωση του SVM η έννοια της γραμμικής διαχωρισιμότητας είναι κάπως σχετική, αφού με συνήθως υπάρχει η δυνατότητα να μετασχηματίσουμε τα δεδομένα μας σε κάποιο χώρο περισσότερων διαστάσεων στον οποίο και είναι διαχωρίσιμα και μάλιστα με χρήση του kernel trick αποφεύγουμε και τους υπολογισμους σε περισσότερες διαστάσεις. Με αυτόν ακριβώς τον τρόπο προκύπτει η μέθοδος Kernel PCA [66], η οποία λειτουργεί με αντίστοιχο τρόπο. Η kernel PCA χρησιμοποιείται ευρέως για την αποθορυβοποίηση δεδομένων όπως για παράδειγμα εικόνες και χρονοσειρές [67].

6.1.3 Εφαρμογές και υλοποιήσεις PCA

Η PCA συναντά ευρεία εφαρμογή σε πεδία της επιστήμης όπως οι νευροεπιστήμες, η πρόβλεψη χρηματιστηριακών μεγεθών [68] και σε μελέτη χρονοσειρών εν γένει, στον εντοπισμό outliers μέσα σε ένα σύνολο δεδομένων προς μελέτη [63]. Είναι πολύ σημαντικό να αναφέρουμε πως εφαρμόζεται σα βήμα προεπεξεργασίας δεδομένων κυρίως σε προβλήματα ταξινόμησης και όχι παλινδρόμησης διότι ενδέχεται να μας στερήσει χρήσιμες σχέσεις παλινδρόμηση μεταξύ κάποιου feature και της συνεχούς εξόδου.

Όσον αφορά στις υλοποιήσεις, Matlab, weka της java και python προσφέρουν αφθονία από εργαλεία εφαρμογής της μεθόδου και παραλλαγών της σε δεδομένα. Στη scikit-learn της python έχουμε κλάσεις για υλοποίηση τόσο της PCA όσο και παραλλαγών της όπως η Sparse PCA, η Incremental PCA (για τμηματική εφαρμογή σε μεγάλα datasets),η kernel PCA κ.α

6.2 Linear Discriminant Analysis (LDA)

6.2.1 Περιγραφή της μεθόδου

Η τεχνική αυτή απευθύνεται κατα κύριο λόγο σε προβλήματα ταξινόμησης. Η λογική η οποία ακολουθεί είναι πολύ κοντά σε αυτή της PCA. Η κύρια διαφορά είναι ότι η PCA εντοπίζει τους άξονες (principal components) με τη μεγαλύτερο ποσοστό διακύμανσης επί των δεδομένων εισόδου ενώ η LDA εντοπίζει τους άξονες (linear discriminants) που μεγιστοποιούν το διαχωρισμό μεταξύ των κλάσεων εξόδου, εξου και ο supervised χαρακτήρας της.

6.2.2 Παραδοχές της μεθόδου

Η τεχνική LDA βασίζεται σε μια σειρά από "δύσκαμπτες" παραδοχές όπως είναι η κανονική κατανομή των δεδομένων, η στατιστική ανεξαρτησία των χαρακτηριστικών (ανεξάρτητων μεταβλητών) και η ταυτοσημότητα των πινάκων συνδιακύμανσης των δεδομένων για κάθε κλάση της εξόδου. Παρ' όλα αυτά συχνά έχει καλές επιδόσεις ακόμα κι όταν αυτές καταστρατηγούνται σημαντικά όπως σε προβλήματα αναγνώρισης προσώπου και αντικειμένων [69] [70].

6.2.3 Η επέκταση GDA (General Discriminant Analysis)

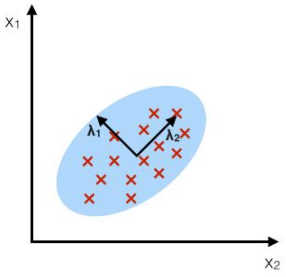
Αντίστοιχη μη γραμμική επέκταση της LDA αποτελεί η GDA (General Discriminant Analysis) η οποία, όπως στην περίπτωση των kernel PCA και kernel SVM, βασίζεται στην ιδέα της

απεικόνισης των δεδομένων σε ένα πολυδιάστατο χώρο περισσότερων διαστάσεων, στον οποίο να μπορεί να εφαρμοστεί η παραδοσιακή LDA μεσω του kernel trick.

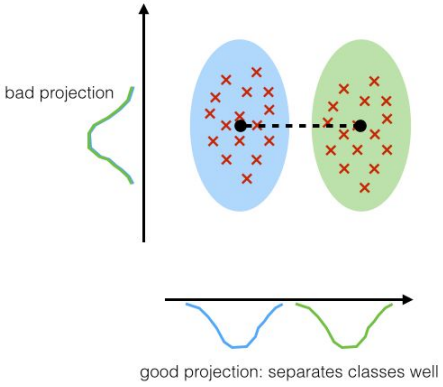
6.3 Συγκριση PCA και LDA

Συγκριτικά με την PCA, παρ' ότι η LDA φαντάζει πιο αποτελεσματική σε εργασίες ταξινόμησης, πειράματα, κυρίως σε αναγνώριση εικόνας, έχουν δείξει ότι η PCA τείνει να φέρνει καλύτερα αποτελέσματα σε περιπτώσεις όπου ο αριθμός δειγμάτων ανα κλάση είναι περιορισμένος [71]. Δεν είναι μάλιστα σπάνιο το φαινόμενο να γίνεται χρήση και των δύο μεθόδων συνδυαστικά (πρώτα PCA μετά LDA) σε κάποια προβλήματα μείωσης διαστατικότητας. Ένα μικρό πλεονέκτημα έναντι της PCA είναι η απουσία ανάγκης κανονικοποίησης (standardization) των δεδομένων.

Στην εικόνα 6.3 φαίνεται η λογική εφαρμογής της PCA πάνω σ' ένα σύνολο δεδομένων. Εντοπίζονται τα principal components (νέες βάσεις) που εξηγούν το μεγαλύτερο ποσοστό της διακύμανσης. Αν θέλαμε να πέσουμε σε μία διάσταση θα επιλέγαμε τη συνιστώσα λ_2. Στην εικόνα 6.4 φαίνεται η λογική εφαρμογής της LDA πάνω σ΄ ένα σύνολο δεδομένων δύο κλάσεων. Προσπαθούμε να κάνουμε μια προβολή των δεδομένων σε νέες βάσεις έτσι ώστε να διαχωρίζονται με τον καλύτερο τρόπο οι δύο κλάσεις (στην περίπτωση αυτή η νέα βάση είναι η οριζόντια συνιστώσα και είναι αρκετή για να διαχωρίσει πλήρως τα δεδομένα) .



Εικόνα 6.3: Η λογική εφαρμογής της PCA.[30]



Εικόνα 6.4: Η λογική εφαρμογής της LDA.

[30] *[Πηγή: https://sebastianraschka.com/Articles/2014_python_lda.html#normality-assumptions]*

7. Μια εφαρμογή πρόβλεψης για μεγάλα γεγονότα στα πλαίσια των Smart Cities, με χρήση τεχνικών μηχανικής μάθησης σε συστήματα Fog Computing.

7.1 Large Events - Γεγονότα μεγάλης κλίμακας

7.1.1 Η έννοια των Large Events

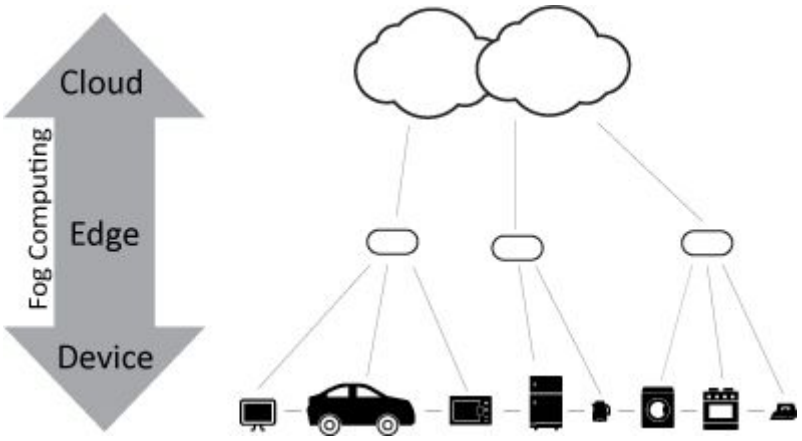
Τα Large Events αποτελούν ένα βασικό αντικείμενο μελέτης στον κλάδο των Smart Cities [72] [X] και έχουν μια σειρά από χαρακτηριστικά τα οποία σχετίζονται με έννοιες όπως το Internet of Things (Iot), το Fog Computing (το οποίο εξετάζεται στην επόμενη ενότητα) και η Μηχανική Μάθηση στην οποία έχουμε αναφερθεί εκτενώς σε προηγούμενες ενότητες. Η έννοια του Large Event παραπέμπει στη μαζική συγκέντρωση χιλιάδων ανθρώπων σε μια περιοχή με σκοπό να παρακολουθήσουν κάποιο γεγονός όπως είναι για παράδειγμα μία συναυλία, παρέλαση, αθλητικό συμβάν ή μία ομιλία γνωστού προσώπου.

Ο χώρος μέσα σε ένα Large Event, είναι λογικό να αποτελείται από διαφορετικά σημεία ενδιαφέροντος (PoIs) τα οποία αντίστοιχα ορίζουν σε μια μικρή περιοχή γύρω τους περιοχές ενδιαφέροντος (AoIs). Ως AoI ενός PoI ορίζεται η περιοχή γύρω από το PoI στην οποία όταν βρεθεί ένας επισκέπτης αυτόματα γίνεται η θεώρηση πως ο επισκέπτης αυτός εξυπηρετείται η διασκεδάζει στο PoI αυτό. Για παράδειγμα, ένα stand που πουλάει burger μπορεί να είναι ένα σημείο (PoI) σε ένα χάρτη ενός Large Event, ωστόσο όλη η ουρά που δημιουργείται για αγορά burgers απασχολεί μία επιφάνεια στο χώρο (AoI) και δεν είναι σημειακη. Όλοι οι άνθρωποι που βρίσκονται στο εσωτερικό της πρέπει να θεωρηθούν πελάτες του PoI πώλησης burger.

Οι διοργανωτές των Large Events, με τη σειρά τους, αναπτύσσουν ποικίλες ανάγκες οι οποίες σχετίζονται με την ασφαλή και ομαλή διεξαγωγή του γεγονότος. Τέτοιες ανάγκες είναι η καλύτερη δυνατή προετοιμασία για καταστάσεις εκτάκτου κινδύνου, η βέλτιστη συνεργασία και προσαρμογή του προσωπικού του event στις ανάγκες των επισκεπτών για την καλύτερη εξυπηρέτηση τους. Επίσης, ένας καθοριστικός παράγοντας ικανοποίησης των επισκεπτών είναι και η καλή λειτουργία των όποιων αυτοματοποιημένων η ηλεκτρονικών υπηρεσιών παρέχει το γεγονός όπως είναι οι αυτόματοι πωλητές, το wifi, οι ηλεκτρονικές εφαρμογές εξυπηρέτησης για κινητά κ.α

7.1.2 Fog Computing σε Large Events

Μιλώντας για Large Events στα πλαίσια των Smart Cities είναι δεδομένο ότι υπάρχει ένα καλοστημένο background από από ηλεκτρονικές υπηρεσίες και εφαρμογές για την καλύτερη διεξαγωγή και την επιτήρηση του γεγονότος. Οι υπηρεσίες αυτές, υποστηρίζονται με επεξεργαστική ισχύ η οποία παρέχεται είτε από τοπικό hardware κατανεμημένο σε πολλαπλές συσκευές του χώρου (Edge Computing) είτε από το cloud (Cloud Computing) [73]. Γίνεται αντιληπτό ότι είναι επιθυμητό ο φόρτος εργασίας κατανέμεται κατά προτεραιότητα στο Edge τόσο για λόγους κόστους και ταχύτητας (low latency) [74], αλλά και ιδιωτικότητας δεδομένων και ασφάλειας. Ωστόσο σε ώρες αιχμής είναι λογικό το Edge να αδυνατεί να ανταποκριθεί στη ζήτηση οπότε και επιβάλλεται η ανάθεση του επιπλέον φορτίου σε υπηρεσίες Cloud [75] [76]. Η διαδικασία αυτή της εξισορρόπησης υπολογιστικού φόρτου ανάμεσα σε Cloud και Edge παραπέμπει στην έννοια του Fog Computing [77]. Στην εικόνα 7.1 βλέπουμε μια αναπαράσταση του ρόλου του Fog Computing στα πλαίσια ενός Smart City και, κατά προέκταση, ενός Large Event.



Εικόνα 7.1: Το Fog Computing στα πλαίσια των Smart Cities.[31]

7.1.3 Η ανάγκη για πρόβλεψη κατανομής των επισκεπτών σε Large Events

Από τις προηγούμενες ενότητες γίνεται εμφανής η αναγκη για συνεχή γνώση και για πρόβλεψη του πλήθους και της τοποθεσίας στις AoI των επισκεπτών ενός Large Event στα εξής σημεία:

- Οι υπεύθυνοι ασφαλείας μπορούν να είναι κατάλληλα προετοιμασμένοι για οποιοδήποτε έκτακτο γεγονός προκύψει προσαρμόζοντας τα μέτρα στις παρούσες αλλά και μελλοντικές συγκεντρώσεις επισκεπτων στα διάφορα σημεία του χώρου.
- Τα σημεία πωλήσεων μπορούν να ενισχύσουν το προσωπικό και την εξυπηρέτηση τους όταν υπάρχει πρόβλεψη αύξησης των πελατών τους σε κάποια χρονική περίοδο εξυπηρετώντας τους έτσι καλύτερα και ταυτόχρονα αυξάνοντας τα κέρδη τους.
- Η όποια αντίστοιχη εφαρμογή recommendations μπορεί να κάνει τις προτάσεις τις όλο και καλύτερες δεδομένης της πρόβλεψης κατανομής ανθρώπων στα διάφορα σημεία ενδιαφέροντος του χώρου.
- Fog Computing: Είναι δεδομένο ότι ο χώρος εξυπηρετείται από ένα σύνολο συσκευών IoT οι οποίες είναι διασκορπισμένες στο χώρο και καθεμία έχει πεπερασμένη επεξεργαστική ισχύ και εμβέλεια εξυπηρέτησης. Επιβάλλεται, λοιπόν, οι μεταβάσεις από το Edge στο Cloud, και αντίστροφα, να μπορούν να γίνουν όσο το δυνατόν πιο αποτελεσματικά, γρήγορα, αξιόπιστα και οικονομικά η μετάβαση από το Edge στο Cloud, χωρίς αφενός κενά στην εξυπηρέτηση και αφετέρου περιττές χρεώσεις και άλογη έκθεση ιδιωτικών δεδομένων μέσω της απασχόλησης Cloud πόρων.

7.1.4 Το μουσικό φεστιβάλ DasFest.

Στη δική μας περίπτωση, το Large event που εξετάζουμε είναι το DasFest, ένα festival το οποίο πραγματοποιείται ετησίως, κάθε Ιούλιο, στην περιοχή Karlsruhe, νοτιοδυτικά της Γερμανίας. Πρόκειται για ένα κατα κόρον μουσικό φεστιβάλ το οποίο ωστόσο φιλοξενεί και δραστηριότητες ψυχαγωγίας άθλησης και παιδικής απασχόλησης. Το φεστιβάλ φιλοξενεί κατά μέσο όρο 300.000 άτομα κατα τη διάρκεια και των τριών ημερών, ενώ στις ώρες αιχμής μπορεί κανείς να συναντήσει στιγμιαία συγκέντρωση ανθρώπων της τάξης των 150.000 ατόμων. Στην εικόνα 7.2 βλέπουμε ένα χάρτη του DasFest όπως παρέχεται από τους διοργανωτές.

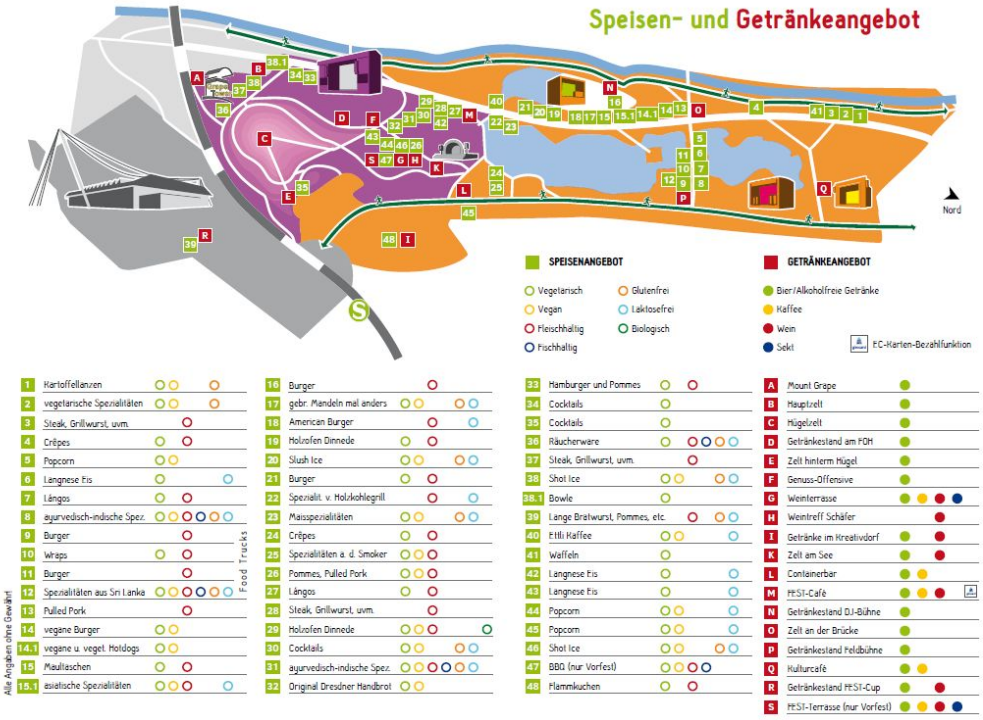
[31] *[Πηγή: https://www.openfogconsortium.org/a-plain-language-post-about-fog-computing-that-anyone-can-understand/]*

7.1.4.1 Η εφαρμογή κινητών συσκευών του DasFest

Οι διοργανωτές παρέχουν στους επισκέπτες του φεστιβάλ μια εφαρμογή[32] για Android και iOS για την καλύτερη εξυπηρέτηση τους. Η εφαρμογή έχει χρήση παροχής πληροφοριών σε σχέση με δρώμενα του φεστιβάλ, χάρτες, πάρκινγκ, εξόδους κινδύνου. Επιπλέον, γίνεται σχεδιασμός για το μέλλον ώστε να παρέχει recommendations σε σχέση με την καλύτερη εξυπηρέτηση των αναγκών των επισκεπτών. Για παράδειγμα αν κάποιος χρήστης επιλέξει ότι θέλει να χρησιμοποιήσει ένα WC η εφαρμογή θα τον παραπέμψει σε κάποιο WC με κριτήριο τόσο την απόσταση του από αυτό όσο και με την ουρά που υπάρχει εκεί. Όπως είναι αναμενόμενο, κάθε φορά που ένας χρήστης ξεκινάει ένα session της εφαρμογής, αυτή ξεκινάει να συλλέγει χωρικά δεδομένα τοποθεσίας του χρήστη ώστε να μπορεί να τον εξυπηρετήσει κατάλληλα.

7.1.4.2 Οι ανάγκες των διοργανωτών του DasFest

Η αλληλεπίδραση με τους διοργανωτές του φεστιβάλ, δηλαδή με τους υπεύθυνους για ασφάλεια, με τους υπεύθυνους προσωπικού αλλά και με την ομάδα πληροφοριακών συστημάτων έκανε φανερή την ανάγκη τους για πρόβλεψη σε κάθε χρονική στιγμή της κατανομής όπως σε κάθε Large Event για λόγους παρόμοιους με τους παραπάνω.



Εικόνα 7.2: Ο χάρτης του φεστιβάλ DasFest για το έτος 2017

7.2 Το πρόβλημα πρόβλεψης κατανομής χρηστών με μεθόδους μηχανικής μάθησης.

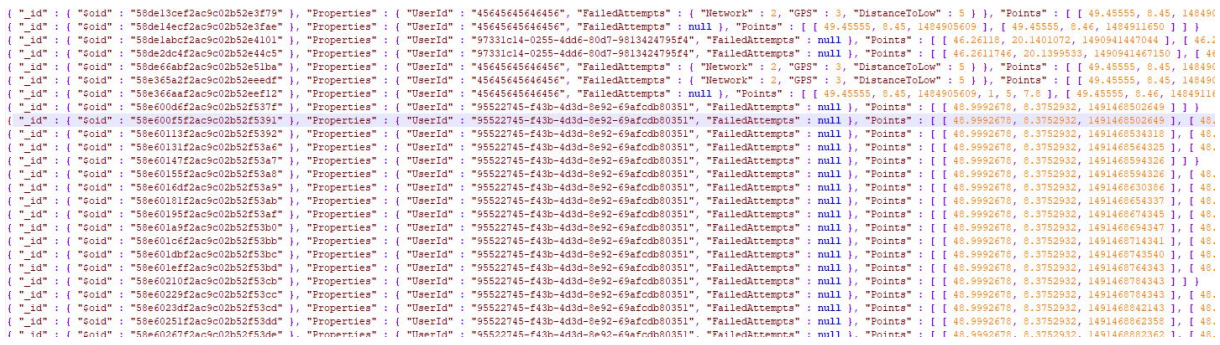
Στο κεφάλαιο αυτό περιγράφουμε τη διαδικασία πρόβλεψης της θέσης των χρηστών στο DasFest για τη χρονιά 2017 κάνοντας χρήση τόσο τεχνικών τόσο ταξινόμησης, όσο και παλινδρόμησης.

[32] *http://www.dasfest.de/index.php?article_id=249&clang=0*

7.2.1 Η συλλογή και αρχική δομή των δεδομένων του προβλήματος πρόβλεψης.

Στη διπλωματική αυτή εργασία κληθήκαμε να εργαστούμε πάνω σε ένα raw dataset το οποίο αφορά στο Basmati Project στα πλαίσια του Horizon2020 Framework Program. Το dataset αυτό αφορά σε δεδομένα τοποθεσίας που συλλέχτηκαν τις χρονιές 2017, 2018 κατά την εξυπηρέτηση των επισκεπτών του Dasfest από την εφαρμογή του φεστιβάλ. Χρησιμοποιείται ο όρος "raw" εφόσον το dataset βρίσκεται σε μια πολύ πρώιμη μορφή η οποία, από μόνη της και χωρίς ένα σημαντικό κομμάτι ανάλυσης και προεπεξεργασίας δεδομένων, δε δύναται να μας δώσει ούτε χρήσιμες πληροφορίες ούτε να ορίσει ένα συγκεκριμένο πρόβλημα προς επίλυση. Η αρχική μορφή του dataset που παραλάβαμε έχει την εξής δομή σε .json τύπου αρχείο:

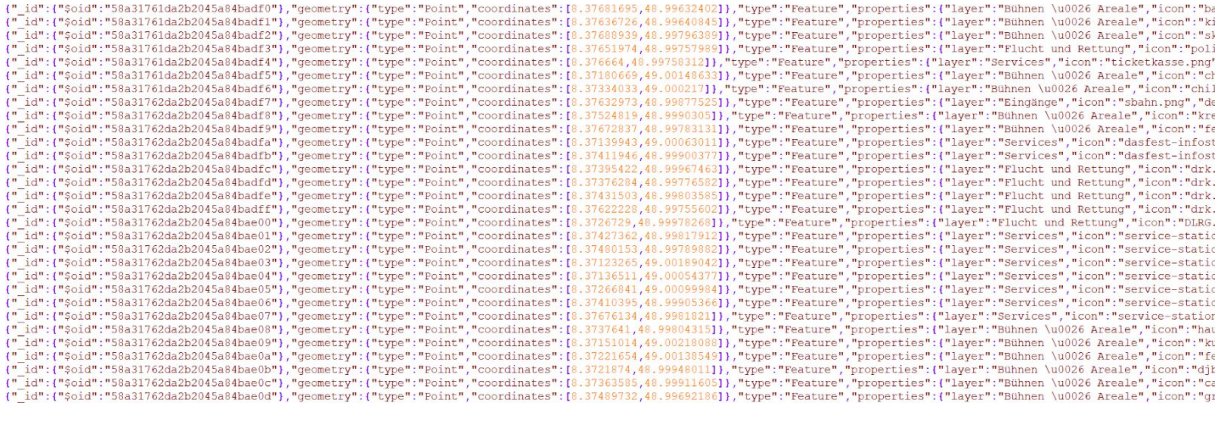
α) Αρχείο TESTING_Basmati-Track_2018.json



Στο παραπάνω dataset μπορεί κανείς να διακρίνει ένα σύνολο από καταχωρήσεις (μία ανα γραμμή) σε μορφή λεξικού - dictionary. Κάθε καταχώρηση αφορά ένα session γι' αυτό έχει και μοναδικό $oid δηλαδή object-id, το οποίο μπορεί να λειτουργήσει και σαν primary key στο σύνολο δεδομένων αυτό. Στη συνέχεια περνάμε στην κατηγορία Properties η οποία εμπεριέχει υποκατηγορίες οι οποίες αφορούν το ID του χρήστη δηλαδή της συσκευής που συνδέεται, και την αποτυχία σύνδεσης, η οποία, αν είναι αληθής, συνοδεύεται και από μια σειρά από σχετικές με την αποτυχία πληροφορίες. Τρίτη και βασικότερη για εμάς κατηγορία πληροφοριών του λεξικού είναι η Points, η οποία περιέχει μια σειρά από διανύσματα τριών στοιχείων τα οποία περιλαμβάνουν τις χωρικές συντεταγμένες και ένα χρονικό Timestamp στη γνωστή μορφή τύπου epoch (Latitude, Longitude, Timestamp) ή (y,x,t) κάθε στίγματος GPS που αντλεί η εφαρμογή από τη συσκευή του χρήστη.

Εκτός του παραπάνω αρχείου που αποτελεί το κύριο, προς επεξεργασία, dataset μας δόθηκαν επιπλέον τα εξής δύο αρχεία της μορφής .json:

β) Αρχείο festinfrastructure.json



Πρόκειται για ένα αρχείο στο οποίο ορίζονται όλες οι πληροφορίες για κάθε σημείο ενδιαφέροντος εντός του χώρου του φεστιβάλ (στα αγγλικά Points of Interest και από εδώ και στο εξής PoI). Ένα PoI μπορεί να είναι από μια σκηνή εώς ένα stand το οποίο πουλάει vegan φαγητό ή ακόμη και μια τουαλέτα. Τέτοιες πληροφορίες είναι το όνομα του PoI, οι χωρικές συντεταγμένες (γεωγραφικό υψος, γεωγραφικό μήκος), το είδος του, η κατηγορία στην οποία ανήκει ανάλογα με τις ανάγκες που εξυπηρετεί, το είδος του εικονιδίου με το οποίο αναπαρίσταται στο χάρτη του φεστιβάλ κ.α. Όλα αυτά τα δεδομένα, προφανώς μας δίνουν μια εικόνα της δομής του φεστιβάλ όπως θα δούμε στη συνέχεια.

γ) Αρχείο geofences.json


```
{  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "Name": "das Fest"
      },
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [
              8.369522094726562,
              48.99595939341933
            ],
            [
              8.376517295837402,
              48.99595939341933
            ],
            [
              8.376517295837402,
              49.001140064087146
            ],
            [
              8.369522094726562,
              49.001140064087146
            ],
            [
              8.369522094726562,
              48.99595939341933
            ]
          ]
        ]
      }
    }
```


Το αρχείο αυτό περιέχει μια βασική πληροφορία η οποία είναι οι χωρικές συντεταγμένες ενός πολυγώνου το οποίο ορίζει γεωγραφικά το χώρο πραγματοποίησης του φεστιβάλ. Το πολύγωνο αυτό ορίζεται από 4 σημεία συντεταγμένων τα οποία στην πραγματικότητα ορίζουν ένα κλειστό παραλληλόγραμμο όπως μπορεί να παρατηρήσει κανείς από τις επιμέρους συντεταγμένες των σημείων.

7.2.2 Συνοπτική περιγραφή του προβλήματος

Δεδομένων των παραπάνω, στόχος μας είναι η ανάλυση μελέτη και η πρόβλεψη της κίνησης και της κατανομής των επισκεπτών του φεστιβάλ στο δοθέντα χώρο κατα τη διάρκεια των ημερών διεξαγωγής του φεστιβάλ δηλαδη 21/7/2017-23/7/2017 και 20/7/2018 και 22/7/2018. Οι προοπτικές προσέγγισης του προβλήματος ποικίλλουν ως προς διάφορες παραμέτρους τις οποίες θα εξετάσουμε αναλυτικά σε επόμενες ενότητες.

Μακροσκοπικά, λοιπόν, ο γενικός σκοπός της εργασίας αυτής είναι η κατασκευή και μελέτη ενός συνόλου προβλημάτων επιβλεπόμενης μάθησης με τα εξής χαρακτηριστικά:

Είσοδοι (input variables) των μοντέλων επιβλεπόμενης μάθησης:

- Η κατανομή των επισκεπτών σε επιλεγμένες περιοχές ενδιαφέροντος κατα τη διάρκεια μιας η περισσότερων, προεπιλεγμένης διάρκειας, χρονικών περιόδων. Ως περιοχή ενδιαφέροντος (Area of Interest και από εδώ και στο εξής AoI) ενός PoI ορίζουμε κάθε περιοχή του χώρου που σχηματίζεται γύρω από ένα PoI και περιέχει όλα τα σημεία του χώρου που βρίσκονται κοντύτερα σε αυτό το PoI σε σχέση με τα υπόλοιπα.

- Ένα σύνολο επιπλέον χαρακτηριστικών εισόδου τα οποία προκύπτουν από διαδικασίες εξόρυξης γνώσης (data mining) και πιο συγκεκριμένα και τα οποία θα δοθούν σαν επιπλέον είσοδοι στο μοντέλο επιβλεπόμενης μάθησης. Στην περίπτωση μας τέτοια χαρακτηριστικά υπήρξαν οι δείκτες χρονικών περιόδων, ο καιρός αλλα και κάποιοι δείκτες "δημοτικότητας" των παρόντων καλλιτεχνών στο πρόγραμμα του φεστιβάλ.

Έξοδοι (dependent variables) των μοντέλων επιβλεπόμενης μάθησης:

- Η πρόβλεψη της κατανομής των επισκεπτών στις AoI σε επόμενη χρονική στιγμή από αυτές της εισόδου.

Παρακάτω στον πίνακα 7.1 βλέπουμε ένα ενδεικτικό προσχέδιο της μορφης του τελικού dataset προς τροφοδοσία στους αλγόριθμους ML, το οποίο ωστόσο επιδέχεται πολλών τροποποιήσεων ανάλογα με το είδος της μελέτης, την προσέγγιση του προβλήματος που πραγματοποιούμε και το είδος του αλγόριθμου που χρησιμοποιούμε. Τα γράμματα A,B,C,D,E,F αντιπροσωπεύουν 6 επιλεχθείσες AoIs.

Input							Output					
A(t)	B(t)	C(t)	D(t)	E(t)	F(t)	Extra Features (t)	A(t+1)	B(t+1)	C(t+1)	D(t+1)	E(t+1)	F(t+1)

Πίνακας 7.1: Μια υπεραπλουστευμένη αναπαράσταση του dataset του προβλήματος επιτηρούμενης μάθησης.

7.2.3 Εργαλεία

Κατά την εκπόνηση της διπλωματικής εργασίας και την επίλυση του προβλήματος, η οποία παρουσιάζεται στις επόμενες ενότητες, έγινε χρήση python 3.6 σε περιβάλλον εργασίας Spyder, και έγινε ευρεία χρήση των βιβλιοθηκών:

- Pandas [2] για σκοπούς αναπαράστασης και πρώιμης επεξεργασίας σε μορφή πινάκων.
- Numpy [3] για αριθμητικές πράξεις δεδομένων πινάκων
- Matplotlib [78] για γραφικές παραστάσεις
- Scikit-learn [4] για την προεπεξεργασία, τους αλγόριθμους ML και τις μετρικές αξιολόγησης
- Keras [79] για τα νευρωνικά δίκτυα, η οποία τρέχει σε tensorflow [80] backend.

Επιπλέον, τα αρχεία και τα scripts, στα οποία γίνεται αναφορά, βρίσκονται σε repository[33] του github.

7.2.4. Προεπεξεργασία δεδομένων

Η προεπεξεργασία των δεδομένων του προβλήματος αποτελεί ίσως το πιο σημαντικό στάδιο της επίλυσης του. Τα δεδομένα, έτσι όπως τα είδαμε στα αρχεία της προηγούμενης ενότητας, πρέπει αρχικά να ερμηνευθούν και να αποκτήσουν μια φυσική υπόσταση. Επιβάλλεται επιπλέον, να αναπαρασταθούν σε εύχρηστες δομές. Στην περίπτωση μας έγινε χρήση DataFrames και γενικότερα των εργαλείων της βιβλιοθήκης pandas. Κατ' αυτό τον τρόπο μπορούμε να τα παρατηρήσουμε, να τα αναπαραστήσουμε γραφικά και να δομήσουμε, κατ' επέκταση, ένα πρόβλημα επιτηρούμενης μάθησης με νόημα και πρακτική χρησιμότητα.

7.2.4.1 Φιλτράρισμα δεδομένων και δομή σε DataFrame

Πρώτα απ' όλα, ήταν απαραίτητο τα δεδομένα να φιλτραριστούν στο χώρο και στο χρόνο, εφόσον υπήρχαν πολλά δοκιμαστικά sessions της εφαρμογής εκτός του χωρικού πολυγώνου και των ημερών πραγματοποίησης του φεστιβάλ. Επιπλέον, τα δεδομένα καθαρίστηκαν από διπλότυπα δηλαδή entries με κοινό ID χρήστη και Timestamp. Τα παραπάνω πραγματοποιούνται στο script datapreprocessingTime.py και jsonFences_an.py του κώδικα μας. Στο script

[33] *https://github.com/pelekhs/Visitor_distribution_prediction*

datapreprocessingTime.py πραγματοποιείται το χρόνικό φιλτράρισμα των δεδομένων, δηλαδή απορρίπτονται όλα τα entries τα οποία είναι εκτός των ημερών διεξαγωγής του φεστιβάλ. Στο αρχείο jsonFences_an.py πραγματιοποιείται η σύνθεση ενός συνόλου μεθόδων, οι οποίες περιέχονται στο αρχείο defs.py. Οι μέθοδοι αυτές χρησιμεύουν στο φιλτράρισμα επιλογής του χρήστη, στο χωρικό φιλτράρισμα, στην επιστροφή των διανυσμάτων θέσεων των χρηστών και των σημείων ενδιαφέροντος. Ως αποτέλεσμα παίρνουμε ένα DataFrame της μορφής του πίνακα 7.2. Αξίζει να αναφερθούν τα εξής:

1. Τα δύο αρχικά datasets των 2 ετών περιέχουν αριθμό entries τάξης μεγέθους μερικών εκατοντάδων χιλιάδων
2. Μετά το φιλτράρισμα καταλήγουμε σε ένα DataFrame 45602 στοιχείων εκ των οποίων τα 29966 (65%) αφορούν στο 2017.
3. Στο DataFrame εντοπίζονται μόνο 817 μοναδικοί χρήστες εκ των οποίων 603 (73%) ανήκουν στο 2017. Για το 2017, αυτό σημαίνει πως, πρακτικά, μελετάμε τις θέσεις 603 χρηστών της εφαρμογής οι οποία έχουν δώσει μέσα σε 3 ημέρες 29966 χρήσιμα στίγματα κατά τη χρήση της εφαρμογής.

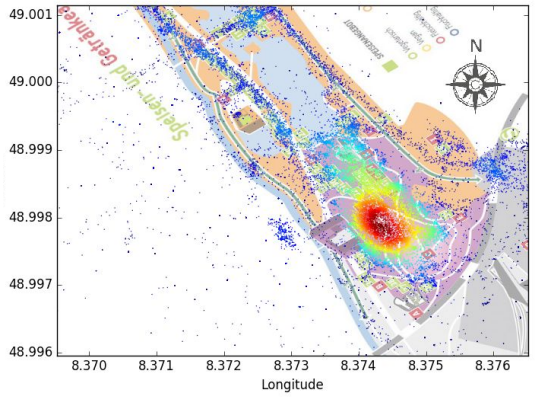
Τα παραπάνω αντανακλούν σε μεγάλο βαθμό το πόσο μη αντιπροσωπευτικά του συνόλου είναι τα δεδομένα που έχουμε στην κατοχή μας.

	$oid	ID	Epoch	Timestamp	Date	Time	X	Y
0	597188dac1acdf13a82dd456	9e3a954d-550a-423c-b6da-42bfd61a7bbb	1500612618	2017-07-21 04:50:18	2017-07-21	04:50:18	8.3716083	48.9980785
1	597248c2c1acdf13a82debfc	9e3a954d-550a-423c-b6da-42bfd61a7bbb	1500612618	2017-07-21 04:50:18	2017-07-21	04:50:55	8.3716083	48.9980785

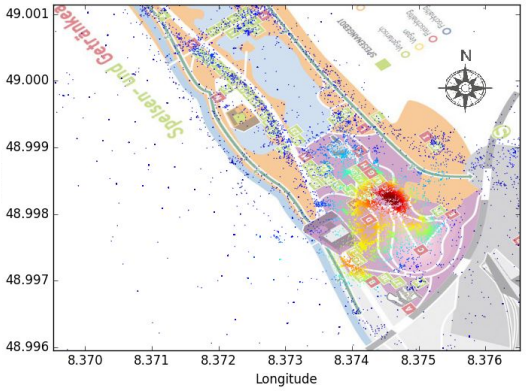
Πίνακας 7.2: Η απεικόνιση των δεδομένων σε δομή pandas DataFrame των δεδομένων όπως προκύπτει από το φιλτράρισμα.

7.2.4.2 Παρατήρηση των θέσεων επισκεπτών και των σημείων ενδιαφέροντος.

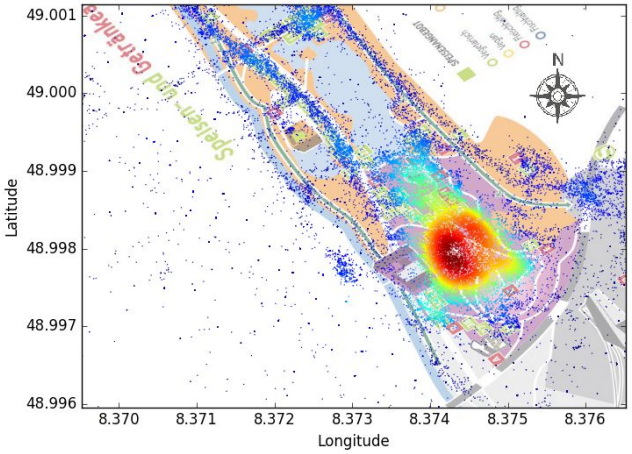
Μια πολύ χρήσιμη αντίληψη του χώρου, και της συγκέντρωσης του κόσμου στο φεστιβάλ, και για τις δύο χρονιές 2017, 2018 μας προσέφερε η οπτικοποίηση του χάρτη συνδυαστικά με τον κόσμο που παρήλθε από το φεστιβάλ τις τρεις μέρες πραγματοποίησης του η οποία υλοποιείται στο script gaussian_density_plot_script.py. Θερμότερα χρώματα υποδηλώνουν μεγαλύτερη πυκνότητα σημείων, δηλαδή μεγαλύτερη συγκέντρωση ανθρώπων στην αντίστοιχη περιοχή του χάρτη (εικόνες 7.3, 7.4, 7.5). Ήδη παρατηρούμε, όπως αναφέρθηκε νωρίτερα, την πολύ μικρή παρουσία στιγμάτων στο φεστιβάλ το 2018.



Εικόνα 7.3: Ένα συσσωρευτικό heatmap των επισκεπτών για τις 3 ημέρες του 2017

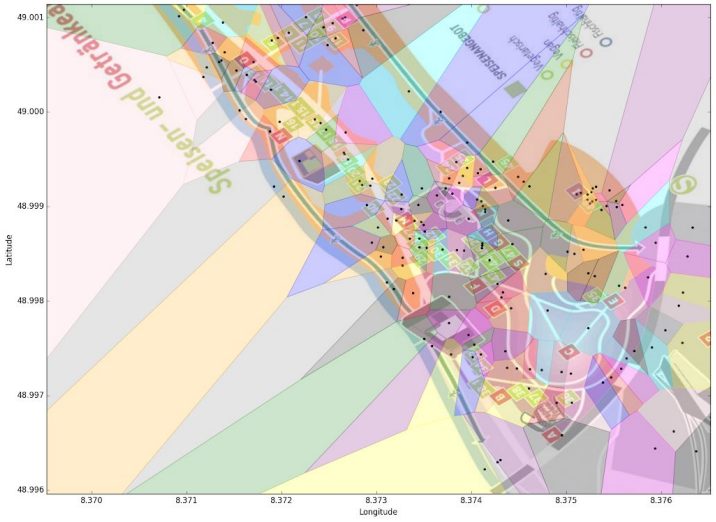


Εικόνα 7.4: Ένα συσσωρευτικό heatmap των επισκεπτών για τις 3 ημέρες του 2017



Εικόνα 7.5: Ένα αθροιστικό συσσωρευτικό heatmap των επισκεπτών και για τις 2 χρονιές

Σε επόμενο στάδιο, πραγματοποιείται μια οπτικοποίηση των σημείων ενδιαφέροντος του φεστιβάλ όπως αυτά παρουσιάζονται στο αρχείο B) αφού προηγουμένως και αυτά αναπαραστάθηκαν με μορφή συντεταγμένων σε δομή δεδομένων pandas DataFrame. Μαζί με τα PoIs παρέχεται και μια θεωρητική παρουσίαση των αντίστοιχων τους AoIs με χρήση του γνωστού διαγράμματος Voronoi (εικόνα 7.6) της οποίας η υλοποίηση βρίσκεται στο αρχείο voronoi_plot_script.py. Στην πραγματικότητα πρόκειται για μία απλοική προσέγγιση όπου κάθε PoI αντιστοιχίζεται στα σημεία του χώρου τα οποία απέχουν τη μικρότερη απόσταση από αυτό σε σχέση με οποιοδήποτε άλλο PoI. Από αυτό το σημείο και έπειτα γίνεται εμφανής η αδυναμία μας να κάνουμε προβλέψεις για κάθε PoI το οποίο μας παρέχεται από το αντίστοιχο αρχείο του Basmati δεδομένης της υπέρογκης πυκνότητας PoIs σε σχέση με τα στίγματα που έχουμε συλλέξει ανά χρονική στιγμή.



Εικόνα 7.6: Διάγραμμα Voronoi των PoIs και AoIs

7.2.4.3 Παραδοχές κατά την προεπεξεργασία δεδομένων

Η φύση της συλλογής των δεδομένων αλλά και τα αποτελέσματα του προεπεξεργαστικου σταδίου μας οδήγησαν στις εξής παραδοχές κατα την εκπόνηση του τμήματος επιτηρούμενης μάθησης της εργασίας:

1. Ο κόσμος που παρευρίσκεται στο festival κάθε χρονική στιγμή είναι ευθέως ανάλογος του κόσμου που δίνει στίγμα μέσω χρήσης της εφαρμογής.
2. Ο συνολικός πραγματικός αριθμός ανθρώπων εντός του festival (ειδικά σε peak hours) είναι δυνατόν να προσεγγιστεί με καλή ακρίβεια από την ποσότητα αγορασμένων εισιτηρίων για την ημέρα και από το σκανάρισμα τους στην είσοδο.
3. Προκειμένου να μοντελοποιήσουμε την έννοια της χρονικής στιγμής επιβάλλεται να θεωρήσουμε χρονικά διαστήματα κάποιων λεπτών ως μια σταθερή χρονική περίοδο μέσα στην οποία όλα τα στίγματα που παρατηρήθηκαν θα ομαδοποιούνται με κοινό Timestamp. Η περίοδος αυτή έχει νόημα να είναι αρκετά μεγάλη ώστε να περιέχει αρκετές παρατηρήσεις-στίγματα δημιουργώντας έτσι μια αξιόπιστη κατανομή ανθρώπων στο χώρο. Ταυτόχρονα η περίοδος αυτή πρέπει να είναι αρκετά μικρή ώστε:
    a. Το στιγμιότυπο που θεωρούμε να είναι σχετικά στατικό χωρίς πολλές χαμένες (μη καταγεγραμμένες) μεταβολές στην κατανομή των χρηστών στο χώρο.
    b. Να έχει νόημα η διαδικασία πρόβλεψης. Για παράδειγμα μια χρονική περίοδος (timestep) διάρκειας μιας ώρας είναι πολύ μακροπρόθεσμη για να ενδιαφέρουν το διοργανωτή τα αποτελέσματα της πρόβλεψης τόσο σε επίπεδο ασφαλείας όσο και σε επίπεδο edge computing. Σε αντίθεση σε επίπεδο λίγων λεπτών θα ήταν πολύ χρήσιμο να μπορούμε να γνωρίζουμε τι θα συμβεί στο μέλλον.
    c. Για το λόγο αυτό έγιναν πειράματα κυρίως σε χρονικές περιόδους (timesteps) διάρκειας 15 λεπτών για τις γρήγορες εναλλαγές της κατανομής, ωστόσο έγιναν και κάποιες αναλύσεις σε επίπεδο 30-60 λεπτών για τις πιο μακροπροθεσμες μεταβάσεις όπως αυτές του καιρού και των καλλιτεχνών που θα μελετήσουμε σε επόμενη ενότητα.

4. Γίνεται αντιληπτό ότι δε μας ενδιαφέρει η διαδρομή που ακολουθεί ο κάθε επισκέπτης, εφόσον τα στίγματα συλλέγονται μόνο όταν εκείνος επιλέγει να χρησιμοποιήσει την εφαρμογή του festival και άρα δε μπορούν να μας τροφοδοτήσουν με ένα συνεχές μονοπάτι κίνησης. Το βασικό στοιχείο που μας ενδιαφέρει είναι απρόσωπα και μακροσκοπικά η κατανομή του συνόλου των χρηστών στο χώρο από στιγμή σε στιγμή. Οφείλουμε λοιπόν να χωρίσουμε το χώρο στον οποίο πραγματοποιείται το festival σε επιμέρους, συνεχείς AoIs στις οποίες θα μετρήσουμε τους χρήστες στο εσωτερικό τους και θα προσπαθήσουμε στη συνέχεια να προβλέψουμε την επόμενη κατανομή τους. Οι περιοχές αυτές μπορούν να επιλεχθούν είτε στατικά, ορίζοντας πολύγωνα στο χώρο, είτε επιστρατεύοντας μεθόδους μη επιβλεπόμενης μηχανικής μάθησης που θα δούμε στην επόμενη ενότητα.

7.2.5 Εξαγωγή και αναπαράσταση χρήσιμων χαρακτηριστικών - Feature Engineering

Όπως είναι φυσικό, για να πραγματοποιήσουμε μια πρόβλεψη σε ένα πρόβλημα επιτηρούμενης μάθησης είναι απαραίτητο να "ανακαλύψουμε" χαρακτηριστικά - features με τα οποία θα τροφοδοτήσουμε τους αλγόριθμους, τέτοια ώστε να παρέχουν πληροφορίες και να φανερώνουν μοτίβα τα οποία μπορούν οι αλγόριθμοι να εντοπίσουν και να αξιοποιήσουν.

7.2.5.1 Τα χαρακτηριστικά της κατανομής επισκεπτών

Όπως έγινε φανερό στην ενότητα 7.2.2 το βασικότερο χαρακτηριστικό εισόδου αποτελεί η κατανομή των επισκεπτών της παρούσας χρονικής περιόδου ανάμεσα στα AoIs. Το χαρακτηριστικό αυτό είναι το πιο ιδιαίτερο καθώς βρίσκεται τόσο στην είσοδο όσο και στην επιθυμητή έξοδο με μία χρονική ολίσθηση ενός timestep (βλ. Πίνακα ενότητας 2.2.2). Αποτελείται από επιμέρους χαρακτηριστικά που είναι τα πλήθη ή τα ποσοστά επισκεπτών σε κάθε AoI. Το μόνο που μένει λοιπόν είναι ο καθορισμός του πλήθους και της τοποθεσίας των PoIs. Ακριβώς επειδή ο αριθμός δοθέντων PoIs είναι πολύ μεγάλος και μη πρακτικός για τις προβλέψεις μας, καθώς θα εισάγει ένα τεράστιο αριθμό features με πολύ αραιές (sparse) τιμές, έγινε προσπάθεια εναλλακτικού ορισμού των PoIs. Η έρευνα που έγινε στο κομμάτι της εξαγωγής της τοποθεσίας και του πλήθους των PoIs και AoIs αναλύεται σε βάθος σε επόμενη ενότητα.

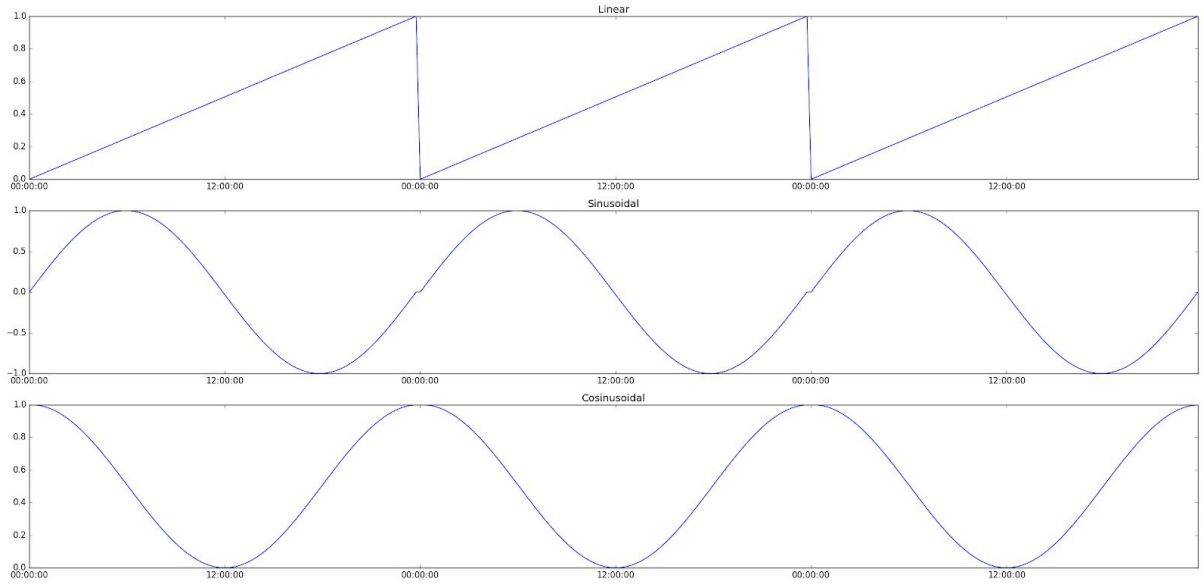
A	B	C	D	E	F	Time
26	2	11	4	4	7	2017-07-21 17:45:00
26	9	6	5	8	9	2017-07-21 18:00:00
20	6	3	8	6	4	2017-07-21 18:15:00
16	6	4	4	6	8	2017-07-21 18:30:00

Πίνακας 7.3: Ο αριθμός ανθρώπων σε 6 PoIs για 4 συνεχόμενες χρονικές περιόδους

7.2.5.2 Το χαρακτηριστικό προσδιορισμού της χρονικής στιγμής (time index)

Δεδομένου ότι ο στόχος μας είναι να κάνουμε πρόβλεψη στο χρόνο, καθίσταται απαραίτητη η ένταξη ενός χαρακτηριστικού εισόδου το οποίο θα αφορά στο χρόνο. Όπως ήδη αναφέρθηκε στην ενότητα 7.2.4.3 ο χρόνος θα μελετηθεί ως μια σειρά από χρονικές περιόδους κάποιας διάρκειας.

Επιπλέον, είναι πολύ σημαντικό να μοντελοποιηθούν τα κοινα χαρακτηριστικά μεταξύ της ίδιας χρονικής στιγμής για δύο διαφορετικές μέρες. Για παράδειγμα αναμένουμε παρόμοιες συμπεριφορές στην κινητικότητα και τα πλήθη των επισκεπτών στις 2μμ του Σαββάτου με τις 2μμ της Κυριακής. Αυτό σημαίνει πως χρειαζόμαστε μία περιοδική αναπαράσταση του χρόνου η οποία θα ορίζει ένα δείκτη για την παρούσα χρονική περίοδο του κάθε instance. Οι δείκτες αυτοί (time indexes) αντιστοιχίζουν κάθε χρονική περίοδο σε έναν αύξοντα αριθμό κανονικοποιημένο στο [0,1]. Η αναπαράσταση αυτή θα είναι περιοδική με περιόδο μήκους όσων timesteps περιέχονται σε ένα 24ωρο. Έγινε πειραματισμός σε μια σειρά από μετασχηματισμούς όπως γραμμικοί και ημιτονοειδείς στο αρχείο time_indexes_plot.py (εικόνα 7.7) και τελικά επιλέχτηκε ο γραμμικός μετασχηματισμός. Οι ημιτονοειδής απορρίφθηκαν, παρότι διαισθητικά φαίνονται πιο σωστοί, κι αυτό διότι διατηρούν τις εξάρτηση μεταξύ για παράδειγμα 11μμ και 1πμ της επόμενης ημέρας δίνοντας τους ίδιο index (τιμή κατακόρυφου άξονα) πράγμα που φάνηκε να μπερδεύει τον αλγόριθμο καθώς τις νυχτερινές ώρες ο χώρος παρέμενε άδειος. Τελικά η μορφή του χαρακτηριστικού του time index πήρε τη μορφή του ενδεικτικού πίνακα 7.4 στον οποίο απεικονίζονται οι τιμές της ανεξάρτητης μεταβλητής Time Index κατά μετάβαση από τη μία μέρα στην επόμενη για timestep 15 λεπτων.



Εικόνα 7.7: Οι εναλλακτικές αναπαραστάσεις για τους δείκτες των χρονικών περιόδων κανονικοποιημένες στο [0,1].

Time index	Time
92	2017-07-22 23:00:00
93	2017-07-22 23:15:00
94	2017-07-22 23:30:00
95	2017-07-22 23:45:00
0	2017-07-23 00:00:00
1	2017-07-23 00:15:00

Πίνακας 7.4: Το διάνυσμα του Time Index σε σχέση με τον πραγματικό χρόνο για timestep 15 λεπτών

7.2.5.3 Τα χαρακτηριστικά του καιρού

Αναζητώντας νέα χαρακτηριστικά τα οποία μπορεί να επηρεάζουν την προσέλευση και την κατανομή των επισκεπτών στο χώρο του φεστιβάλ, ο καιρός φάνηκε να είναι μια σημαντική παράμετρος η οποία και αποφασίστηκε να συμπεριληφθεί στο σύνολο των ανεξάρτητων μεταβλητών - χαρακτηριστικών του προβλήματος. Τα ιστορικά στοιχεία του καιρού για τις ημερομηνίες του φεστιβάλ αναζητήθηκαν σε αξιόπιστη ιστοσελίδα καιρικών προγνώσεων[34]. Κρατήθηκαν δεδομένα θερμοκρασίας και καιρικών συνθηκών στο αρχείο weather history.csv, με τη μορφή του πίνακα 7.5. Τα missing data (π.χ έλλειψη μετρήσεων τις πρωινές ώρες 00:00 - 04:50) αντικαταστάθηκαν με τις συνθήκες των πιο κοντινών χρονικών στιγμών. Στα ενδιάμεσα χρονικά διαστήματα μεταξύ μετρήσεων θεωρούμε συνθήκες και θερμοκρασία ίδιες με της τελευταίας μέτρησης καθώς τα στοιχεία που έχουμε αφορούν μισάωρα, ενώ οι προβλέψεις μας έγιναν κυρίως σε πιο μικρά χρονικά διαστήματα.

Time (GMT)	Temperature	Conditions
21-07-2017 00:00:00	15 °C	Clear.
21-07-2017 4:50:00	16 °C	Clear.
21-07-2017 5:20:00	16 °C	Clear.
21-07-2017 5:50:00	15 °C	Sunny.
21-07-2017 6:20:00	15 °C	Sunny.
21-07-2017 6:50:00	16 °C	Sunny.

Πίνακας 7.5: Το ιστορικά δεδομένα του καιρού που χρησιμοποιήθηκαν ως χαρακτηριστικά εισόδου

[34] *https://www.timeanddate.com/weather/germany/karlsruhe/historic?month=7&year=2017*

Οι καιρικές συνθήκες, όντας κατηγορική μεταβλητή, κωδικοποιήθηκαν χειροκίνητα σε μία κλίμακα ακεραίων στο [0,3] με απλό τρόπο όπως φαίνεται στο αρχείο cond.yaml το οποίου το περιεχόμενο παρατίθεται ενδεικτικά και στην εικόνα 7.8

```
Broken clouds.: 2
Clear.: 3
Fog.: 2
Mostly cloudy.: 1
Overcast.: 1
Partly cloudy.: 2
Partly sunny.: 2
Passing clouds.: 2
Rain showers. Mostly cloudy.: 1
Rain showers. Passing clouds.: 1
Scattered clouds.: 1
Scattered showers. Broken clouds.: 1
Scattered showers. Clear.: 1
Scattered showers. Fog.: 1
Scattered showers. Overcast.: 1
Scattered showers. Partly sunny.: 1
Scattered showers. Passing clouds.: 1
Scattered showers. Scattered clouds.: 1
Sprinkles. Cloudy.: 1
Sprinkles. Overcast.: 1
Sunny.: 3
Thundershowers. Broken clouds.: 0
Thundershowers. Mostly cloudy.: 0
Thundershowers. Partly cloudy.: 0
Thundershowers. Partly sunny.: 0
Thunderstorms. Partly cloudy.: 0
Thunderstorms. Partly sunny.: 0
Thunderstorms. Passing clouds.: 0
```

Εικόνα 7.8: Το αρχείο κωδικοποίησης των δεδομένων καιρικών συνθηκών.

Συνδυάζοντας όλα τα παραπάνω, καταλήγουμε με χρήση της συνάρτησης weather.py να επιστρέφουμε ένα pandas DataFrame δύο στηλών με τα ζητούμενα χαρακτηριστικά καιρου (πίνακας 7.6). Η συνάρτηση weather δέχεται σαν όρισμα το μεταβλητού μεγέθους DataFrame για το οποίο καλείται να κατασκευάσει τις στήλες καιρού και το κάνει με τον κατάλληλο τρόπο ώστε να αντιστοιχίζει τον καιρό με τις τις ώρες που αφορούν τα time indexes του DataFrame αυτού.

Temperature	Conditions	Time index	Time Period
15	3	17	2017-07-21 04:15:00
15	3	18	2017-07-21 04:30:00
15	3	19	2017-07-21 04:45:00
16	3	20	2017-07-21 05:00:00
16	3	21	2017-07-21 05:15:00
16	3	22	2017-07-21 05:30:00
16	3	23	2017-07-21 05:45:00
15	3	24	2017-07-21 06:00:00
15	3	25	2017-07-21 06:15:00

Πίνακας 7.6: Οι στήλες χαρακτηριστικών καιρού αντιστοιχισμένες στις σωστές χρονικές περιόδους

7.2.5.4 Τα χαρακτηριστικό της δημοτικότητας των καλλιτεχνών (popularity meter)

Λαμβάνοντας υπόψη τον κατ' εξοχήν μουσικό χαρακτήρα του DasFest καταλήξαμε στο συμπέρασμα ότι, ενδεχομένως, οι δημοτικότητες των καλλιτεχνών - συγκροτημάτων που δίνουν ζωντανή εμφάνιση ανά τις χρονικές στιγμές να επηρεάζουν την μετακίνηση και την κατανομή των επισκεπτών, κυρίως στις περιοχές των συναυλιακών χώρων. Για παράδειγμα είναι αναμενόμενη η ραγδαία αύξηση της συγκέντρωσης ανθρώπων στην AoI που ορίζει η κεντρικής σκηνή, όταν εμφανίζεται ένας διάσημος καλλιτέχνης, συνοδευόμενη από αραίωση στις υπόλοιπες περιοχές.

Αρχικά συμβουλευτήκαμε το πρόγραμμα του φεστιβάλ προκειμένου να εντοπίσουμε τους καλλιτέχνες που παρουσιάζονται, την ώρα και τη σκηνή στην οποία εμφανίζεται ο καθένας. Καταλήξαμε, λοιπόν, στο αρχείο *artists_list.csv*, με τη μορφή του πίνακα 7.7, στο οποίο αναγράφονται τα ονόματα των καλλιτεχνών, οι ώρες και οι τοποθεσίες εμφάνισης με όνομα σκηνής και ακριβείς συντεταγμένες. Για τις ακριβείς συντεταγμένες έγινε χρήση του δοθέντος από το basmati αρχείου *festinfrastructure.json* της ενότητας 7.2.1.

Artist	Time	Buhne	Timestamp	X	Y
Donots	17:30	Hauptbuhne	2017-07-21 17:30:00	8.37376284	48.99776582
Jennifer Rostock	19:10	Hauptbuhne	2017-07-21 19:10:00	8.37376284	48.99776582
Sportfreunde Stiller	21:00	Hauptbuhne	2017-07-21 21:00:00	8.37376284	48.99776582
Meute	23:00	Hauptbuhne	2017-07-21 23:00:00	8.37376284	48.99776582
Mars of Illyricum	20:00	Feldbuhne	2017-07-21 20:00:00	8.37221654	49.00138549
Astronautalis	21:15	Feldbuhne	2017-07-21 21:15:00	8.37221654	49.00138549
Curse	22:30	Feldbuhne	2017-07-21 22:30:00	8.37221654	49.00138549
OstWest Brothers	18:00	DJ-Buhne	2017-07-21 18:00:00	8.3721874	48.99948011
DJ SiMa	19:30	DJ-Buhne	2017-07-21 19:30:00	8.3721874	48.99948011

Πίνακας 7.7: Η μορφή του αρχείου μουσικού προγράμματος του φεστιβάλ

Αναζητήσαμε, λοιπόν, στο διαδίκτυο ιστοσελίδες με κατατάξεις, βαθμολογίες και δημοτικότητες καλλιτεχνών. Καταλήξαμε να αντλήσουμε μεσω του API[35] της ιστοσελίδας Next Big Sound[36] για όλους τους καλλιτέχνες οι οποίοι εμφανίστηκαν στο φεστιβάλ και τις δύο χρονιές 2017, 2018. Το Next Big Sound παρέχει μια σειρά από μετρικές αξιολόγησης όπως είναι η διασημότητα του καλλιτέχνη, οι φορές αναπαραγωγής κομματιών του, το δέσιμο με το κοινό του, τα likes kai follows σε μέσα κοινωνικής δικτύωσης όπως το facebook, το twitter και το instagram. Θέλοντας να απλοποιήσουμε τα πράγματα, περιοριστήκαμε σε δύο κύριες μετρικές που είναι το stage και το audience engagement του καλλιτέχνη.

[35] *https://www.programmableweb.com/api/next-big-sound*
[36] *https://www.nextbigsound.com*

Στο αρχείο *artists_metrics.py* γίνεται άντληση των μετρικών αυτών, ρυθμίζοντας και τις περιπτώσεις μη ύπαρξης τους για ορισμένους καλλιτέχνες, και εναπόθεση τους στο αρχείο metrs.xls όπως φαίνεται στον πίνακα 7.8.

Artist	Stage	Audience eng.
Reaching 62 F	-2	Undiscovered
All Haze Red	-2	Undiscovered
Drangsal	0.61611207471973	Promising
Zebrahead	0.79349809596928	Established
Feine Sahne Fischfilet	1.0014764727241	Established

Πίνακας 7.8: Η δομή του αρχείου μετρικών των καλλιτεχνών του φεστιβάλ

Τελικά κάνουμε χρήση μίας υβριδικής μετρικής (Popularity meter) η οποία κατασκευάζεται στο script *artists_period.py* και αποτελεί ένα σταθμισμένο μέσο των άλλων δύο, ενώ ταυτόχρονα είναι κανονικοποιημένη στο διάστημα [0,1]. Σημειώνονται τα εξής:

1. Το stage παίρνει συνεχείς τιμές στο [-2,2]
2. Το audience engagement αποτελεί κατηγορική μεταβλητή αλλα μέσα στο script ανάγεται κ αυτό με χρήση Label Encoder στο διάστημα ακεραίων {0,3}

Το script *artists_period.py* ζητά ως παραμέτρους τη διάρκεια του timestep, ένα αρχείο της μορφής *pois_to_clusters.csv*, και μια παράμετρο *only_start* που καθορίζει κάποιες εσωτερικές διατάξεις στις μετρικές. Επεξηγηματικά, το αρχείο pois_to_clusters.csv έχει τη μορφή του πίνακα 7.7, με μία παραπάνω στηλη η οποία προσδιορίζει τα PoIs στα οποία ανήκουν οι σκηνές. Η αντιστοίχιση των συντεταγμένων των σκηνών σε PoIs πραγματοποιείται σε scripts τα οποία θα δούμε σε επόμενη ενότητα και συναρτάται άμεσα με τον τρόπο εξαγωγής των PoIs. Επιπλέον, η παράμετρος *only_start* μας δίνει τη δυνατότητα να θέσουμε τη μετρική του καλλιτέχνη μόνο τη χρονική περίοδο που ξεκινάει η για όλες τις περιόδους κατα τις οποίες υποθέτουμε ότι κάνει την εμφάνιση του.

7.2.5.5 Παρατηρήσεις της διαδικασίας εξαγωγής χαρακτηριστικών

Έχοντας εισάγει τα δύο "εξωτερικά" χαρακτηριστικά του καιρου και του popularity στο πρόβλημα μας είναι πολύ σημαντικό να αναφερθούμε στον "open data" χαρακτήρα τους. Είδαμε ότι το Next Big Sound παρέχει δωρεάν API. Επιπλέον, είναι δεδομένο πως η πρόγνωση του καιρού είναι διαθέσιμη από το διαδίκτυο ανά πάσα στιγμή. Τα παραπάνω μας οδηγούν σε δύο βασικά συμπεράσματα:

1. Είναι ιδιαίτερα εύκολο, σε οποιοδήποτε σύστημα να αντλει δεδομένα καιρού και popularity με αμεσότητα, ποιότητα δεδομένων και ακρίβεια προκειμένου να τα εισάγει στα μοντέλα μηχανικής μάθησης.
2. Σε κάθε timestep t δεν είμαστε υποχρεωμένοι να χρησιμοποιούμε ως γνώρισμα εισόδου τις τιμές της παρούσας χρονικής στιγμής για τα δύο αυτά features. Αυτό σημαίνει ότι μπορούμε να εισάγουμε την πρόγνωση του καιρού της χρονικής στιγμής t+1 ως χαρακτηριστικό εισόδου της στιγμής t εφόσον αυτή είναι γνωστή, ελπίζοντας έτσι να βελτιώσουμε περαιτέρω την πρόβλεψης κατανομής μας. Αντίστοιχα εφόσον το πρόγραμμα και τα popularities των

καλλιτεχνών είναι εκ των προτέρων γνωστά και διαθέσιμα, είναι αντίστοιχα ρεαλιστικό να "ολισθήσουμε" προς το μέλλον το χαρακτηριστικό αυτό, αν αυτό μας εξυπηρετεί.

7.2.6 Εντοπισμός PoIs με μεθόδους συσταδοποίησης

Οπως έγινε αντιληπτό το δοθέντα PoIs είναι πάρα πολλά στο πλήθος ώστε να δομήσουμε κατανομές ανάμεσα σε όλα. Η "κατάρα της διαστατικότητας" δημιουργεί πολλα προβλήματα, καθώς κάνει τα δεδομένα μας να είναι πολύ πιο αραιά στο χώρο και άρα πολύ πιο δύσκολο τον εντοπισμό μοτίβων. Για το λόγο αυτό αποφασίσαμε να περιοριστούμε σε ένα σημαντικά μικρότερο αριθμό από PoIs.

Δεδομένου ότι δεν υπήρξε κάποιος περιορισμός ως προς τα PoIs, αποφασίσαμε να αγνοήσουμε τα δοθέντα PoIs και να προσπαθήσουμε να εντοπίσουμε νέα με μεθόδους μη επιτηρούμενης μάθησης και πιο συγκεκριμένα συσταδοποίησης. Όπως είδαμε στο κεφάλαιο 5, σκοπός της είναι η δημιουργία ομάδων σημείων τα οποία παρουσιάζουν κοινά χαρακτηριστικά μεταξύ τους. Συνεπώς, θεωρήσαμε πως αυτή τη διαδικασία θα μπορούσε να αποκαλύψει κάποια μοτίβα στο τρόπο μετακίνησης και κατανομής των επισκεπτών μέσα στο χώρο του φεστιβάλ.

Ταυτόχρονα, είχαμε στο μυαλό μας ότι η κίνηση και οι επιλογές των επισκεπτών μέσα στο χώρο του φεστιβάλ, είναι αυτά που στην πραγματικότητα καθορίζουν τα σημεία ενδιαφέροντος. Αντιμετωπίζουμε δηλαδή τον ορισμό των PoIs ως δυναμική διαδικασία. Σαν ενδεικτικό παράδειγμα μπορούμε να φανταστούμε ότι οι επισκέπτες, λόγω του ότι έχει καλό καιρό επιλέγουν, να κάνουν πικ-νικ στα πάρκα του χώρου του φεστιβάλ αντί για παράδειγμα να ψωνίζουν από τα εστιατόρια. Αυτόματα έχει δημιουργηθεί μία περιοχή ενδιαφέροντος, η οποία δεν είναι προβλεπόμενη από τις εγκαταστάσεις του φεστιβάλ και δεν υφίσταται ως PoI στο αρχείο festinfrastructure.json. Ωστόσο αξίζει να μελετηθεί και να θεωρηθει ως τέτοιο διότι συγκεντρώνει ένα μεγάλο πλήθος επισκεπτών.

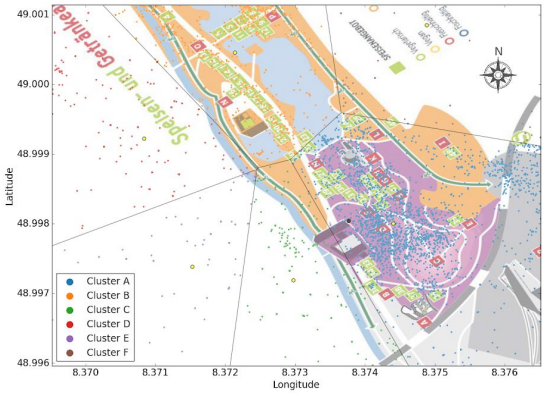
7.2.6.1 Οι υποψήφιοι αλγόριθμοι συσταδοποίησης.

Κατά τον πειραματισμό με τα δεδομένα έγινε χρήση τριών αλγόριθμων συσταδοποίησης:

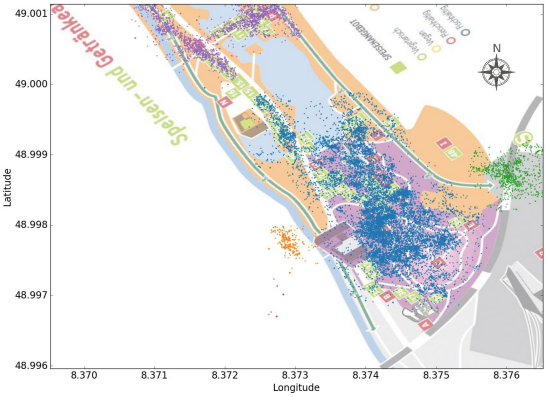
1. k-means,
2. Mean-shift
3. DBSCAN

Παραθέτουμε και ενδεικτικές εικόνες των αποτελεσμάτων συσταδοποίησης τους. Οι k-means και Mean-shift είναι αλγόριθμοι κεντροειδών και για το λόγο αυτό στις αντίστοιχες εικόνες βλέπουμε τα κέντρα των συστάδων (κίτρινα σημεία) και τις αντίστοιχες AoIs, που ορίζονται από διάγραμμα Voronoi για τα κέντρα αυτά.

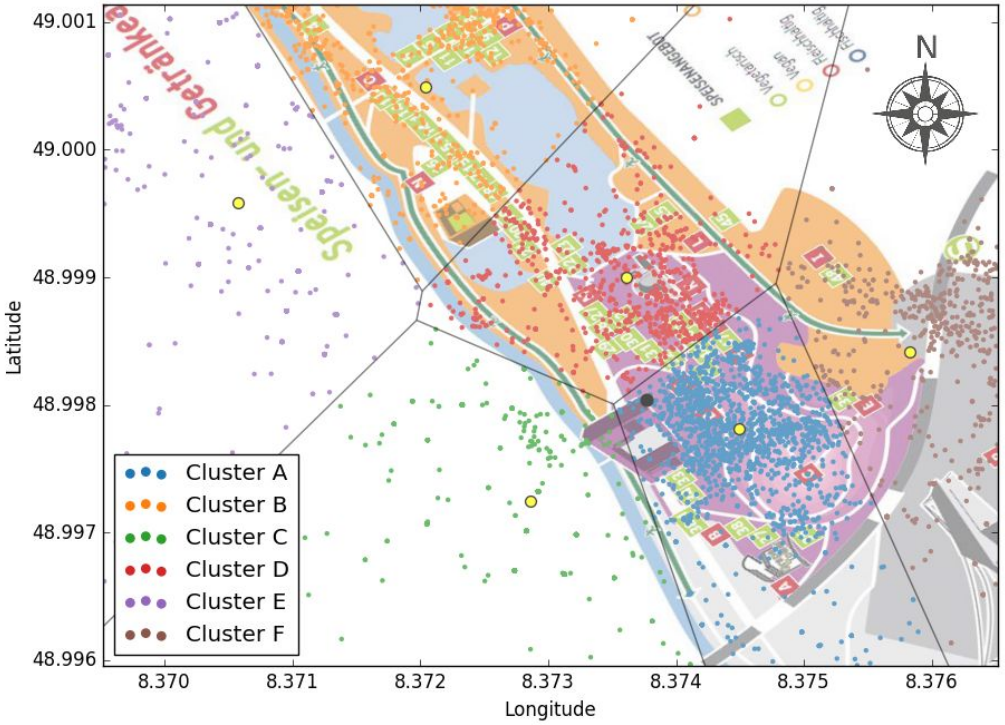
Ο αλγόριθμος mean-shift (εικόνα 7.9) εκκινεί με ορισμένα δοθέντα κεντρικά σημεία. Έχει ως κύρια υπερπαράμετρο το bandwidth το οποίο ορίζει την ακτίνα της περιοχής γύρω από κάθε κεντρικό σημείο, για την οποία υπολογίζεται η μέση τιμή των σημείων που περιλαμβάνει, με σκοπό μετακίνησης του κέντρου του αντίστοιχου cluster στο επόμενο βήμα. Ο αλγόριθμος DBSCAN (εικόνα 7.10) είναι density based και απαιτεί ορισμό δύο παραμέτρων eps, min_samples .



Εικόνα 7.9: Ενδεικτικό mean-shift clustering (bandwidth=0.00098)



Εικόνα 7.10: Ενδεικτικό DBSCAN clustering (eps=0.000345, min_samples=700)



Εικόνα 7.11: Ενδεικτικός k-means clustering για k=6

Όσον αφορά στην εφαρμογή της συσταδοποίησης για την τελική εξαγωγή των AoIs χρησιμοποιήθηκε τελικά ο αλγόριθμος k-means (εικόνα 7.11) , για τους εξής λόγους:

1. Ο k-means είναι ένα σχετικά γρήγορος αλγόριθμος ο οποίος μπορεί πολύ εύκολα να χρησιμοποιηθεί σε εφαρμογές Fog Computing. Έχει ήδη υλοποιηθεί σε περιβάλλον Apache Mahout[37], γεγονός που διευκολύνει τη χρήση του στο Cloud.
2. Ο k-means έχει το πλεονέκτημα, για εμάς, ότι η κύρια υπερπαράμετρος του προς ορισμό, είναι ο αριθμός των clusters. Δεδομένου ότι ασχολούμαστε με μικρό αριθμό AoIs ή συστάδων (k<10), αυτό ήταν πολύ χρήσιμο για εμάς και βοηθητικό ως προς τη διαισθητική επαφή και

[37] *https://mahout.apache.org*

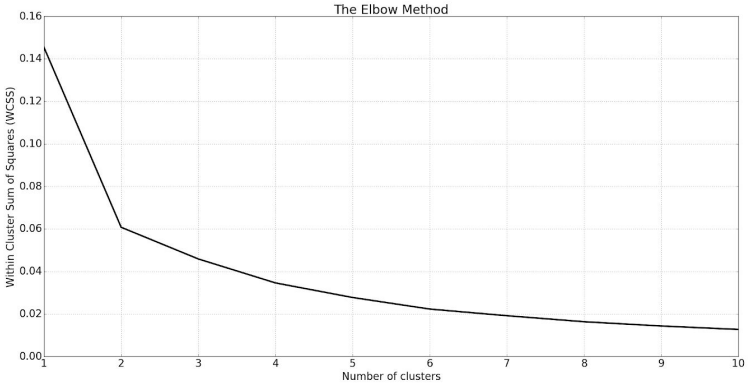
έλεγχο της αναζήτησης του βέλτιστου σχήματος συσταδοποίησης. Με λίγα λόγια, μπορούσαμε να τρέχουμε τον αλγόριθμο για τις διάφορες τιμές του k και να κρίνουμε, κυρίως οπτικά την ποιότητα των αποτελεσμάτων της συσταδοποίησης.

3. Ο k-means, ως αλγόριθμος κεντροειδών, εξάγει τα κέντρα των υποψηφίων PoIs. Στη συνέχεια είναι πολύ εύκολο να δημιουργηθούν οι αντίστοιχες AoIs και να οπτικοποιηθούν μέσω διαγράμματος Voronoi, χωρίς να αφήνονται ανένταχτα σημεία, όπως για παράδειγμα συμβαίνει με τον αλγόριθμο DBSCAN. Τα παραπάνω καθιστούν αρκετά ευκολότερη την αντιστοίχιση των στιγμάτων του χάρτη στη χωρική συστάδα στην οποία ανήκουν
4. Τα αποτελέσματα του k-means για 6 συστάδες, ήταν πάρα πολύ συνεπή ως προς το πρόγραμμα, τα δρώμενα του φεστιβάλ και τη δομή των εγκαταστάσεων του στο χώρο. Για παράδειγμα το cluster F αντιστοιχεί στην περιοχή εισόδου και εξόδου των επισκεπτών στην οποία βρίσκεται και η πύλη του φεστιβάλ. Το cluster A αναπαριστά την περιοχή γύρω από την κεντρική σκηνή του φεστιβάλ, στην οποία λαμβάνουν χώρα οι ζωντανές εμφανίσεις των καλλιτεχνών. Το cluster D αντιστοιχίζεται στην περιοχή αναψυχής εστίασης και αγορών φεστιβάλ, όπου οι επισκέπτες συνεστιάζονται κυρίως όταν δεν υπάρχει κάποιο συγκεκριμένο δρώμενο. Το cluster B περιλαμβάνει 3 σκηνές και λίγα σημεία εστίασης, ενώ τα άλλα δύο clusters αναπαριστούν περιοχές εκτός του φεστιβάλ όπου οι επισκέπτες πιθανόν να άνοιγαν την εφαρμογή προκειμένου να εντοπίσουν την τοποθεσία της κεντρικής εισόδου, ερχόμενοι στο φεστιβάλ.

7.2.6.2 Η υλοποίηση της συσταδοποίησης k-means

Στο αρχείο *clustering_belos.py* έχει κατασκευαστεί κλάση my_kmeans η οποία βασίζεται στον αλγόριθμο k-means της scikit-learn. Ωστόσο, προσφέρει τις εξής παραπάνω χρήσιμές δυνατότητες μέσω μεθόδων:

- ❏ Χρήση του elbow method (εικόνα 7.12) και εκτύπωση των αποτελεσμάτων του με σκοπό την επιλογή του αριθμού των clusters. Ο εντοπισμός του σημείου αγκώνα συνδυαστικά με τα πραγματικά δεδομένα (πρόγραμμα, δομή εγκαταστάσεων) του φεστιβάλ και τη λογική μας οδήγησαν στην επιλογή 6 στο πλήθος συστάδων.
- ❏ Εκτύπωση των αποτελεσμάτων συσταδοποίησης πάνω στον χάρτη του φεστιβάλ και προαιρετική (μέσω ορίσματος) αποθήκευση τους σε αρχείο png.
- ❏ Προαιρετική (μέσω ορίσματος) εκτύπωση Voronoi διαγράμματος γύρω από τα κέντρα των clusters που αποφασίζει ο αλγόριθμος όπως είδαμε στην εικόνα 7.11.



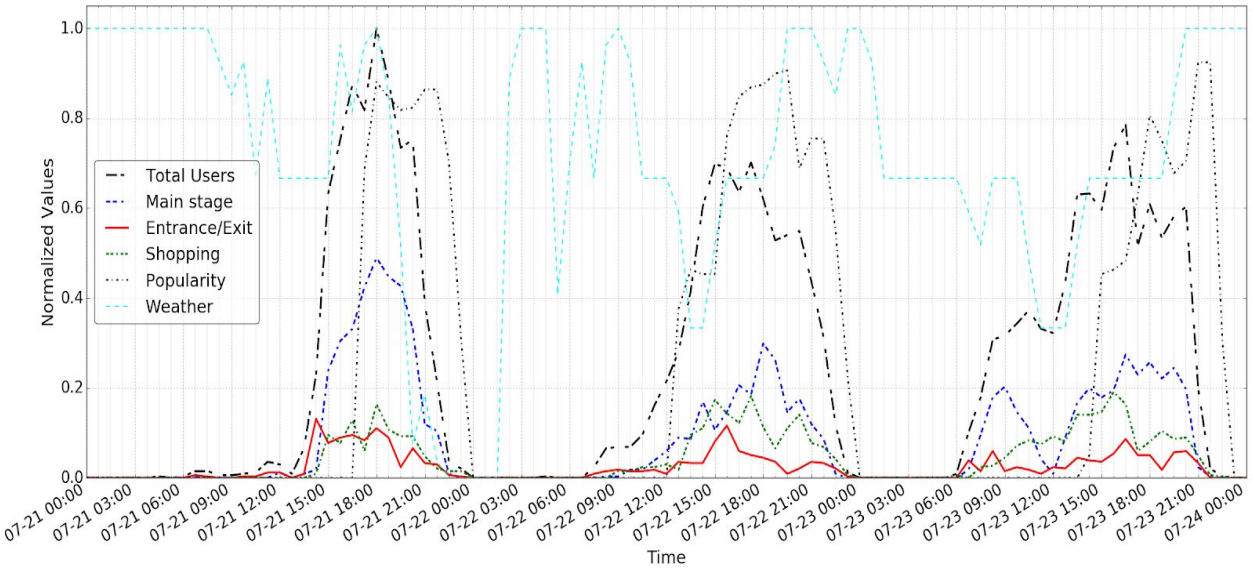
Εικόνα 7.12: Elbow method για k στο [1,10]

7.2.7 Ανάλυση δεδομένων και παρατηρήσεις

Έχοντας ορίσει πλήρως το πρόβλημά μας και τη δομή του, προχωράμε σε μία ανάλυση και παρατήρηση των δεδομένων, η οποία μας προσφέρει μία διαισθητική επαφή και μία ποιοτική απεικόνιση της κατάστασης. Τα παραπάνω είναι ζωτικής σημασίας προκειμένου να έχουμε πλήρη εποπτεία των προβλημάτων επιτηρούμενης μάθησης που θα αναλύσουμε στις επόμενες ενότητες.

7.2.7.1 Το κύριο γράφημα αναπαράστασης των χαρακτηριστικών

Στο γράφημα της εικόνας 7.13 συνοψίζεται ένα σύνολο από πληροφορίες, οι οποίες αφορούν τα χαρακτηριστικά εισόδου κατα το τριήμερο διεξαγωγής του φεστιβάλ για τον Ιούλιο του 2017. Οι διάφορες περιοχές ενδιαφέροντος έχουν τα ονόματα του χώρου στον οποίο απευθύνονται, όπως φαίνεται στο υπόμνημά του γραφήματος ενώ παραθέτουμε ταυτόχρονα και τις γραφικές παραστάσεις της δημοτικότητας των καλλιτεχνών και των καιρικών συνθηκών στο χρόνο. Τα γραφήματα αφορούν σε χρονικές περιόδους διάρκειας 45 λεπτών. Η επιλογή 45 λεπτών έγινε ώστε να έχουμε την καθαρότερη και ομαλότερη δυνατή εικόνα χάνοντας τη λιγότερη δυνατή πληροφορία. Επιπλέον, όλα τα μεγέθη είναι κανονικοποιημένα στο διάστημα (0,1) ώστε να γίνεται εμφανής η ποιοτική σχέση μεταξύ τους. Ο συντελεστής κανονικοποίησης των πληθών επισκεπτών στην κάθε AoI είναι η μέγιστη τιμή που λαμβάνει η μεταβλητή “Total Users” η οποία είναι 125 και παρατηρείται στο 45λεπτο 2017-07-21 18:30:00-19:15:00. Η μεταβλητή αυτή αναφέρεται στο συνολικό αριθμό επισκεπτών στο χώρο του φεστιβάλ ανα πάσα στιγμή. Σημειώνεται πως η κανονικοποίηση πλήθους γίνεται ως προς τον ίδιο συντελεστή για όλες τις AoIs προκειμένου να καταδεικνύεται και η ποσοτική σχέση μεταξύ τους. Όσον αφορά στη δημοτικότητα και στον καιρό κανονικοποιούνται το καθένα σε δική του κλίμακα καθώς δεν υπάρχει νόημα αναζήτησης κάποιας ποσοτικής σχέσης μεταξύ τους και μεταξύ αυτών και των πληθών.



Εικόνα 7.13: Συνδυασμένες γραφικές παραστάσεις όλων των χαρακτηριστικών του προβλήματος κανονικοποιημένων στο [0,1]

7.2.7.2 Συμπεράσματα της ανάλυσης δεδομένων ως προς τις ημέρες διεξαγωγής

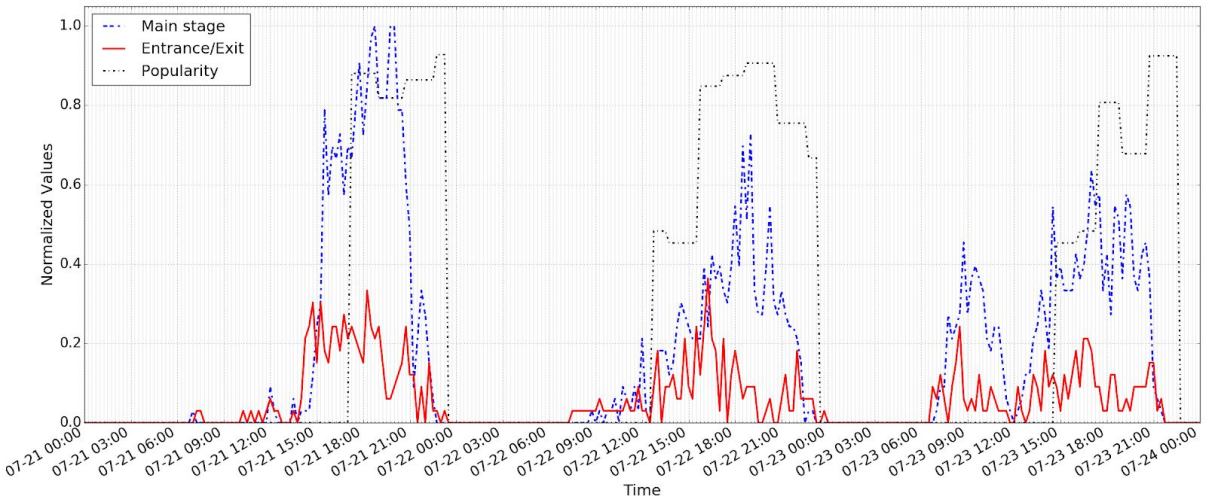
1. Καθημερινά μεταξύ 12:00 και 18:00 παρατηρούμε μία αυξητική τάση στα πλήθη επισκεπτών σε όλες τις μέρες.

2. Αντιθέτως καθημερινά μεταξύ 18 και μηδέν παρατηρούμε μία πτωτική τάση στα πλήθη επισκεπτών σε όλες τις μέρες.
3. Οι Παρασκευές είναι εργάσιμες μέρες και για αυτό το λόγο οι επισκέπτες καταφθάνουν αργότερα στο φεστιβάλ και παρατηρούμε μία απότομη αύξηση κατά το απόγευμα.
4. Το Σάββατο οι επισκέπτες είναι κατανεμημένοι πιο ισορροπημένα στις διάφορες ώρες της ημέρας.
5. Την Κυριακή παρατηρούμε μία χρονική περίοδο μεταξύ 8 και 12 την οποία θα χαρακτηρίζαμε ως outlier. Αυτό σημαίνει ότι δεν παρατηρείται αντίστοιχο μοτίβο κατανομής στις υπόλοιπες ημέρες. Παρατηρήσαμε ότι αυτή η συγκέντρωση ανθρώπων οφείλεται στην οργάνωση κυριακάτικων, πρωινών δραστηριοτήτων για παιδιά τις οποίες ωστόσο δεν έχουμε συμπεριλάβει στη μελέτη μας. Το γεγονός αυτό αναμένεται να βλάψει σε ένα βαθμό τις προβλέψεις μας, εντούτοις η μοντελοποίηση μιας περιόδου - outlier θα είχε χειρότερο αντίκτυπο στα μοντέλα μας.

7.2.7.3 Συμπεράσματα της ανάλυσης δεδομένων για 2 βασικές περιοχές ενδιαφέροντος

1. Το μέτρο της δημοτικότητας των καλλιτεχνών τείνει να πορεύεται μαζί με τη συγκέντρωση ανθρώπων στην κεντρική σκηνή, με μία σταθερή διαφορά φάσης η οποία ενδεχομένων να οφείλεται στο ότι ο κόσμος ξεκινάει να συγκεντρώνεται στην κεντρική σκηνή λίγο πριν τις εμφανίσεις των καλλιτεχνών για να μην τις χάσει. Αυτή η αρχή παραβιάζεται τις πολύ βραδινές ώρες, οπότε και η συγκέντρωση ανθρώπων πέφτει ξαφνικά προς στο μηδέν, πιθανώς επειδή ένα κλάσμα των ανθρώπων αναχωρεί λίγο πριν το τέλος της παράστασης η επειδή σταματά να χρησιμοποιεί το app του φεστιβάλ με σκοπό να απολαύσει την τελευταία ζωντανή εμφάνιση.
2. Όσον αφορά το cluster εισόδου/εξόδου παρατηρούνται απότομες αυξήσεις νωρίς το απόγευμα γύρω στις 14:00, οι οποίες πάντοτε συμβαίνουν μερικές χρονικές περιόδους πριν την αύξηση του μέτρου δημοτικότητας στην κεντρική σκηνή. Αυτό δικαιολογείται από το γεγονός ότι οι επισκέπτες καταφθάνουν νωρίτερα προκειμένου να παρακολουθήσουν τις ζωντανές εμφανίσεις. Αργά τη νύχτα τη στιγμή που παρατηρείται και η απότομη μείωση επισκεπτών στην κεντρική σκηνή παρατηρούνται μικρές αυξήσεις στην έξοδο, δεδομένου ότι οι επισκέπτες αναχωρούν μαζικά όπως προαναφέραμε. Προφανώς η αύξησεις αυτές είναι μικρότερες κατά μέτρο δεδομένου ότι δεν υπάρχει λόγος να ανοίξουν το app και να αναζητήσουν πληροφορίες κατά την αναχώρηση τους.
3. Από τα παραπάνω γίνεται φανερό ότι αυτές οι δύο συστάδες συμπεριφέρονται με ένα συμπληρωματικό τρόπο κυρίως σε ότι αφορά το ρυθμό μεταβολής - παράγωγο του πλήθους επισκεπτών τους. Με άλλα λόγια, απότομες αυξήσεις πλήθους στο ένα cluster συνοδεύονται από μειώσεις στο άλλο.

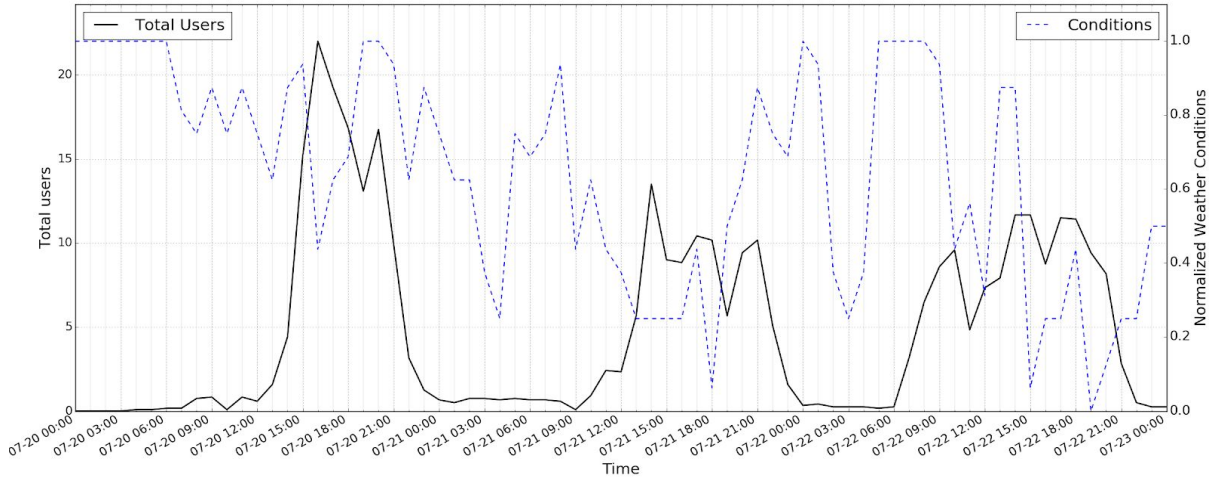
Στην εικόνα 7.14 παρατίθενται οι γραφικές παραστάσεις των δύο clusters δειγματοληπτημένες σε 15 λεπτο timestep, συνοδευόμενες και από το δείκτη δημοτικότητας των καλλιτεχνών που εμφανίζονται στην κεντρική σκηνή. Η δειγματοληψία έγινε, σε αυτή την περίπτωση, σε επίπεδο 15 λεπτών ώστε να απεικονίζονται οι γρήγορες μεταβολές. Επίσης αυτή η διάρκεια timestep χρησιμοποιείται και μετέπειτα, κατά τις διαδικασίες πρόβλεψης.



Εικόνα 7.14: Τα πλήθη στην είσοδο έξοδο και στην κεντρική σκηνή συνοδευόμενα και από τους δείκτες δημοτικότητας. Η δειγματοληψία έχει γίνει σε επίπεδο 15 λεπτών.

7.2.7.4 Συμπεράσματα της ανάλυσης δεδομένων για τις μεταβλητές του καιρού

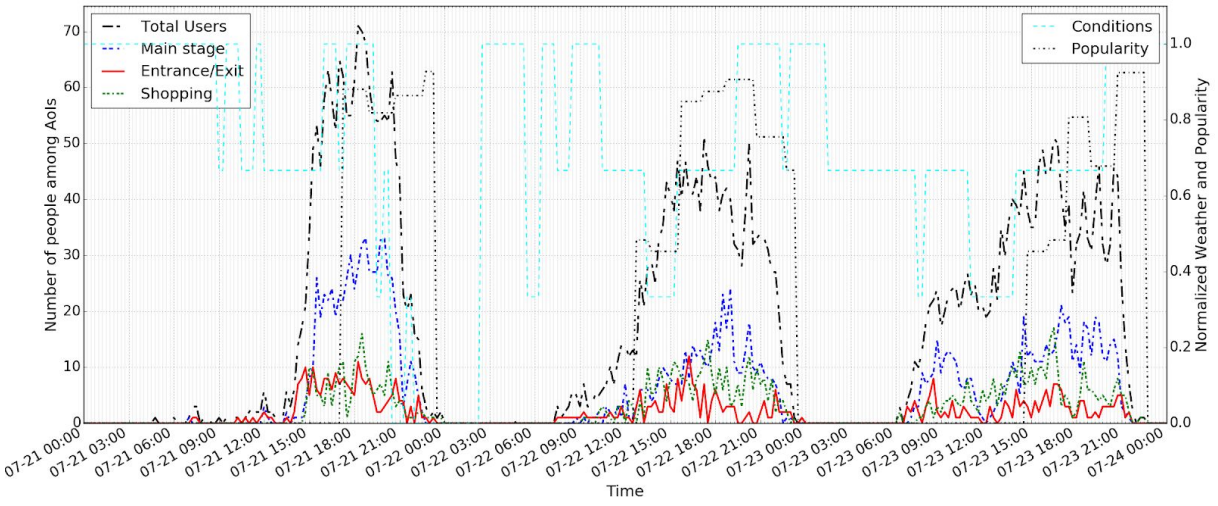
Ο καιρός φάνηκε να έχει πολύ σημαντικό ρόλο ως προς το συνολικό αριθμό των ατόμων που παρευρέθησαν στο φεστιβάλ. Παρ' όλα αυτα η επίδραση του φάνηκε να είναι αρκετά μακροπρόθεσμη ώστε να μπορεί να συνεισφέρει σε επίπεδο προβλέψεων μικρής διάρκειας timestep. Αρκετά έντονη, ωστόσο φάνηκε να είναι σε ότι αφορά την επισκεψιμότητα του 2018, κυρίως τις μέρες του Σαββάτου και της Κυριακής, όπου οι συγκεντρώσεις επισκεπτών ήταν μειωμένες τόσο ως πρός την Παρασκευή όσο και ως προς το 2017 εν γένει (Εικόνα 7.15).



Εικόνα 7.15: Η επίδραση των καιρικών συνθηκών στους αριθμούς συνολικών επισκεπτών του 2018 σε επίπεδο ώρας. Η δειγματοληψια έχει προκύψει από τα επιμέρους 5λεπτα.

7.2.7.5 Η τεχνική του κινητού μέσου για χρονοσειρές ως αποτέλεσμα της ανάλυσης δεδομένων

Στην ενότητα 7.2.7.1 χρησιμοποιήσαμε περιόδους δειγματοληψίας των 45 λεπτών προκειμένου να έχουμε μία καθαρή εποπτεία της μακροσκοπικής μεταβολής των μεγεθών στο χρόνο. Στο συμπέρασμα αυτό καταλήξαμε διότι, σε επίπεδο 15 λεπτών (εικόνα 7.14), που είναι και το αντικείμενο μελέτης μας ήταν πολύ δύσκολο να γίνει καθαρή διάκριση των διαστημάτων αυξομείωσης των μεγεθών σε μια πιο μακροπρόθεσμη βάση.



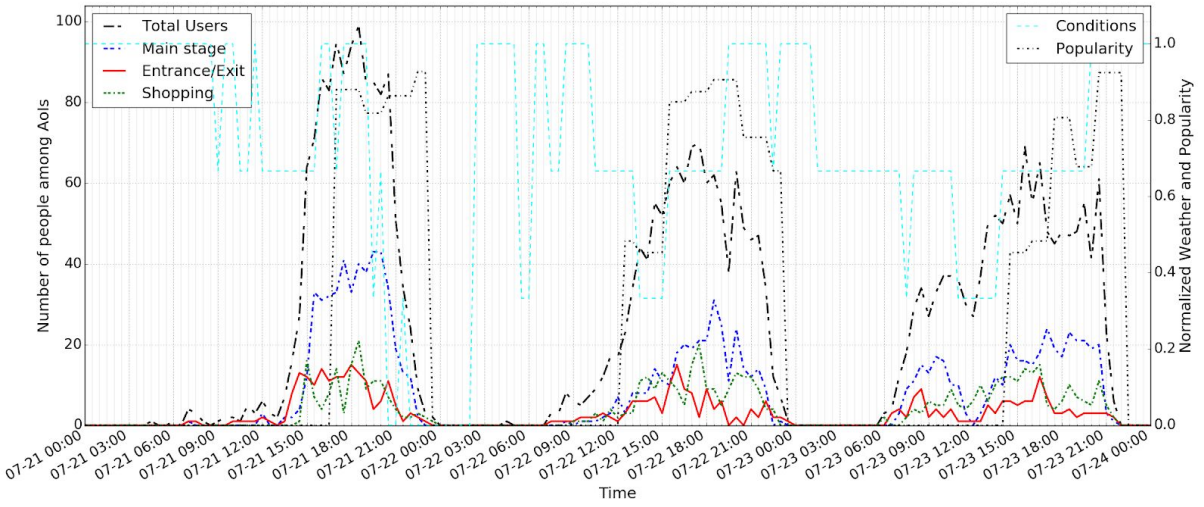
Εικόνα 7.16: Συνδυασμένες γραφικές παραστάσεις όλων των χαρακτηριστικών του προβλήματος σε timestep 15 λεπτών. Ο καιρός και ο δείκτης δημοτικότητας είναι κανονικοποιημένοι στο [0,1] ενώ τα πλήθη απεικονίζονται με τις απόλυτες τιμές τους.

Επιπλέον, τα δεδομένα του dataset timestep 45 λεπτών υπολογίστηκαν από τις μέσες τιμές των επιμέρους 5λέπτων. Ο υπολογισμός αυτός γίνεται με μία προσέγγιση κινητού μέσου ή rolling mean [81], η οποία αποτελεί μία ιδιαίτερα δημοφιλή προσέγγιση στον κλάδο της ανάλυσης χρονοσειρών (Time Series Analysis). Με την προσέγγιση αυτή προσπαθούμε να κατασκευάσουμε datasets περιόδων μεγάλης διάρκειας χρησιμοποιώντας τη μέση τιμή, των πολλών μικρότερων διαστημάτων που τα απαρτίζουν.

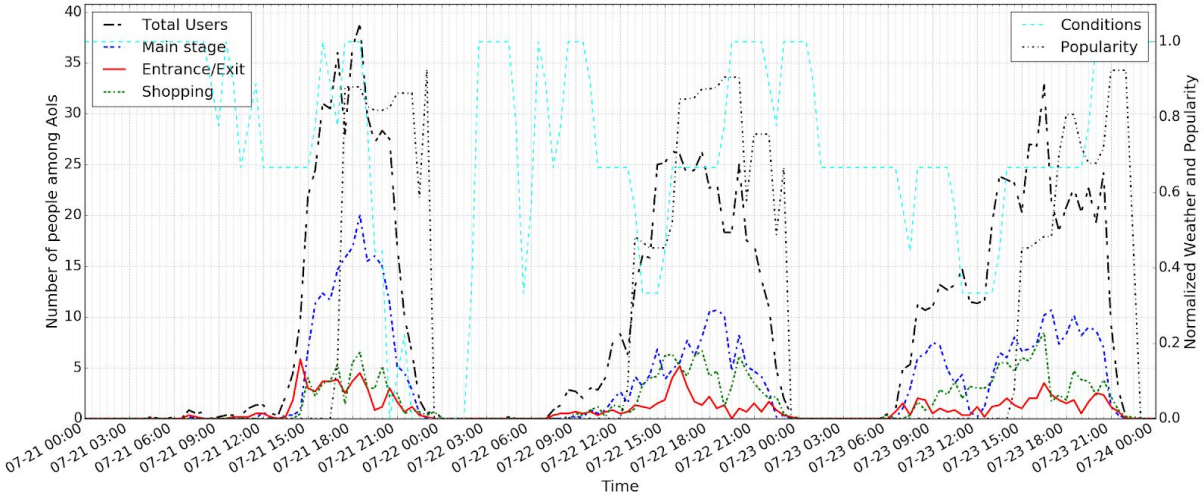
Το κύριο πλεονέκτημα της τεχνικής αυτής είναι ότι μπορούμε να εντάξουμε γρήγορες μεταβολές στα χρονικά πλαίσιο μεγάλης διάρκειας, οπότε έχουμε ένα πιο αντιπροσωπευτικό, λεπτομερές διάγραμμα. Για παράδειγμα ένας επισκέπτης οποίος μέσα σε ένα 45λεπτο αλλάζει τρεις φορές σημείο ενδιαφέροντος (κατά τη διάρκεια τριών διαφορετικών επιμέρους 5λεπτων του) μπορεί να προσμετρηθεί και στις τρεις αυτές περιοχές για το σύνολο του 45λεπτου.

Ένα μειονέκτημα της τακτικής αυτής είναι ότι αλλάζει τις τιμές των μεγεθών. Για παράδειγμα, ένα 30λεπτο dataset δειγματοληπτημένο από 5λεπτα (εικόνα 7.18), σε σχέση με ένα αυτούσια δειγματοληπτημένο 30λεπτο dataset (εικόνα 7.17), παρουσιάζει διαφορετικά μέγιστα στα επιμέρους πλήθη επισκεπτών σε κάθε AoI. Συνεπώς ένα τέτοιο γράφημα μπορεί να μας δώσει κυρίως ποιοτική πληροφορία.

Το γεγονός αυτό, στην πραγματικότητα, δε δημιουργεί πρόβλημα, δεδομένου ότι αντικείμενο μελέτης μας είναι η κατανομή των επισκεπτών ανάμεσα στις περιοχές ενδιαφέροντος και όχι οι αριθμητικές τιμές. Και οι κατανομές, αναλογίες και σχετικότητα των μεγεθών δεν επηρεάζονται με τη μέθοδο rolling mean. Επιπλέον είναι πολύ εύκολο να γίνει εκτίμηση του συνολικού αριθμού επισκεπτών στο φεστιβάλ ανά πάσα στιγμή μέσω ενός συστήματος ελέγχου εισόδου και εξόδου στην κεντρική πύλη.



Εικόνα 7.17: Συνδυασμένες γραφικές παραστάσεις όλων των χαρακτηριστικών του προβλήματος σε timestep 30 λεπτών.



Εικόνα 7.18: Συνδυασμένες γραφικές παραστάσεις όλων των χαρακτηριστικών του προβλήματος σε timestep 30 λεπτών. Η δειγματοληψια έχει προκύψει από τα επιμέρους 5λεπτα.

7.2.8 Η κατασκευή των τελικών dataset εισόδου των αλγόριθμων μηχανικής μάθησης

Στο αρχείο *papework2.py* γίνεται η κατασκευή των τελικών datasets εισόδου και εξόδου τα οποία θα χρησιμοποιηθούν από τους αλγόριθμους μηχανικής μάθησης. Σ' αυτό το script μας δίνεται η δυνατότητα να ορίσουμε μία σειρά από παραμέτρους σύμφωνα με τις οποίες θέλουμε να κατασκευάσουμε τα τελικά datasets. Οι παράμετροι αυτές αφορούν στον αριθμό των επιθυμητών περιοχών ενδιαφέροντος (AoIs-clusters), στον επιθυμητό αλγόριθμο clustering, στην επιθυμητή διάρκεια timestep, στην επιθυμητή χρόνια προς επεξεργασία. Επιπλέον, μας δίνεται η δυνατότητα να επιλέξουμε αν θέλουμε να εισάγουμε δεδομένα δημοτικότητας καλλιτεχνών και καιρού στο dataset μας. Στη συνέχεια, αν θέλουμε, μπορούμε να εφαρμόσουμε επιλεκτικά την τεχνική του moving average που είδαμε προηγουμένως. Το script αυτό επιτυγχάνει όλες αυτές οι λειτουργίες βασιζόμενο σε μεθόδους οι οποίες βρίσκονται κατά κόρον στο script με όνομα *defs_after.py*.

7.2.8.1 Περιγραφή λειτουργιών των επιμέρους μεθόδων

Στο σημείο αυτό περιγράφονται συνοπτικά οι εργασίες που επιτελούν οι μέθοδοι που χρησιμοποιούνται στο script papework2.py και συνεισφέρουν στη δημιουργία του τελικού dataset προς τα μοντέλα μηχανικής μάθησης.

- Η κλάση *clustersPeriods(cluster, minutes),* ανήκει στο script *defs_after..py* και περιλαμβάνει τις εξής 2 βασικές μεθόδους της υλοποίησης μας (τις οποίες και τροφοδοτεί με τις 2 βασικές μεταβλητές του αριθμού των clusters και των λεπτών ανα timestep):

    1. Η μέθοδος *assign_users_in_regions(df2 = "artists_list.csv", export_csv = False, plot = False, years = years, random_state = 42, clustering = 'kmeans')*
    η οποία δέχεται το dataset όπως προκύπτει από το αρχείου jsonFences.py (βλ. Πίνακας στο φιλτραρισμα), εφαρμόζει clustering της επιλογής μας στη χρονιά της επιλογής μας και το επιστρέφει με μια ακόμα στήλη η οποία ορίζει για κάθε γραμμή το όνομα της AoI (cluster) στην οποία ανήκει. Επιπλέον επιστρέφει το ίδιο αποτέλεσμα για το cluster στο οποίο ανήκει η σκηνη που εμφανίζεται κάθε καλλιτέχνης.

$oid	ID	Epoch	Timestamp	Date	Time	X	Y	Cluster
597188dac1acdf13a82dd456	9e3a954d-550a-423c-b6da-42bfd61a7bbb	1500612618	2017-07-21 04:50:18	2017-07-21	04:50:18	8.3716083	48.9980785	A
597248c2c1acdf13a82debfc	9e3a954d-550a-423c-b6da-42bfd61a7bbb	1500612618	2017-07-21 04:50:18	2017-07-21	04:50:55	8.3716083	48.9980790	A

Πίνακας 7.9: Το dataset στο ενδιάμεσο στάδιο μετά την επεξεργασία από την assign_users_in_regions

Artist	Buhne	Date	Timestamp	X	Y	Cluster
Meute	Hauptbuhne	2017-07-21	2017-07-21 23:00:00	8.37376284	48.99776582	E
Mars of Illyricum	Feldbuhne	2017-07-21	2017-07-21 20:00:00	8.37221654	49.00138549	B

Πίνακας 7.10: Το αρχείο προγράμματος καλλιτεχνών μετά την επεξεργασία από την assign_users_in_regions

    2. Η μέθοδος cluster_distribution_per_period(data, min_users_per_period):
    Η μέθοδος αυτή ομαδοποιεί τα στίγματα στις περιόδους προεπιλεγμένης διάρκειας, οπότε πλέον ξεχνάμε τη λογική των Timestamps και περνάμε σε timesteps.

- Το script final1_tune.py περιλαμβάνει τη μέθοδο
final1_create_and_tune(data=data, pois_to_clusters, t11, t12, t21, t22,artist_metrics, cluster, minutes, only_start, years)

Η μέθοδος αυτή συνδυάζει όλες τις μεθόδους που έχουμε δει ως τώρα και καταλήγει στην πιο πλήρη μορφή του τελικού dataset όπως θα τη δούμε παρακάτω. Επιπλέον επιστρέφεται ένα διάνυσμα με τα ονόματα των περιοχών στις οποίες υπάρχουν live, πράγμα πού χρήσιμο δεδομένου ότι τα ονόματα των AoIs μεταβάλλονται από τρέξιμο σε τρέξιμο καθιστώντας έτσι δύσκολη την εποπτεία του προβλήματος.

- Η μέθοδος *weather,* η οποία βρίσκεται στο script *weather.py* και καλείται προκειμένου να προσθέσει με τις σειρά της τις δύο στήλες καιρού, προσαρμοσμένες στη μέχρι τώρα δομή του dataset.

- Τελικά, χρησιμοποιείται προαιρετικά και το script *theregulator.py* στο οποίο με τη μέθοδο *theregulator(final2, minutes, multiplier, clwithlive)* γίνεται υλοποίηση του κινητού μέσου της ενότητα . Η μέθοδος χρειάζεται ως ορίσματα το αρχικό dataset μικρού timestep, το timestep του έναν ακέραιο πολλαπλασιαστή του timestep της αρέσκειας μας και τα ονόματα των AoIs στις οποίες υπάρχει live και άρα αντίστοιχη στήλη με κατάληξη pop. Για παράδειγμα είναι δυνατόν να κατασκευάσουμε ένα dataset χρονικής περιόδου διάρκειας 45 λεπτών παίρνοντας κάθε ένα από τα 9 πεντάλεπτα που το απαρτίζουν και υπολογίζοντας τις μέσες τιμές τους στο σύνολο του 45λεπτου. Η χρησιμότητα αυτής της μεθόδου έγινε εμφανής κυρίως στο κομμάτι της ανάλυσης δεδομένων.

7.2.8.2 Η δομή των τελικών δεδομένων εισόδου

Στον πίνακα 7.11, βλέπουμε ενδεικτικά τη δομή ενός παραδείγματος dataset εισόδου .Όπως προαναφέρθηκε μπορούμε να χρησιμοποιήσουμε διανύσματα πρόγνωσης για τους δείκτες δημοτικότητας και τον καιρό, εξού και ο δείκτης (t+1). Κρίσιμη είναι η σημασία του δείκτη χρονικής περιόδου, ο οποίος απλώς παρέχει ένα αύξοντα αριθμό αναπαράστασης κάθε timestep ξεκινώντας σε κάθε περίπτωση από τα ξημερώματα παρασκευής και μετρώντας timesteps εώς τα μεσάνυχτα Κυριακής.

A(t)	B(t)	C(t)	D(t)	E(t)	F(t)	Apop(t+1)	Bpop(t+1)	Tot(t)	Temp(t)	Cond(t)	Time index(t)
9	5	10	0	7	2	0.755	0.822	33	27	3	83
11	4	7	2	4	6	0.755	0.822	34	26	3	84
9	5	8	4	4	3	0.755	0.822	33	26	3	85
8	7	3	1	6	6	0.755	0.822	31	25	3	86

Πίνακας 7.11: Ενδεικτικός πίνακας όλων των χαρακτηριστικών εισόδου

7.2.9 Η κατασκευή των προβλημάτων και των μοντέλων πρόβλεψης ταξινόμησης και παλινδρόμησης

Χρησιμοποιήσαμε το αρχείο papework2.py, όπως είδαμε στην προηγούμενη ενότητα, τροποποιώντας τις παραμέτρους των μεθόδων και κατασκευάζοντας διαφορετικά datasets της μορφής του πίνακα 7.11. Οι διαφοροποιήσεις στα datasets αφορούσαν στη διάρκεια των timesteps, στον

αριθμό των clusters, στον αριθμό χρονικών περιόδων ολίσθησης της πρόγνωσης του καιρού και της δημοτικότητας των καλλιτεχνών, ακόμα και στους εναλλακτικούς τρόπους μοντελοποίησης και κανονικοποίησης των μεταβλητών μας. Εργαστήκαμε σε μια σειρά από προβλήματα παλινδρόμησης (regression) αλλά και ταξινόμησης (classification), αναζητώντας τον πιο αποτελεσματικό τρόπο αναπαράστασης της επιθυμητής εξόδου, με σκοπό τα μοντέλα μας να είναι αποτελεσματικά και οι προβλέψεις μας να είναι χρήσιμες.

Οι δοκιμές μας πραγματοποιήθηκαν στα scripts *single_output_classifiers.py, single_output_regressors.py*. Κάθε script μπορεί να εκτελεστεί μέσω terminal. Παρακάτω βλέπουμε ένα παράδειγμα εκτέλεσης:

& python single_output_classifiers.py -d 15_15min -s 1

όπου:

-d DIR, --dir DIR

-s SHIFTED, --shifted SHIFTED

Η παράμετρος SHIFTED παίρνει ακέραιες τιμές και μεταφράζεται ως η χρονική ολίσθηση στο μέλλον των μεταβλητών δημοτικότητας και καιρού ωστε να εισαχθούν στο dataset με μορφη προγνωσης.

Η παράμετρος DIR ορίζει το υποφάκελο του φακέλου datasets από τον οποίον θα ληφθεί το dataset προς μελέτη και στον οποίο θα εναποτεθεί στη συνέχεια το .csv αρχείο αποτελεσμάτων. Στο φάκελο /datasets του repository μπορει να βρει κανείς μια πληθώρα από προκατασκευασμένα datasets στα οποία μπορούν να τρέξουν τα παραπάνω scripts. Η ονοματοδοσία τους X_Ymin μεταφράζεται ως εξής ως φάκελος datasets με timestep X λεπτών δειγματοληπτημένο από επιμέρους timesteps Y λεπτών με την τεχνική moving average.

Επιπλέον, για λόγους προσαρμογής στις ανάγκες πρόβλεψης του φεστιβάλ, και για αποφυγή επαναληψιμότητας, δεδομένου ότι τα συμπεράσματα επεκτείνονται αναλόγως στα ποικίλα datasets, παρουσιάζουμε αναλυτικά τις εργασίες, τα μοντέλα μηχανικής μάθησης και τα αποτελέσματα της εκπαίδευσης, βελτιστοποίησης και πρόβλεψης, όπως αυτά διαρθρώθηκαν πάνω σε ένα dataset εισόδου με διάρκεια timestep 15 λεπτών, 6 στο πλήθος clusters και χρονική ολίσθηση (time shift) στο μέλλον ενός timestep για τις προγνώσεις καιρού και δημοτικότητας. Επισημαίνεται ότι η αναλογία training-test set κυμάνθηκε στο 75-25%.

7.2.9.1 Συνδυασμοί χαρακτηριστικών εισόδου στα δύο προβλήματα

Ως προς τα χαρακτηριστικά του προβλήματος, έγιναν προσπάθειες ελάττωσης τους και επιλογής των καλύτερων, όπως μπορεί κανείς να διαπιστώσει στο script *regression_classification/extras/PCA_mpelos.py,* όπου έγινε χρήση τεχνικής PCA του Dimensionality Reduction. Επιπλέον έγινε χρήση της μετρικής f-value για το dataset παλινδρόμησης και mutual information για το dataset ταξινόμησης στην προσπάθεια αξιολόγησης των χαρακτηριστικών εισόδου, όπως θα δούμε στις επόμενες ενότητες.

Ωστόσο, σε πρακτικό επίπεδο, προτιμήσαμε, λόγω του μικρού πλήθους features, να κάνουμε δοκιμές σε όλους τους πιθανούς συνδυασμούς χαρακτηριστικών ή Combinations of features (CoF) με δεδομένα πάντα τη μεταβλητή χρόνου, τα διανύσματα κατανομών στις AoIs και του συνολικού αριθμού επισκεπτών προσθαφαιρόντας εναλλάξ τις μεταβλητές καιρικών συνθηκών και δημοτικότητας καλλιτεχνών. Ορίζουμε, λοιπόν, στον πίνακα 7.12 τους 4 εναλλακτικούς συνδυασμούς χαρακτηριστικών που χρησιμοποιήσαμε για να εκπαιδεύσουμε τόσο τα μοντέλα ταξινόμησης, όσο και παλινδρόμησης.

Συνδυασμος (CoF)	Χαρακτηριστικά εισόδου συνδυασμου
1	A, B, C, D, E, F, Total users, Time index
2	A, B, C, D, E, F, Total users, Time index, Apop, Bpop
3	A, B, C, D, E, F, Total users, Time index, Temp, Cond
4	A, B, C, D, E, F, Total users, Time index, Apop, Bpop, Temp, Cond

Πίνακας 7.12: Οι συνδυασμοί χαρακτηριστικών που εφαρμόστηκαν στις εργασίες επιτηρούμενης μάθησης

7.2.9.2 Το πρόβλημα ταξινόμησης

7.2.9.2.1 Ορισμός του προβλήματος

Αρχικά επιλέξαμε να κάνουμε την πρόβλεψη της μεταβολής πλήθους επισκεπτών σε κάθε AoI με χρήση τεχνικών ταξινόμησης για τα δεδομένα του 2017. Καταλήξαμε, λοιπόν, σε τρεις κλάσεις επιθυμητής εξόδου με τις εξής ετικέτες:

1. Equal: Σχεδόν ίδια συγκέντρωση επισκεπτών στην AoI τη χρονική στιγμή t+1 σε σχέση με τη χρονική στιγμή t. Η λέξη "σχεδόν" αναφέρεται σ' ένα κατώφλι μεταβολής της τάξης του ± 5% επί του συνόλου των επισκεπτών της χρονικής αυτής περιόδου (στήλη "Total users").
2. Plus: Αξιόλογη αύξηση του αριθμού επισκεπτών στην AoI κατά τη χρονική στιγμή t+1 σε σχέση με τη χρονική στιγμή t. (πάνω από 5% επι του συνόλου)
3. Minus: Αξιόλογη μείωση του αριθμού επισκεπτών στηνAoI κατά τη χρονική στιγμή t+1 σε σχέση με τη χρονική στιγμή t.

Με βάση τα παραπάνω και έχοντας υπόψη τον πίνακα δεδομένων εισόδου 7.11, τα δεδομένα εξόδου για το πρόβλημα ταξινόμησης θα έχουν τη μορφή του πίνακα. 7.13.

A(t+1)	B(t+1)	C(t+1)	D(t+1)	E(t+1)	F(t+1)
plus	minus	minus	plus	minus	plus
minus	plus	plus	plus	equal	minus
minus	plus	minus	minus	plus	plus
equal	minus	plus	equal	minus	minus

Πίνακας 7.13: Η επιθυμητή έξοδος του προβλήματος πρόβλεψης με χρήση παλινδρόμησης

Είναι σαφές ότι δε στοχεύουμε σε μία πρόβλεψη ακρίβειας, ωστόσο πρακτικά σε επίπεδο έτσι computing και ασφάλειας μία τέτοια πρόβλεψη παρέχει επαρκή πληροφορία για τις ανάγκες των διοργανωτών.

7.2.9.2.2 Επιλογή αλγόριθμων ταξινόμησης

Για την επίλυση του προβλήματος ταξινόμησης έγινε, χρήση και σύγκριση των εξής ταξινομητών της scikit:

1. Support Vector Classifier (SVC)
2. Random Forest Classifier
3. K-Nearest Neighbors Classifier (KNN)
4. Gaussian Naive-Bayes Classifier
5. Deep Network Classifier (DN)

7.2.9.2.3 Αξιολόγηση χαρακτηριστικών εισόδου

Με χρήση της μεθόδου SelectKBest[38], η οποία αξιολογεί τα χαρακτηριστικά με χρήση μετρικών, και της μετρικής mutual information για classification κατασκευάσαμε ένα πίνακα (πίνακας 7.14) ο οποίος δείχνει τη σημαντικότητα κάθε χαρακτηριστικού εισόδου για την πρόβλεψη της αντίστοιχης εξόδου. Οι περιοχές ενδιαφέροντος αντιστοιχίζονται στα ονόματα τους βάσει της εικόνας 7.11 με το χάρτη του φεστιβάλ. Όσο μεγαλύτερη η τιμή της μετρικής mutual information τόσο μεγαλύτερη η συσχέτιση μεταβλητής εισόδου με την αντίστοιχη μεταβλητή εξόδου κατά την πρόβλεψη.

Input | Output	A (κεντρ. σκηνή)	B (δευτερ. σκηνές)	C (εκτός)	D (εστίαση)	E (σταθμος)	F (είσοδος)
A	0.24	0.09	0.04	0.21	0.07	0.11
B	0.10	0.21	0.03	0.10	0.07	0.03
C	0.16	0.04	0.17	0.08	0.10	0.00
D	0.11	0.08	0.00	0.28	0.01	0.00
E	0.16	0.09	0.00	0.14	0.21	0.06
F	0.21	0.07	0.03	0.05	0.17	0.08
Apop	0.06	0.01	0.03	0.07	0.05	0.06
Bpop	0.08	0.00	0.00	0.00	0.11	0.00
Total users	0.22	0.14	0.08	0.20	0.17	0.16
Temperature	0.12	0.00	0.00	0.14	0.09	0.06
Conditions	0.11	0.03	0.06	0.00	0.00	0.07
Time index	0.13	0.12	0.08	0.10	0.12	0.10

Πίνακας 7.14: Η μετρική mutual information για τις μεταβλητές του προβλήματος ταξινόμησης

Παρατηρούμε ότι η διαγώνιος του αρχικού 6x6 τμήματος του πίνακα παίρνει τις μεγαλύτερες τιμές καθώς η πρόβλεψη της επόμενης κατάστασης σε κάθε AoI εξαρτάται κατ' εξοχήν από την τιμή της προηγούμενης χρονικής στιγμής στην ίδια AoI.

[38] *https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html*

Επιπλέον, παρατηρούμε ότι το cluster εισόδου-εξόδου (F) επηρεάζεται αρκετά από την κεντρική σκηνή (A) αλλά και από το E το οποίο σηματοδοτεί την άφιξη κόσμου στους σταθμούς τρένων και κατα προέκταση σε επόμενο χρόνο πλησιάζουν την κυρία είσοδο. Ισχύει και η αντίστροφη εξάρτηση. Αντίστοιχα βλέπουμε πως η είσοδος περιέχει αρκετή πληροφορία για την πρόβλεψη των επισκεπτών της κεντρικής σκηνής σε επόμενο χρόνο. Αυτό είναι αναμενόμενο καθώς οι επισκέπτες τείνουν να επισκέπτονται την κεντρική σκηνή αμέσως μετά την άφιξη τους. Εν γένει, παρατηρούμε ότι οι εξαρτήσεις μεταξύ εισόδου και εξόδου ως προς τις AoIs είναι αμφίδρομες.

Τα χαρακτηριστικά της δημοτικότητας και του καιρού τείνουν να παρουσιάζουν, δυστυχώς, μικρή συσχέτιση σύμφωνα με το δείκτη mutual information, ωστόσο εξετάζουμε τις επιδράσεις τους στα μοντέλα προβλέψεις στις επόμενες υποενότητες.

Τέλος, η μεταβλητή του χρόνου είναι εμφανές ότι παρουσιάζει μια σταθερή συσχέτιση με τις μεταβλητές εξόδου. Αυτό είναι απόλυτα φυσιολογικό, καθώς ο χρόνος τρέχει εξίσου για οποιαδήποτε μεταβλητή του προβλήματος και δεν αναμένεται να ερμηνεύει διαφορετικές περιοχές με συστηματικά διαφορετική επιτυχία. Ο αριθμός συνολικών επισκεπτών παρατηρείται επίσης να έχει σημαντική συσχέτιση με τις εναλλαγές των κλάσεων.

7.2.9.2.4 Εκπαίδευση, αξιολόγηση και βελτιστοποίηση μοντέλων

Αρχικά, σημειώνεται πως οι παραδοσιακοί αλγόριθμοι ML, σε αντίθεση με τους αλγόριθμους DL, δεν έχουν τη δυνατότητα, από μόνοι τους, να εκπαιδεύονται σε πολλές εξόδους ταυτόχρονα (A,B,C,D,E,F). Αυτό σημαίνει πως κάθε ταξινομητής οφείλει να εκπαιδεύεται πάνω σε μία συγκεκριμένη στήλη εξόδου κάθε φορά και κατ' επέκταση για 6 εξόδους των AoIs A,B,C,D,E,F χρειάζονται 6x4 διαφορετικά μοντέλα, 6 για κάθε είδος ταξινομητή. Όπως προαναφέραμε ο διαχωρισμός training και test set είναι της τάξης του 75-25%. Η μετρική στην οποία βασιστήκαμε για την εκπαίδευση και την αξιολόγηση των μοντέλων είναι αυτή της ευστοχίας, δηλαδή:

$$accuracy = \frac{\sigma\omega\sigma\tau\acute{\epsilon}\varsigma\ \pi\rho o\beta\lambda\acute{\epsilon}\psi\epsilon\iota\varsigma}{\sigma\upsilon\nu o\lambda\iota\kappa\acute{\epsilon}\varsigma\ \pi\rho o\beta\lambda\acute{\epsilon}\psi\epsilon\iota\varsigma} \qquad \text{(Εξίσωση 7.1)}$$

Στο *script single_output_classifiers.py* πραγματοποιείται η διαδικασία εντοπισμού του καλύτερου δυνατού μοντέλο για κάθε CoF σε κάθε cluster ξεχωριστά. Γίνεται χρήση ενός pipeline. Αρχικά πραγματοποιείται κανονικοποίηση των δεδομένων μας (feature scaling) μέσω ενός StandardScaler. Στη συνέχεια, για κάθε CoF και κάθε AoI πραγματοποιείται ένα grid search βέλτιστων παραμέτρων. Η μέθοδος grid search δοκιμάζει ένα σύνολο διαφορετικών συνδυασμών υπερπαραμέτρων σε κάθε εκτιμητή και επιλέγει βάσει μιας διαδικασίας k-folds cross validation (στην περίπτωση μας k=5) το βέλτιστο

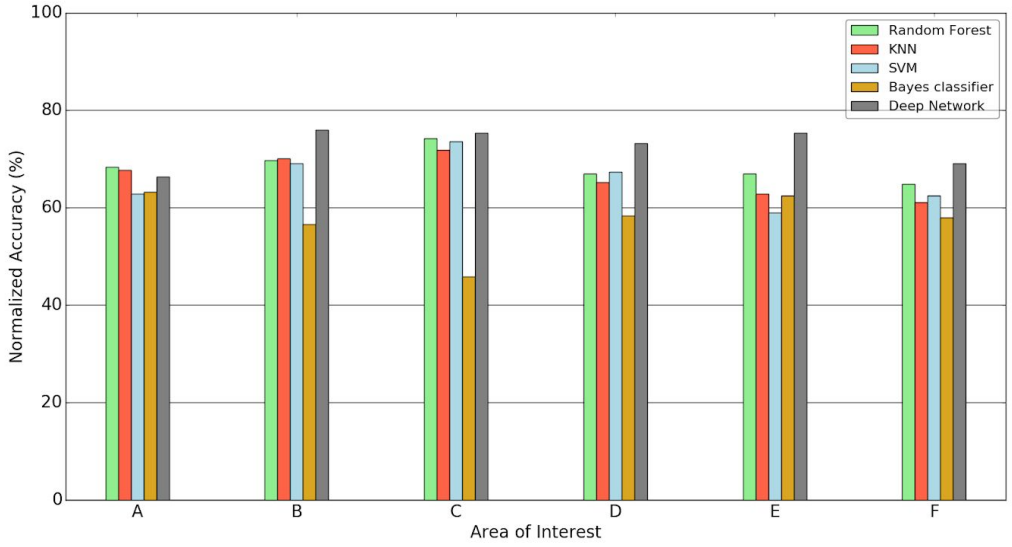
7.2.9.2.5 Παρουσίαση αποτελεσμάτων πρόβλεψης και παρατηρήσεις στα μοντέλα ταξινόμησης

Χρησιμοποιώντας τα βέλτιστα μοντέλα που προκύπτουν για κάθε εκτιμητή στο test set, κάνουμε προβλέψεις για τα διάφορα ζεύγη CoF και AoIs. Είναι εμφανές ότι μας χρειάζεται ένας πίνακας 3 διαστάσεων (πλήθος ταξινομητών x πλήθος AoIs x πλήθος CoF = 5 x 6 x 4) για να αποδώσουμε όλα τα ποσοστά ευστοχίας των προβλέψεων. Για το λόγο αυτό παρουσιάζουμε 2 υποπίνακες με τις μέσες ευστοχίες κάθε εκτιμητή ανα AoI (πιν. 7.15) και ανα CoF (πιν. 7.16) και τα αντιστοιχα ραβδογράμματα (εικ. 7.19, 7.20). Παρουσιάζουμε, επιπλέον, ένα λεπτομερές ραβδόγραμμα (εικόνα 7.21) των συνόλου των αποτελεσμάτων το οποίο κάνει πολύ ευκολότερη την

εξαγωγή παρατηρήσεων και συμπερασμάτων. Τα παρακάτω κατασκευάζονται με βάση το dataset 15λεπτου στο φάκελο *regression_classification\results and barcharts example\classification* του repository μέσω του script *barcharts_class.py*.

AoI | Estimator	Random Forest	KNN	SVM	GN Bayesian	Deep Network	Average
A	68.40	67.71	62.85	63.20	66.32	65.69
B	69.79	70.14	69.10	56.60	76.04	68.33
C	74.31	71.87	73.61	45.83	75.35	68.19
D	67.01	65.28	67.36	58.33	73.26	66.25
E	67.02	62.85	59.03	62.50	75.35	65.35
F	64.93	61.11	62.50	57.98	69.10	63.12

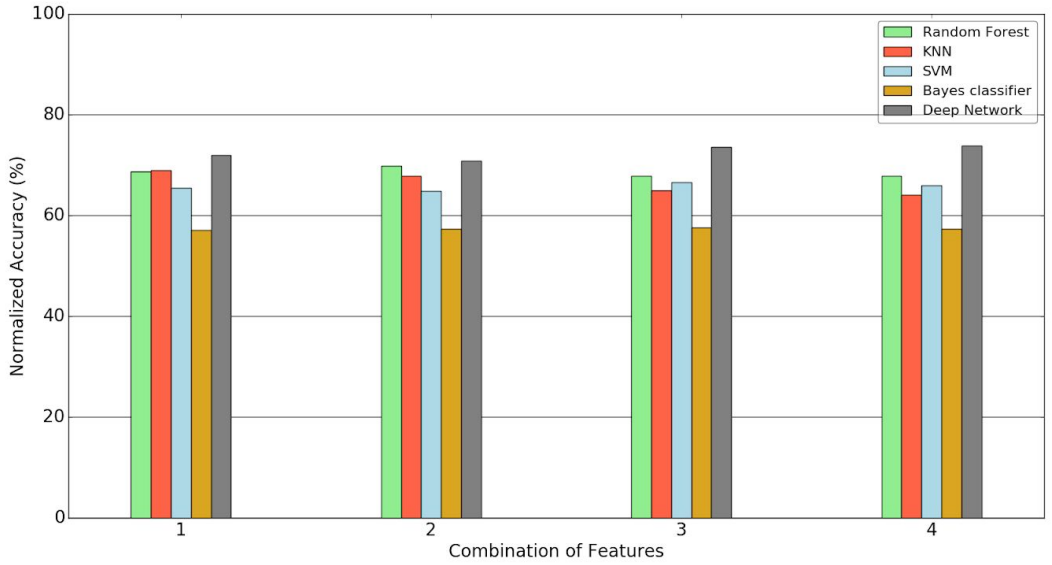
Πίνακας 7.15: Μέσες ευστοχίες (%) κάθε ταξινομητή ανά AoI



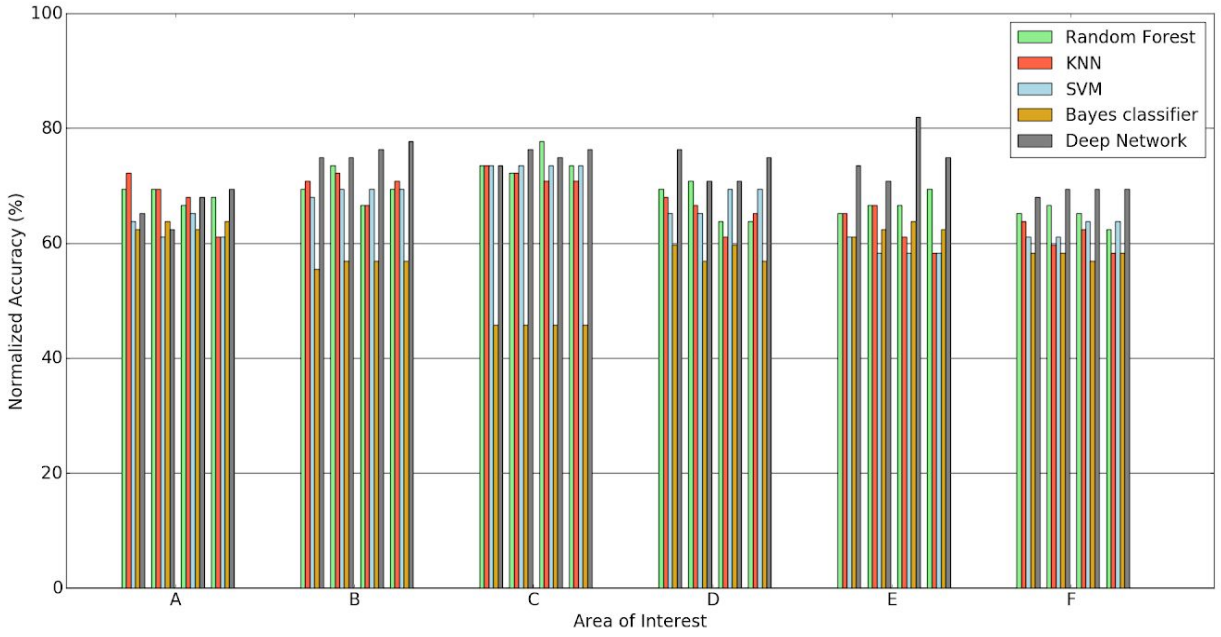
Εικόνα 7.19: Ραβδόγραμμα για τις μέσες ευστοχίες (%) κάθε ταξινομητή ανά AoI

CoF | Estimator	Random Forest	KNN	SVM	Bayes classifier	Deep Network	Average
1	68.75	68.98	65.51	57.18	71.99	66.48
2	69.91	67.82	64.81	57.41	70.83	66.16
3	67.83	65.05	66.67	57.64	73.61	66.16
4	67.82	64.12	65.97	57.41	73.84	65.83

Πίνακας 7.16: Μέσες ευστοχίες (%) κάθε ταξινομητή ανά CoF



Εικόνα 7.20: Ραβδόγραμμα για τις μέσες ευστοχίες (%) κάθε ταξινομητή ανά CoF



Εικόνα 7.21: Λεπτομερές ραβδόγραμμα όλων των αποτελεσμάτων αναλυτικά σε όλες τις AoIs για κάθε CoF. Τα CoF παρουσιάζονται σειριακά σύμφωνα με τον ήδη ορισμένο αύξοντα αριθμό τους.

Κατα τη σύγκριση των αποτελεσμάτων, παρατηρούμε τα εξής για τους ταξινομητές:

- Random Forest: Πετυχαίνει με σταθερότητα καλά αποτελέσματα χωρίς να επηρεάζεται από τους συνδυασμούς χαρακτηριστικών και την περιοχή-στόχο. Σημειώνεται πως οι συνδυασμοί παραμέτρων προς το GridSearch, έλαβαν σοβαρά υπόψη τους την τάση για overfitting του Random Forest και φροντίζουν να συμπεριλάβουν διάφορα είδη κλαδεμάτων (prunning) για την εύρεση του καλύτερου δυνατού μοντέλου.
- KNN: Παρουσιάζει σχετικά υψηλές ευστοχίες, ωστόσο μπορούμε να παρατηρήσουμε μια μείωση κατα την προσθήκη, όλο και περισσότερων χαρακτηριστικών στο πρόβλημα και ιδιαίτερα όταν σε αυτές υπεισέρχεται ο καιρός. Αυτό είναι αναμενόμενο λόγω του απλοϊκού χαρακτήρα του KNN, ο οποίος τον καθιστά αδύναμο σε περίπλοκες εξαρτήσεις. Παρ' όλα

αυτά, είναι αδιαμφισβήτητη η συμβολή του ταξινομητή KNN στην πρόβλεψη της κεντρικής σκηνής (A) που είναι και η σημαντικότερη περιοχή ενδιαφέροντος.

- ❖ SVM: Παρατηρούμε επίσης μια σταθερή απόδοση για όλους τους συνδυασμούς χαρακτηριστικών. Αυτό είναι λογικό καθώς ο SVM μπορεί να αντιμετωπίσει πολυπλοκότερες δομές χαρακτηριστικών, χωρίς να επηρεάζεται η απόδοση του. Αντιθέτως καταφέρνει να αξιοποιήσει σε ένα βαθμό κάποιες πληροφορίες με την προσθήκη των δεδομένων καιρού και δημοτικότητας.
- ❖ Gaussian Naive Bayes: Γενικώς αποτυγχάνει αρκετά να πετύχει καλά αποτελέσματα. Αυτό λογικά οφείλεται στην υπόθεση της ανεξαρτησίας μεταξύ χαρακτηριστικών στην οποία βασίζεται, εξού και ο όρος Naive. Στην περίπτωση μας η υπόθεση αυτή καταστρατηγείται πλήρως εξαιτίας της έντονης αλληλεξάρτησης μεταξύ των χαρακτηριστικών εισόδου.
- ❖ Deep Network: Το νευρωνικό δίκτυο το κατασκευάσαμε ξεχωριστά στο script *regression_classification\DL Models\DLClassification.py* . Διαρθρώνεται σε 5 layers, όπου κάθε hidden layer αποτελείται από 100 νευρώνες και ένα softmax layer στην έξοδο. Επιπλέον, προσπαθήσαμε να αποφύγουμε το overfitting με χρήση μεγάλου dropout. Το νευρωνικό δίκτυο, νικάει κατά κράτος τους υπόλοιπους ταξινομητές ενώ αυξάνει ελαφρώς την απόδοση του, με την προσθήκη όλων των features.

Επιπλέον παρατηρήσεις:

- ❑ Η περιοχή C, η οποία όπως είδαμε βρίσκεται εκτός φεστιβάλ εμφανίζει καλά ποσοστά και αυτό είναι αναμενόμενο διότι οι μεταβολές εκεί είναι ιδιαίτερα αργές και αδιάφορες. Οι αλγόριθμοι εκεί ενδεχομένως υιοθετούν κάποια λογική dummy classifier[39] κατα την εκπαίδευση. Οι dummy classifiers απλώς παράγουν προβλέψεις σύμφωνα με τη συχνότητα εμφάνισης των κλάσεων στο dataset. Η λογική αυτή αποφέρει καρπούς, δεδομένου ότι το πλήθος επισκεπτών εκεί είναι διαρκώς 0 ή παίρνει τιμές κοντά στο 0.
- ❑ Σε γενικές γραμμές, γίνεται εμφανής η αδυναμία των εξωτερικών χαρακτηριστικών (καιρός δημοτικότητα) να βελτιώσουν τις προβλέψεις των μοντέλων μας κατα μέσο όρο. Ειδικά για τον καιρό αυτό ήταν κάπως αναμενόμενο καθώς όπως αναφέρθηκε είχε πιο μακροπρόθεσμη επιρροη στην παρουσία κόσμου. Ωστόσο, ορισμένοι από τους αλγόριθμους ατομικά, όπως είδαμε, καταφέρνουν να αποσπάσουν τις λιγοστές πληροφορίες που αυτές προσφέρουν.
- ❑ Η AoI της κεντρικής σκηνής παρουσιάζει μέτρια αποτελέσματα, πράγμα που φαντάζει απογοητευτικό, ωστόσο στην επόμενη ενότητα γίνεται μια επιτυχημένη προσπάθεια βελτίωσης τους με μεθόδους παλινδρόμησης.

7.2.9.3 Το πρόβλημα παλινδρόμησης

7.2.9.3.1 Ορισμός του προβλήματος παλινδρόμησης

Συνεχίζοντας την ανάλυση μας, έγινε προσπάθεια να επέκτασης των προβλέψεων με χρήση τεχνικών παλινδρόμησης,. Στόχος είναι πλέον η πρόβλεψη του πραγματικού αριθμού των συγκεντρώσεων επόμενων στιγμών των επισκεπτών ανάμεσα στις AoIs.

Η επιθυμητή έξοδος για ένα πρόβλημα παλινδρόμησης πρόβλεψης του επόμενου timestep, είναι δυνατόν να κατασκευαστεί με μια απλή ολίσθηση των κατανομών εισόδου στο χρόνο (Πιν. 1).
Το script παλινδρόμησης είναι το *single_output_regressors.py* του φακέλου /*regression_classification* και οι λειτουργίες του περιγράφονται αναλυτικα στις επόμενες ενότητες.

[39] *https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html*

A(t+1)	B(t+1)	C(t+1)	D(t+1)	E(t+1)	F(t+1)
11	4	7	2	4	6
9	5	8	4	4	3
8	7	3	1	6	6
8	4	8	1	2	4

Πίνακας 7.17: Η επιθυμητή έξοδος του προβλήματος πρόβλεψης για με χρήση ταξινόμησης

7.2.9.3.2 Επιλογή αλγόριθμων παλινδρόμησης

Για την επίλυση του προβλήματος παλινδρόμησης έγινε, χρήση και σύγκριση των εξής εκτιμητών της scikit:

1. Support Vector Regressor (SVR)
2. Random Forest Regressor
3. K-Nearest Neighbors Regressor (KNN)
4. Kernel Ridge Regressor (GNB)
5. Deep Network Regressor (DN)

7.2.9.3.3 Αξιολόγηση χαρακτηριστικών εισόδου

Ξανά με χρήση της μεθόδου SelectKBest και της μετρικής f-value για regression η οποία προκύπτει από διαδικασίες ανάλυσης διασποράς (ANOVA) [41] κατασκευάσαμε ένα πίνακα (πίν. 7.18) ο οποίος δείχνει τη σημαντικότητα κάθε χαρακτηριστικού εισόδου για την πρόβλεψη της αντίστοιχης εξόδου. Το f-value παίρνει τιμές στο [0,1] με το 1 να συμβολίζει τη δυνατότητα ενός χαρακτηριστικού να διαχωρίζει πλήρως την έξοδο σε διακριτές, ως προς μέση τιμή και διασπορά, ομάδες δειγμάτων.

Input | Output	A (κεντρ. σκηνή)	B (δευτερ. σκηνές)	C (εκτός)	D (εστίαση)	E (σταθμος)	F (είσοδος)
A	0.72	0.28	0.43	0.45	0.42	0.40
B	0.33	0.57	0.24	0.53	0.40	0.26
C	0.34	0.34	0.44	0.34	0.31	0.28
D	0.42	0.54	0.25	0.57	0.52	0.27
E	0.44	0.46	0.35	0.57	0.52	0.31
F	0.33	0.20	0.30	0.32	0.35	0.34
Apop	0.22	0.25	0.19	0.24	0.22	0.24
Bpop	0.18	0.24	0.21	0.34	0.21	0.24
Total users	0.77	0.53	0.51	0.67	0.54	0.55
Temperature	0.32	0.31	0.24	0.34	0.29	0.29
Conditions	0.37	0.11	0.15	0.14	0.10	0.26
Time index	0.46	0.39	0.45	0.49	0.48	0.47

Πίνακας 7.18: Η μετρική mutual information για τις μεταβλητές του προβλήματος ταξινόμησης

Παρατηρούμε, όπως και στην ταξινόμηση, ότι η διαγώνιος του αρχικού 6x6 τμήματος του πίνακα παίρνει τις μεγαλύτερες τιμές καθώς η πρόβλεψη της επόμενης κατάστασης σε κάθε AoI εξαρτάται κατ' εξοχήν από την τιμή της προηγούμενης χρονικής στιγμής στην ίδια AoI.

Οι παρατηρήσεις είναι ανάλογες με πριν σε μεγάλο βαθμό, ωστόσο είναι σημαντικό να αναφέρουμε πως πλέον ο ρόλος της χρονικής μεταβλητής είναι καθοριστικής σημασίας για την πρόβλεψη όλων των περιοχών με παρόμοιες βαρύτητες στην καθεμιά. Αυτό είναι λογικό καθώς η ώρα της ημέρας έχει πολύ μεγάλη σημασία για τις αριθμητικές τιμές επισκεπτών στις περιοχές, όπως είδαμε στη ενότητα 7.2.7 Το ίδιο ισχύει και για τη μεταβλητή Total users αφού η συγκέντρωση σε κάθε AoI συναρταται άμεσο με το συνολικό αριθμό επισκεπτών.

7.2.9.3.4 Εκπαίδευση, αξιολόγηση και βελτιστοποίηση μοντέλων

Ο διαχωρισμός σε training και test set είναι της τάξης του 75-25% και στην περίπτωση παλινδρόμησης. Κατά το στάδιο της εκπαίδευσης οι αλγόριθμοι εκπαιδεύτηκαν στην ελαχιστοποίηση της μετρικής του μέσου τετραγωνικού σφάλματος (mse) της εξίσωσης 4.9 για κάθε AoI ξεχωριστά. Στη συνέχεια βελτιστοποιήθηκαν με GridSearch και 5-folds cross validation. Για την αξιολόγηση των μοντέλων χρησιμοποιήθηκε το μέσο απόλυτο σφάλμα (mae) της εξίσωσης 4.10 για λόγους διαισθητικής επαφής και αντιστοιχίας με τις μονάδες του προβλήματος Ωστόσο, κάθε AoI, ανάλογα με την επισκεψιμότητα της παρουσιάζει διαφορετική δυναμική. Συνεπώς τα σφάλματα-μετρικές αξιολόγησης δε θα μπορούσαν να δίνονται σε μονάδες επισκεπτών για κάθε περιοχή. Πιο συγκεκριμένα, ένα σφάλμα 10 ατόμων σε μία πρόβλεψη της τάξης των 1000 ατόμων δεν είναι δυνατόν να έχει ίδια βαρύτητα με ένα σφάλμα 10 ατόμων σε μία πρόβλεψη της τάξης των 20. Στη μία περίπτωση η απόκλιση είναι 1%, στην άλλη 50% παρ' ότι το σφάλμα mae της πρόβλεψης είναι το ίδιο και κατα προέκταση. Κατασκευάσαμε, λοιπόν μια νέα σχετική μετρική αξιολόγησης:

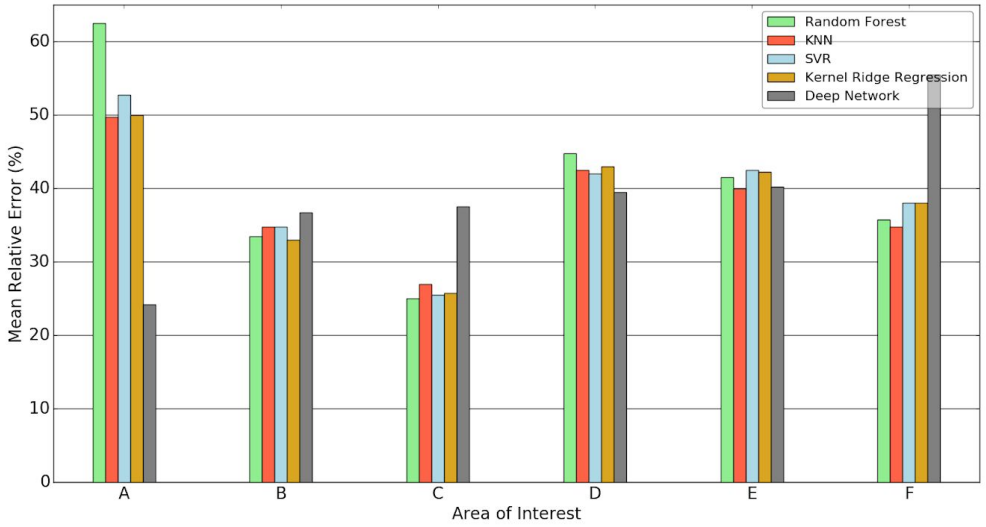
$$MRE(AoI) = Mean\ Relative\ Error(AoI) = \frac{MAE\ (AoI)}{Mean(AoI)} \qquad \text{(Εξίσωση 7.2)}$$

7.2.9.3.5 Παρουσίαση αποτελεσμάτων πρόβλεψης και παρατηρήσεις

Όπως και στην περίπτωση ταξινόμησης, παρουσιάζουμε τα MRE κάθε εκτιμητή ανα AoI (πιν. 7.19) και ανα CoF (πιν. 7.20) και τα αντιστοιχα ραβδογράμματα (εικ. 7.22, 7.23). Παρουσιάζουμε, επιπλέον, το λεπτομερές ραβδόγραμμα (εικ. 7.24) των συνόλου των αποτελεσμάτων. Τα παρακάτω κατασκευάζονται με βάση το dataset 15λεπτου για παλινδρόμηση στο φάκελο *regression_classification\results and barcharts example\regression* του repository μέσω του script *barcharts_regpy*.

AoI | Estimator	Random Forest	KNN	SVM	Ridge Regressor	Deep Network	Average
A	62.5	49.75	52.75	50	24.25	47.85
B	33.5	34.75	34.75	33	36.75	34.55
C	25	27	25.5	25.75	37.5	28.15
D	44.75	42.5	42	43	39.5	42.35
E	41.5	40	42.5	42.25	40.25	41.3
F	35.75	34.75	38	38	55.5	40.4

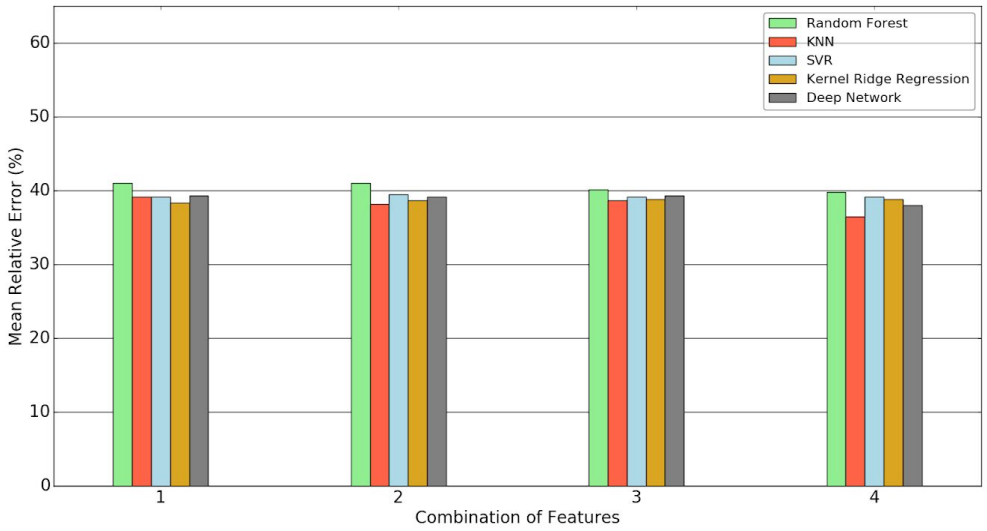
Πίνακας 7.19: Μέσα MRE (%) κάθε εκτιμητή ανά AoI



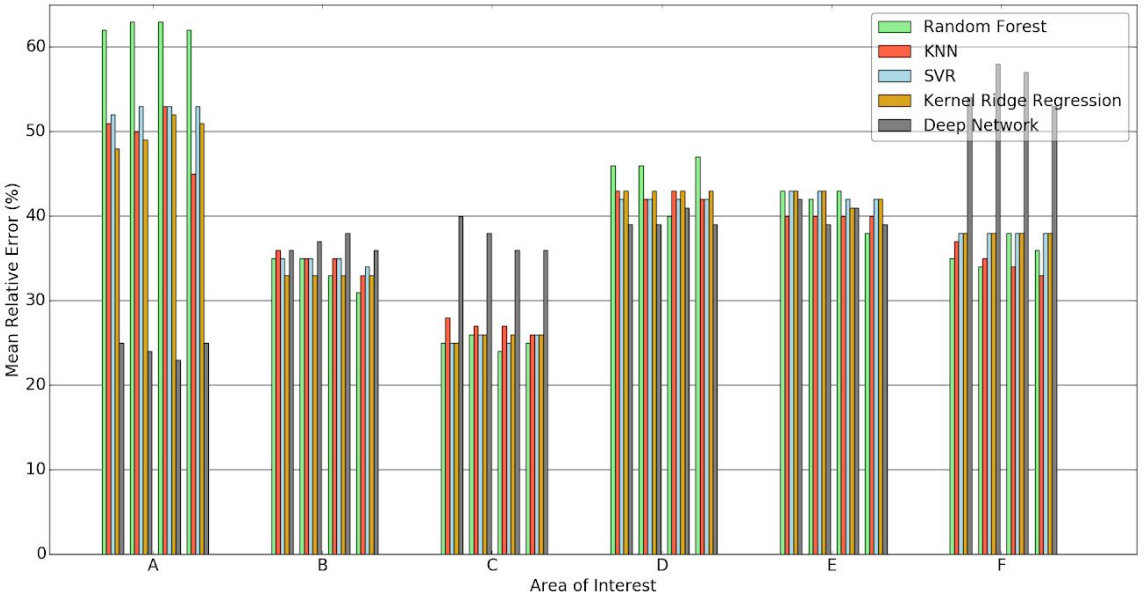
Εικόνα 7.22: Ραβδόγραμμα για τα μέσα MRE (%) κάθε εκτιμητή ανά AoI

CoF | Estimator	Random Forest	KNN	SVM	Ridge Regressor	Deep Network	Average
1	41.00	39.17	39.17	38.33	39.33	39.40
2	41.00	38.17	39.50	38.67	39.17	39.30
3	40.17	38.67	39.17	38.83	39.33	39.23
4	39.83	36.50	39.17	38.83	38.00	38.47

Πίνακας 7.20: Μέσες ευστοχίες (%) κάθε εκτιμητή ανά CoF



Εικόνα 7.23: Ραβδόγραμμα για τά μέσα MRE (%) κάθε εκτιμητή ανά CoF



Εικόνα 7.24: Λεπτομερές ραβδόγραμμα όλων των αποτελεσμάτων αναλυτικά σε όλες τις AoIs για κάθε CoF. Τα CoF παρουσιάζονται σειριακά σύμφωνα με τον ήδη ορισμένο αύξοντα αριθμό τους.

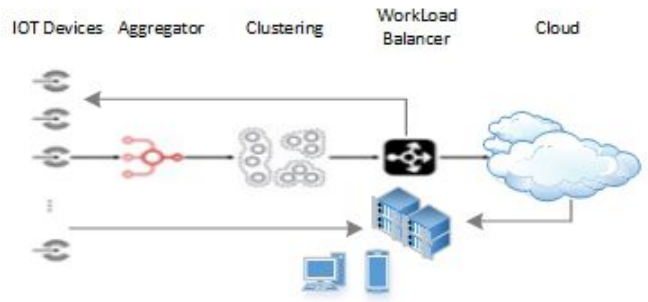
- Στο πρόβλημα παλινδρόμησης, κάναμε χρήση του ενός kernel ridge regression. Θεωρήσαμε πως ένας τέτοιος εκτιμητής είναι κατάλληλος για την περίπτωση μας, αφού, όπως είδαμε στην ενότητα 4.5.4, είναι κατάλληλος για να αντιμετωπίζει προβλήματα των οποίων τα χαρακτηριστικά εισόδου πάσχουν από multicollinearity. Για τους λόγους αυτους, παρουσιάζει μάλλον τα μικρότερα σφάλματα εκ των παραδοσιακών αλγόριθμων ML
- Σε ότι αφορά τις επιδόσεις των αλγόριθμων, θα έλεγε κανείς ότι υπάρχει μια σχετική αδιαφορία τους ως πρός την επιλογή συνδυασμού χαρακτηριστικών. Παρατηρείται ωστόσο μια μικρή βελτίωση όταν εντάσσονται όλα τα features (CoF4) .
- Χαρακτηριστική είναι η αδυναμία όλων των "παραδοσιακών" αλγόριθμων ML να προβλέψουν την AoI A, οι οποίες δίνουν "εξωπραγματικά" σφάλματα, πιθανώς λόγω της πολυπλοκότητας των εναλλαγών των επισκεπτών της. Το πρόβλημα αυτό ήρθε να λύσει, με τον καλύτερο τρόπο, ένα νευρωνικό δίκτυο 3 hidden layers 128 νευρώνων το καθένα και ενός ReLU layer 72 νευρώνων, το οποίο κάνει χρήση τόσο σιγμοειδών, όσο και γραμμικών συναρτήσεων ενεργοποίησης. Το νευρωνικό δίκτυο το κατασκευάσαμε ξεχωριστά στο script *regression_classification\DL Models\DLRegression.py.* Αντίστοιχα βέβαια, οι απώλειες είναι εμφανείς στο cluster F της εισόδου/εξόδου και στο C.
- Η περιοχή C, εμφανίζει κατα μέσο όρο μικρά σφάλματα λόγω της προβλεψιμότητας της.
- Οι προβλέψεις των εκτιμητών είναι αρκετά πιο ασταθείς, σε σχέση με την ταξινόμηση, όχι ώς προς τους συνδυασμούς χαρακτηριστικών, αλλά κυρίως ως πρός τις περιοχές ενδιαφέροντος. Αυτό το γεγονός πρέπει να ληφθεί σοβαρά υπόψη κατα την επιλογή μοντέλου, ανάλογα με τη φύση της AoI που μελετάμε. Από την άλλη, δεν είναι πρόβλημα πόσο μάλλον στην περίπτωση που βασιζόμαστε σε μια κατανεμημένη αρχιτεκτονική όπως θα δούμε στη συνέχεια.

7.3 Ένα προτεινόμενο σενάριο αρχιτεκτονικής Fog Computing

Σημαντικότερη συνεισφορά της εργασίας αυτής ήταν η ένταξη της έρευνας της στο κατατεθειμένο paper με τίτλο: "Prediction of the distribution of visitors for Large Events in Smart Cities" στο οποίο προτείνεται μια καινοτόμος Fog αρχιτεκτονική η οποία ενσωματώνει την πρόβλεψη τοποθεσίας με μεθόδους ML και DL και βρίσκει χρησιμότητα σε Large Events.

7.3.1 Pipeline της αρχιτεκτονικής

Στην ενότητα αυτή παρουσιάζουμε μια Fog αρχιτεκτονική η οποία έχει τη δυνατότητα εφαρμογής σε Large Events με σκοπό να εξυπηρετήσει ταυτόχρονα όλες τις ανάγκες που περιγράφηκαν στην ενότητα 7.1. Η δομή του pipeline της αρχιτεκτονικής παρουσιάζεται στην εικόνα 7.25.



Εικόνα 7.25: Δομή της προτεινόμενης Fog αρχιτεκτονικής

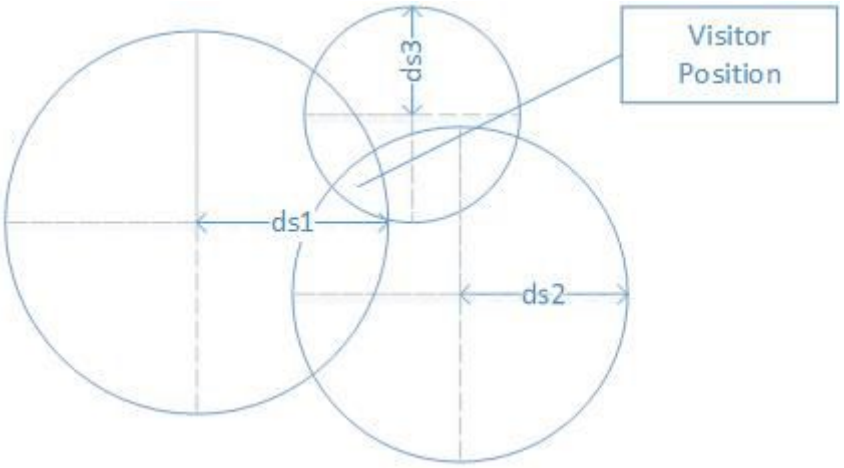
Υποθέτουμε πως βρισκόμαστε σε μία περιοχή του χώρου στην οποία πραγματοποιείται ένα Large Event όπως το DasFest. Στο χώρο βρίσκονται διάσπαρτα Edge Devices. Αυτά απαρτίζονται από IoT αισθητήρες, οι οποίοι λαμβάνουν το δείκτη (RSSI) έντασης του σήματος WiFi των κινητών συσκευών στο χώρο) και είναι συνδεδεμένοι με Raspberry Pis. Τα edge devices έχουν τους εξής δύο ρόλους στην αρχιτεκτονική αυτη:

1. Κάνουν εκτιμηση τη θέση των συσκευων τις οποίες ανιχνεύουν οι αισθητήρες μέσω μιας διαδικασίας Trilateration που περιγράφεται συνοπτικα σε επόμενη στην επόμενη ενότητα και αναλυτικά στο [82].
2. Διαθέτουν CPU και μνήμη ώστε να αναλαμβάνουν υπολογιστικό φόρτο

Στην αρχή του pipeline τα δεδομένα περνάνε από τον aggregator ο φιλτράρει και ανωνυμοποιεί τα δεδομένα. Στη συνέχεια ακολουθεί η διαδικασία clustering όπου τα δεδομένα ομαδοποιούνται βάσει των Timestamps τους και των "ανώνυμων" πλέον IDs τους. Το σημαντικότερο συστατικό (component) του pipeline αποτελεί ο Workload Balancer ο οποίος κατανέμει τον υπολογιστικό φόρτο (workload) ανάμεσα στο Edge και το Cloud ακολουθώντας μια πολιτική η οποία βασίζεται στα αποτελέσματα της πρόβλεψης κατανομής επισκεπτών, όπως αυτή παρουσιάστηκε στις προηγούμενες ενότητες. Συνοπτικά ο ρόλος του Workload Balancer είναι προφανώς να δίνει προτεραιότητα στο Edge, για λόγους που έχουν ήδη αναφερθεί, και στη συνέχεια, όταν υπάρχει πρόβλεψη αύξησης ζήτησης από ένα κατώφλι και πάνω, αιτείται τη δέσμευση VMs στο Cloud για το επιπλέον αναμενόμενο workload.

7.3.2 Συνοπτική περιγραφή της διαδικασίας Trilateration

Τα Edge devices υποτίθεται ότι προσφέρουν δωρεάν WiFi hotspot, με σκοπό οι IoT sensors να εντοπίζουν και να καταγράφουν τις εντάσεις σημάτων των συσκευών που αναζητούν WiFi θέτοντας μοναδικά MAC address based IDs. Η ένταση του σήματος, μπορεί να μεταφραστεί σε μία απόσταση της συσκευής από τις γνωστές συντεταγμένες του αισθητήρα μέσω της σχέσης. Η μέτρηση ενέχει κάποια απόκλιση δεδομένου ότι η ισχύς του σήματος εξαρτάται και από άλλες παραμέτρους, πέραν της απόστασης, όπως παρεμβαλλόμενα άτομα, αντικείμενα και η τρόπος με τον οποίο κρατάει τη συσκευή ο χρήστης. Σε ιδανικές συνθήκες, τρεις τέτοιες μετρήσεις από τους κοντινότερους στη συσκευή αισθητήρες θα μπορούσαν να ορίσουν επακριβώς τη θέση της στο χώρο, ωστόσο τα σφάλματα των μετρήσεων οδηγούν σε μια κατάσταση όπως αυτή της εικόνας 7.26, δηλαδή οι τρεις περιφέρειες δεν τέμνονται σε κοινό σημείο, αλλα ορίζουν μια μικρή περιοχή στην οποία κατα προσέγγιση κινείται ο χρήστης της συσκευής. Στη συνέχεια επιλέγεται ένα συγκεκριμένο σημείο με εφαρμογή τεχνικών ελαχιστοποίησης του σφάλματος που περιγράφονται στο [82].



Εικόνα 7.26: Εκτίμηση θέσης με Trilateration

7.3.3 Η πιθανή εφαρμογή του σεναρίου στο DasFest

Στο DasFest πλέον υπάρχουν τις υποδομές για να να εφαρμοστεί, από το 2019, η αρχιτεκτονική του σεναρίου της ενότητας 2.2 με 30 raspberry PIs τα οποία προσφέρουν επεξεργαστική ισχύ, παροχή δικτύου και συνεχή εντοπισμό τοποθεσίας, μέσω των IoT sensors, όλων των παρόντων συσκευών στο χώρο ανεξαρτήτως χρήσης του mobile app και σύνδεσης στο διαδίκτυο. Αναμένεται η απόκτηση datasets τα οποία όχι μόνο θα βοηθήσουν στο testing των ήδη υπαρχόντων μοντέλων αλλα και στην εκπαίδευση νέων και πολύ πιο ισχυρών μοντέλων πρόβλεψης κατανομής επισκεπτών με χρήση νευρωνικών δικτύων αλλα και με τεχνικές time series analysis καθώς το "παραδοσιακό" machine learning δεν ενδείκνυται πραγματικά για τέτοιου είδους προβλήματα.

7.4. Συμπεράσματα

Στο πρώτο σκέλος της διπλωματικής αυτής εργασίας, κάνουμε μία ανάλυση σε βάθος της επιστήμης της μηχανικής μάθησης παρουσιάζοντας τα είδη, τη διάρθρωση και μια σειρά από διαφορετικές και εφαρμογές της στον επιστημονικό και τον επιχειρησιακό κόσμο. Επιπλέον γίνεται μία εκτενής ανάλυση και θεμελίωση των αλγόριθμων και τεχνικών παραδοσιακής μηχανικής μάθησης.

Στη συνέχεια, καλούμαστε να κάνουμε χρήση των γνώσεων των πρώτων κεφαλαίων σε μια εφαρμογή πρακτικού ενδιαφέροντος, η οποία αφορά στα Large Events, ως αντικείμενο επιστημονικής έρευνας και μελέτης του κλάδου των Smart Cities. Όπως ήδη αναφέραμε, τα Large Events αφορούν σε μεγάλες συγκεντρώσεις πλήθους σε χώρους με σκοπό την παρακολούθηση κάποιου θεάματος. Οι ανάγκες για ασφάλεια και ικανοποίηση των επισκεπτών τα καθιστά ένα σημαντικό αντικείμενο μελέτης. Επιπλέον, έγινε εμφανής σημαντικότητα της εφαρμογής ολοκληρωμένων αρχιτεκτονικών Fog Computing σε τέτοιου είδους συμβάντα, με σκοπό την ομαλή λειτουργία των ηλεκτρονικών υπηρεσιών, και την άψογη εξυπηρέτηση των επισκεπτών.

Πιο συγκεκριμένα, το Large Event, το οποίο μελετήσαμε είναι το Das Fest στην Καρλσρούη της Γερμανίας. Παραλάβαμε από τους διοργανωτές, και πιο συγκεκριμένα από το project Basmati, ένα σύνολο δεδομένων (dataset), το οποίο περιέχει ένα σύνολο συλλεγμένων στιγμάτων τοποθεσίας των επισκεπτών του φεστιβάλ κατά τις ημέρες διεξαγωγής του για τις χρονιές 2017 και 2018. Η συλλογή δεδομένων πραγματοποιήθηκε μέσω της χρήσης του mobile app του φεστιβάλ.

Έχοντας στην κατοχή μας τα δεδομένα τοποθεσίας, προσπαθήσαμε να δομήσουμε μία σειρά από προβλήματα επιτηρούμενης μάθησης με σκοπό την πρόβλεψη των κατανομών επισκεπτών στις

επιμέρους περιοχές του χώρου πραγματοποίησης του φεστιβάλ. Ξεκινήσαμε μία διαδικασία ανάλυσης προεπεξεργασίας τους, προκειμένου να αποκτήσουν κατάλληλη μορφή για την τροφοδότηση τους σε αλγόριθμους μηχανικής μάθησης. Έγινε φιλτράρισμα των δεδομένων και διαχωρισμός τους σε περιόδους 15 λεπτης διάρκειας. Ακολούθησαν τεχνικές feature engineering κατα τις οποίες, αναζητήθηκαν δεδομένα τύπου open data σε σχέση με το ιστορικό του καιρού, αλλα και τη δημοτικότητα των καλλιτεχνών που παρουσιάστηκαν κατα τις ημέρες διεξαγωγής του φεστιβάλ. Τα δεδομένα αυτά εισήχθησαν ως χαρακτηριστικά εισόδου στα μοντέλα επιτηρούμενης μάθησης με την σκοπό βελτιώσουν τις προβλέψεις μας.

Σε επόμενο στάδιο, έγινε μία προσπάθεια διαχωρισμού του χώρου του φεστιβάλ σε επιμέρους περιοχές ενδιαφέροντος με χρήση τεχνικών συσταδοποίησης πάνω στα στίγματα τοποθεσίας των επισκεπτών. Ακολουθήθηκε αυτή η μεθοδολογία διότι τα σημεία ενδιαφέροντος που μας δόθηκαν από τους διοργανωτές του φεστιβάλ ήταν πάρα πολλά σε αριθμό με συνέπεια να έχουν πολύ μικρές αποστάσεις μεταξύ τους και να δυσχεραίνουν τη διαδικασία πρόβλεψης. Επιπλέον έγινε η λογική παραδοχή ότι οι περιοχές ενδιαφέροντος σε ένα Large Event είναι δυναμικά ορισμένες με βάση τη συμπεριφορά του κόσμου κατά τη διεξαγωγή του. Ο αλγόριθμος που επικράτησε για τις διαδικασίες συσταδοποίησης ήταν ο k-means. Αυτή η διαδικασία μας οδηγεί στο συμπέρασμα ότι πράγματι μελετώντας τα δεδομένα τοποθεσίας των επισκεπτών ενός Large Event είναι δυνατόν να παρατηρήσουμε τον τρόπο συγκέντρωσης και μετακίνησης τους στους επιμέρους χώρους και να κρίνουμε ποια πραγματικά είναι τα σημεία ενδιαφέροντος χωρίς να υπάρχει η ανάγκη για χρήση προεπιλεγμένων σημείων ενδιαφέροντος. Επιπλέον με χρήση αλγόριθμων συσταδοποίησης κεντροειδών, είναι πάρα πολύ εύκολο, αφού εντοπιστούν τα σημεία ενδιαφέροντος (PoIs), να οριστούν και συνεχή πολύγωνα περιοχών ενδιαφέροντος (AoIs). Εν γένει, οι τεχνικές συσταδοποίησης φάνηκε να είναι πολύ χρήσιμες και πλούσιες σε πληροφορία σε ότι αφορά σε προβλήματα γεωχωρικής μελέτης της συμπεριφοράς των επισκεπτών ενός Large Event.

Σε ότι αφορά στα μοντέλα μηχανικής μάθησης, έγινε προσπάθεια προσέγγισης του προβλήματος, τόσο με τεχνικές ταξινόμησης, με σκοπό την πρόβλεψη ετικετών αύξησης ή μείωσης ή σταθερής κατανομής των επισκεπτών, όσο και με τεχνικές παλινδρόμησης, με σκοπό την πρόβλεψη του ακριβούς αριθμού επισκεπτών στις διάφορες περιοχές ενδιαφέροντος του χώρου του φεστιβάλ. Στις μελέτες μας χρησιμοποιήσαμε ένα dataset διάρκειας χρονικής περιόδου 15 λεπτών

Όσον αφορά στο πρόβλημα ταξινόμησης, έγινε χρήση των αλγόριθμων Random Forest, KNN, SVM, Gaussian Naive Bayes και ενός νευρωνικού δικτύου 5 layers. Εκτός του εκτιμητή Naive Bayes, οι υπόλοιποι εκτιμητές φάνηκε να έχουν μία σχετικά ισορροπημένη συμπεριφορά την πρόβλεψη όλων των περιοχών ενδιαφέροντος αλλά και για τους διάφορους συνδυασμούς χαρακτηριστικών. Το νευρωνικό δίκτυο τα πήγε σε γενικές γραμμές καλύτερα, παρουσιάζοντας υψηλότερα ποσοστά ευστοχίας στην πλειονότητα των περιπτώσεων. Αυτό είναι ένα σημαντικό πλεονέκτημα των μοντέλων ταξινόμησης τα οποία παρότι δεν έχουν τη δυνατότητα ακριβούς αριθμητικής πρόβλεψης μπορούν να εφαρμοστούν καθολικά μέσα σε μία Fog αρχιτεκτονική χωρίς ιδιαίτερες προσαρμογές ανά περιοχή ενδιαφέροντος.

Σε ότι αφορά στο πρόβλημα παλινδρόμησης έγινε χρήση των αλγόριθμων Random Forest, KNN, SVM, Ridge regression και ενός νευρωνικού δικτύου 5 layers. Στην περίπτωση αυτή, φάνηκε να υπάρχει μία πολύ πιο έντονη αστάθεια στα σφάλματα των προβλέψεων, κυρίως ανάμεσα στις διαφορετικές περιοχές ενδιαφέροντος. Αυτό ενδεχομένως να μην αποτελεί σημαντικό πρόβλημα, δεδομένου ότι κάθε edge device μιας Fog αρχιτεκτονικής μπορεί να λειτουργεί με το δικό του μοντέλο ανάλογα με την περιοχή του χώρου στην οποία βρίσκεται.

Δυστυχώς σε ότι αφορά τα δεδομένα της δημοτικότητας των καλλιτεχνών και του καιρού, η συνεισφορά τους ήταν σχετικά μικρή στα ποσοστά των προβλέψεών μας. Σε ότι αφορά στον καιρό,

παρατηρήθηκε ήδη από την ανάλυση δεδομένων ότι η επίδραση του στις μεταβολές των κατανομών ήταν πολύ πιο μακροπρόθεσμη από 15 λεπτά. Ωστόσο η συνεισφορά τέτοιου είδους δεδομένων μένει να διερευνηθεί σε βάθος σε πιο πλούσια datasets τα οποία θα προσφέρουν συνεχείς πληροφορίες κίνησης των επισκεπτών.

Τέλος, έγινε πρόταση για μια νέα Fog αρχιτεκτονική, η οποία μπορεί να βασιστεί στα μοντέλα μηχανικής μάθησης που εκπαιδεύτηκαν κατά την εκπόνηση της διπλωματικής αυτής εργασίας. Στόχος της αρχιτεκτονικής αυτής είναι η ισορροπημένη κατανομή υπολογιστικού φόρτου ανάμεσα στο Edge και το Cloud, με σκοπό την αποτελεσματική εξυπηρέτηση των επισκεπτών συνδυασμένη με την ασφάλεια δεδομένων και το ελάχιστο δυνατό κόστος. Αυτό επιτυγχάνεται με διαρκή παρακολούθηση και υπολογισμό της τοποθεσίας των επισκεπτών μέσω της διαδικασία Trilateration, με χρήση των μοντέλων μηχανικής μάθησης για πρόβλεψη της θέσης τους σε επόμενες χρονικές στιγμές και στη συνέχεια με απόφαση για τη χρήση ή όχι πόρων Cloud, δίνοντας προτεραιότητα πάντοτε στο Edge. Η αρχιτεκτονική αυτή έχει τη δυνατότητα εφαρμογής και αξιολόγησης στο DasFest του 2019, ενώ ταυτόχρονα αναμένεται η έκδοση νέων πολύ πιο ολοκληρωμένων datasets, τα οποία θα καταστήσουν τα μοντέλα πολύ πιο ισχυρά και ακριβή στις προβλέψεις τους. Εκτός αυτού θα ανοίξουν οι δρόμοι για προσέγγιση του προβλήματος πρόβλεψης κατανομής των επισκεπτών από διαφορετικές οπτικές γωνίες, όπως για παράδειγμα οι συγκεκριμένη διαδρομές που ακολουθεί κάθε επισκέπτης.

Συντομογραφίες

ML	Machine Learning	Μηχανική Μάθηση
AI	Artificial Intelligence	Τεχνητή Νοημοσύνη
ANN	Artificial Neural Network	Τεχνητό Νευρωνικό Δίκτυο
DL	Deep Learning	Βαθιά Μάθηση
SL	Supervised Learning	Επιτηρούμενη μάθηση
SSL	Semi-Supervised Learning	Ημιεπιτηρούμενη Μάθηση
RL	Reinforcement Learning	Ενισχυτική Μάθηση
AUC	Area Under Curve	Εμβαδόν κάτω από την καμπύλη (ROC)
ROC	Receiver Operating Characteristic	Χαρακτηριστική καμπύλη ταξινομητή
TPR	Recall	Ανάκληση ή ευαισθησία
FPR	False Positive Rate	Ποσοστό των λανθασμένων θετικών προβλέψεων επί του συνόλου αρνητικών δειγμάτων
mse	Mean Squared Error	Μέσο τετραγωνικό σφάλμα
mae	Mean Absolute Error	Μέσο απόλυτο σφάλμα
mre	Mean Relative Error	Μέσο σχετικό σφάλμα
IoT	Internet of Things	Διαδίκτυο των Πραγμάτων
PoI	Point of Interest	Σημείο Ενδιαφέροντος
AoI	Area of Interest	Περιοχή Ενδιαφέροντος
KNN	K-Nearest Neighbors	Αλγόριθμος k-κοντινότερων γειτόνων
SVC	Support Vector Classifier	Ταξινομητής SVM
SVR	Support Vector Regressor	Εκτιμητής SVM για παλινδρόμηση
RSSI	Received signal strength indication	Δείκτης έντασης λαμβανόμενου σήματος (WiFi)

Βιβλιογραφικές αναφορές

[1] C.-J. Huang, Y.-J. Yang, D.-X. Yang, and Y.-J. Chen, “Frog classification using machine learning techniques,” *Expert Syst. Appl.*, vol. 36, no. 2, Part 2, pp. 3737–3743, Mar. 2009.

[2] W. McKinney and Others, “Data structures for statistical computing in python,” in *Proceedings of the 9th Python in Science Conference*, 2010, vol. 445, pp. 51–56.

[3] T. E. Oliphant, *A guide to NumPy*, vol. 1. Trelgol Publishing USA, 2006.

[4] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[5] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, vol. 1. Springer series in statistics New York, NY, USA:, 2001.

[6] L. Breiman, “Bagging predictors,” *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[7] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*. CRC press, 1994.

[8] M. Kearns and L. Valiant, “Cryptographic Limitations on Learning Boolean Formulae and Finite Automata,” *J. ACM*, vol. 41, no. 1, pp. 67–95, Jan. 1994.

[9] Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm,” *Icml*, 1996.

[10] J. H. Friedman, “Stochastic gradient boosting,” *Comput. Stat. Data Anal.*, vol. 38, no. 4, pp. 367–378, Feb. 2002.

[11] J. H. Friedman, “Greedy Function Approximation: A Gradient Boosting Machine,” *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[12] O. Chapelle, B. Scholkopf, and A. Zien, “Semi-Supervised Learning (Chapelle, O. et al., Eds.; 2006) [Book reviews],” *IEEE Transactions on Neural*, 2009.

[13] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, “Semi-supervised Learning with Deep Generative Models,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3581–3589.

[14] R. S. Sutton, A. G. Barto, and F. Bach, “Reinforcement learning: An introduction,” 1998.

[15] V. Kuleshov and D. Precup, “Algorithms for multi-armed bandit problems,” *arXiv [cs.AI]*, 25-Feb-2014.

[16] C. J. C. H. Watkins and P. Dayan, “Q-learning,” *Mach. Learn.*, vol. 8, no. 3, pp. 279–292, May 1992.

[17] H. Van Hasselt, A. Guez, and D. Silver, “Deep Reinforcement Learning with Double Q-Learning,” in *AAAI*, 2016, vol. 2, p. 5.

[18] V. Mnih *et al.*, “Playing Atari with Deep Reinforcement Learning,” *arXiv [cs.LG]*, 19-Dec-2013.

[19] G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*, vol. 37. University of Cambridge, Department of Engineering Cambridge, England, 1994.

[20] T. M. Mitchell and Others, “Machine learning. WCB.” McGraw-Hill Boston, MA:, 1997.

[21] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, “Supervised machine learning: A review of classification techniques,” *Emerging artificial intelligence applications in computer engineering*, vol. 160, pp. 3–24, 2007.

[22] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, “KNN Model-Based Approach in Classification,” in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, 2003, pp. 986–996.

[23] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[24] E. B. Hunt, J. Marin, and P. J. Stone, “PsycNET,” 1966. [Online]. Available: http://psycnet.apa.org/record/1966-08232-000. [Accessed: 26-Sep-2018].

[25] J. Chen *et al.*, “A parallel random forest algorithm for big data in a spark cloud computing environment,” *IEEE Trans. Parallel Distrib. Syst.*, no. 1, pp. 1–1, 2017.

[26] J. Shotton *et al.*, “Real-time human pose recognition in parts from single depth images,” in *CVPR 2011*, 2011, pp. 1297–1304.

[27] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297,

Sep. 1995.
[28] C.-W. Hsu and C.-J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Trans. Neural Netw.*, vol. 13, no. 2, pp. 415–425, 2002.
[29] K. Veropoulos, C. Campbell, N. Cristianini, and Others, “Controlling the sensitivity of support vector machines,” in *Proceedings of the international joint conference on AI*, 1999, vol. 55, p. 60.
[30] J. C. Platt, “Using Analytic QP and Sparseness to Speed Training of Support Vector Machines,” in *Advances in Neural Information Processing Systems 11*, M. J. Kearns, S. A. Solla, and D. A. Cohn, Eds. MIT Press, 1999, pp. 557–563.
[31] S. S. Keerthi and E. G. Gilbert, “Convergence of a Generalized SMO Algorithm for SVM Classifier Design,” *Mach. Learn.*, vol. 46, no. 1, pp. 351–360, Jan. 2002.
[32] I. Kavakiotis, O. Tsave, A. Salifoglou, N. Maglaveras, I. Vlahavas, and I. Chouvarda, “Machine Learning and Data Mining Methods in Diabetes Research,” *Comput. Struct. Biotechnol. J.*, vol. 15, pp. 104–116, Jan. 2017.
[33] J. Zhu and T. Hastie, “Kernel Logistic Regression and the Import Vector Machine,” in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. MIT Press, 2002, pp. 1081–1088.
[34] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian Network Classifiers,” *Mach. Learn.*, vol. 29, no. 2, pp. 131–163, Nov. 1997.
[35] R. O. Duda and P. E. Hart, “Pattern classification and scene analysis,” *A Wiley-Interscience Publication, New York: Wiley, 1973*, 1973.
[36] J. Davis and M. Goadrich, “The Relationship Between Precision-Recall and ROC Curves,” in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, Pennsylvania, USA, 2006, pp. 233–240.
[37] A. P. Bradley, “The use of the area under the ROC curve in the evaluation of machine learning algorithms,” *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, Jul. 1997.
[38] T. C. W. Landgrebe and R. P. W. Duin, “Approximating the multiclass ROC by pairwise analysis,” *Pattern Recognit. Lett.*, vol. 28, no. 13, pp. 1747–1758, Oct. 2007.
[39] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
[40] A. Tharwat, “Classification assessment methods,” *Applied Computing and Informatics*, Aug. 2018.
[41] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman, “Applied linear statistical models,” 1996.
[42] S. Weisberg, *Applied linear regression*, vol. 528. John Wiley & Sons, 2005.
[43] F. J. Anscombe, “The American Statistician 27,” *Graphs in Statistical Analysis*, no. 1, pp. 17–21, 1973.
[44] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*, vol. 821. John Wiley & Sons, 2012.
[45] A. E. Hoerl and R. W. Kennard, “Ridge Regression: Biased Estimation for Nonorthogonal Problems,” *Technometrics*, vol. 12, no. 1, pp. 55–67, Feb. 1970.
[46] R. Tibshirani, “Regression Shrinkage and Selection via the Lasso,” *J. R. Stat. Soc. Series B Stat. Methodol.*, vol. 58, no. 1, pp. 267–288, 1996.
[47] H. Duzan and N. S. B. M. Shariff, “Ridge regression for solving the multicollinearity problem: review of methods and models,” *J. Appl. Sci.* , vol. 15, no. 3, p. 392, 2015.
[48] I.-G. Chong and C.-H. Jun, “Performance of some variable selection methods when multicollinearity is present,” *Chemometrics Intellig. Lab. Syst.*, vol. 78, no. 1, pp. 103–112, Jul. 2005.
[49] R. Tibshirani, “Regression shrinkage and selection via the lasso: a retrospective,” *J. R. Stat. Soc. Series B Stat. Methodol.*, vol. 73, no. 3, pp. 273–282, 2011.
[50] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*, vol. 589. John wiley & sons, 2005.

[51] J. H. Friedman, “Multivariate Adaptive Regression Splines,” *Ann. Stat.*, vol. 19, no. 1, pp. 1–67, 1991.
[52] Γ. Ν. Κουρής, “Εφαρμογή τεχνικών data mining σε συστήματα ηλεκτρονικού εμπορίου,” 2006.
[53] J. A. Hartigan, *Clustering algorithms*. John Wiley & Sons, 1975.
[54] A. K. Jain, “Data clustering: 50 years beyond K-means,” *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.
[55] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.
[56] G. Nagy, “State of the art in pattern recognition,” *Proc. IEEE*, vol. 56, no. 5, pp. 836–863, 1968.
[57] E. Diday and J. C. Simon, “Clustering Analysis,” in *Communication and Cybernetics*, 1976, pp. 47–94.
[58] S. Lloyd, “Least squares quantization in PCM,” *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, 1982.
[59] D. Pelleg, A. W. Moore, and Others, “X-means: Extending k-means with efficient estimation of the number of clusters,” in *Icml*, 2000, vol. 1, pp. 727–734.
[60] D. Arthur and S. Vassilvitskii, “k-means++: The advantages of careful seeding,” in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007, pp. 1027–1035.
[61] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, and Others, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Kdd*, 1996, vol. 96, pp. 226–231.
[62] K. Fukunaga and L. Hostetler, “The estimation of the gradient of a density function, with applications in pattern recognition,” *IEEE Trans. Inf. Theory*, vol. 21, no. 1, pp. 32–40, Jan. 1975.
[63] I. Jolliffe, “Principal Component Analysis,” in *International Encyclopedia of Statistical Science*, M. Lovric, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1094–1096.
[64] H. Zou, T. Hastie, and R. Tibshirani, “Sparse Principal Component Analysis,” *J. Comput. Graph. Stat.*, vol. 15, no. 2, pp. 265–286, Jun. 2006.
[65] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust Principal Component Analysis?,” *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, Jun. 2011.
[66] B. Schölkopf, A. Smola, and K.-R. Müller, “Kernel principal component analysis,” in *Artificial Neural Networks — ICANN’97*, 1997, pp. 583–588.
[67] A. M. Jade, B. Srikanth, V. K. Jayaraman, B. D. Kulkarni, J. P. Jog, and L. Priya, “Feature extraction and denoising using kernel PCA,” *Chem. Eng. Sci.*, vol. 58, no. 19, pp. 4441–4448, Oct. 2003.
[68] H. Ince and T. B. Trafalis, “Kernel principal component analysis and support vector machines for stock price prediction,” *IIE Trans.*, vol. 39, no. 6, pp. 629–637, Mar. 2007.
[69] R. O. Duda, P. E. Hart, and D. G. Stork, “Pattern classification,” 2012.
[70] T. Li, S. Zhu, and M. Ogihara, “Using discriminant analysis for multi-class classification: an experimental investigation,” *Knowl. Inf. Syst.*, vol. 10, no. 4, pp. 453–472, Nov. 2006.
[71] A. M. Martínez and A. C. Kak, “Pca versus lda,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 2001.
[72] John Violos, Sotiris Pelekis, Anastasis Berdelis, Stylianos Tsanakas, Konstantinos Tserpes, Theodora Varvarigou, “Predicting Visitor Distribution for Large Events in Smart Cities,” presented at the 1st International Workshop on Big data, cloud, and IoT technologies for smart cities, Kyoto, Japan, 27 February, 2019.
[73] A. Psychas *et al.*, “Cloud toolkit for Provider assessment, optimized Application Cloudification and deployment on IaaS,” *Future Gener. Comput. Syst.*, Sep. 2018.
[74] J. Violos *et al.*, “User Behavior and Application Modeling in Decentralized Edge Cloud Infrastructures,” in *Economics of Grids, Clouds, Systems, and Services*, 2017, pp. 193–203.
[75] G. Z. Santoso *et al.*, “Dynamic Resource Selection in Cloud Service Broker,” in *2017 International Conference on High Performance Computing Simulation (HPCS)*, 2017, pp. 233–235.
[76] E. Carlini *et al.*, “BASMATI: Cloud Brokerage Across Borders for Mobile Users and Applications,” in *Advances in Service-Oriented and Cloud Computing*, 2018, pp. 181–186.

[77] J. Altmann *et al.*, “BASMATI: An Architecture for Managing Cloud and Edge Resources for Mobile Users,” pp. 56–66, 2017.
[78] J. D. Hunter, “Matplotlib: A 2D Graphics Environment,” *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, May 2007.
[79] F. Chollet and Others, “Keras.” 2015.
[80] M. Abadi *et al.*, “Tensorflow: a system for large-scale machine learning,” in *OSDI*, 2016, vol. 16, pp. 265–283.
[81] J. D. Hamilton, *Time series analysis*, vol. 2. Princeton university press Princeton, NJ, 1994.
[82] Z. Li, T. Braun, and D. C. Dimitrova, “A passive WiFi source localization system based on fine-grained power-based trilateration,” in *2015 IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2015, pp. 1–9.