ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Διερεύνηση χρήσης τεχνικών μηχανικής μάθησης για ταξινόμηση (classification) βιογραφικών

Διπλωματική Εργασία

της

Νίκης Γ. Γκόλια

Επιβλέπων : Ασκούνης Δημήτριος

Καθηγητής Ε.Μ.Π

Αθήνα, Μάρτιος 2021

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Διερεύνηση χρήσης τεχνικών μηχανικής μάθησης για ταξινόμηση (classification) βιογραφικών

Διπλωματική Εργασία

της

Νίκης Γ. Γκόλια

Επιβλέπων : Ασκούνης Δημήτριος

Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 18η Μαρτίου 2021.

…….	……..	……..
Δημήτριος Ασκούνης	Χρυσόστομος Δούκας	Ιωάννης Ψαρράς
Καθηγητής Ε.Μ.Π	Καθηγητής Ε.Μ.Π	Καθηγητής Ε.Μ.Π

Αθήνα, Μάρτιος 2021

……

ΝΙΚΗ Γ. ΓΚΟΛΙΑ

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

Ευχαριστίες,

στην οικογένεια και στους φίλους μου για την υποστήριξη, σε όλα τα επίπεδα, καθ'όλη τη διάρκεια των ακαδημαϊκών σπουδών μου, στον Ευάγγελο Καρακόλη για τις συμβουλές και την πολύτιμη καθοδήγησή του κατά τη διάρκεια της εκπόνησης της διπλωματικής μου εργασίας, στον καθηγητή Ασκούνη Δημήτριο για την υποστήριξη τόσο ως επιβλέπων της διπλωματικής όσο και ως καθηγητής στο προπτυχιακό μου στάδιο.

Περίληψη

Ο στόχος της παρούσας διπλωματικής είναι η διερεύνηση και η αξιολόγηση αλγορίθμων μηχανικής μάθησης για την ταξινόμηση βιογραφικών σημειωμάτων ανάλογα με το περιεχόμενό τους. Τα βιογραφικά σημειώματα προέρχονται από διαφορετικές πηγές και καταλήγουν στον υπεύθυνο ανθρώπινου δυναμικού ο οποίος καλείται να τα κατηγοριοποιήσει τόσο ως προς την ειδικότητα του υποψηφίου όσο και ως προς την καταλληλότητά του για την εκάστοτε θέση εργασίας. Η παρούσα διπλωματική εργασία στοχεύει πρωτίστως στο σχεδιασμό κατάλληλης μεθοδολογίας για την υποβοήθηση αυτής της διαδικασίας.

Σε αυτό το πλαίσιο, γνωστοί αλγόριθμοι επιβλεπόμενης μάθησης όπως οι Naïve Bayes, Decision Trees, Random Forest και Support Vector Model χρησιμοποιούνται για να δημιουργηθούν μοντέλα πρόβλεψης. Επιπλέον, στο πλαίσιο της διερεύνησης του συνόλου δεδομένων, εφαρμόζεται και η τεχνική συσταδοποίησης, με τον αλγόριθμο K-means. Προτού εφαρμοστούν οι παραπάνω αλγόριθμοι, τα δεδομένα πρέπει να προεπεξεργαστούν ώστε να μετατραπούν από απλά κείμενα σε διανύσματα συγκεκριμένου μεγέθους χαρακτηριστικών. Τα χαρακτηριστικά αποτελούνται από λέξεις που περιέχουν πληροφορία σχετικά με την κατηγορία του βιογραφικού. Ωστόσο, πολλά από αυτά τα χαρακτηριστικά δε διαθέτουν σημαντικές πληροφορίες για το περιεχόμενο του κειμένου. Για το λόγο αυτό, εφαρμόζονται ειδικές μέθοδοι για εξαγωγή χαρακτηριστικών προκειμένου να διατηρηθούν μόνο τα σημαντικά χαρακτηριστικά των κειμένων. Έπειτα, ο κάθε αλγόριθμος εφαρμόζεται σε ένα σύνολο δεδομένων ελέγχου προκειμένου να γίνει αξιολόγηση του μοντέλου.

Πέραν του πειραματικού μέρους της διπλωματικής εργασίας, παρουσιάζονται λεπτομερώς τόσο τα επιμέρους επιστημονικά πεδία στα οποία εντάσσεται η παρούσα εργασία, όσο και οι αλγόριθμοι και οι τεχνικές που χρησιμοποιήθηκαν.

Λέξεις κλειδιά: Κατηγοριοποίηση, Μηχανική Μάθηση, Naive Bayes, Decision Tree, Επεξεργασία Φυσικής Γλώσσας, Συσταδοποίηση

Abstract

The purpose of this thesis is to investigate and evaluate machine learning algorithms for the classification of CVs according to their content. Curriculum vitae come from different sources and end up to the human resources manager who is called upon to categorize them both in terms of the candidate's specialty and in terms of his suitability for the respective job position. The present thesis aims primarily at designing an appropriate methodology to assist in this process.

Well-known supervised learning algorithms such as Naïve Bayes, Decision Trees, Random Forest and Support Vector Model are used to create prediction models. Furthermore, during the data set investigation, the clustering technique with K-means algorithm is applied. Before the above algorithms can be applied, the data must be pre-processed to be converted from plain text to vectors of a certain size of attributes. Attributes consist of words that contain information about the resume category. However, many of these attributes lack important information of the content of text. For this reason, it is essential to apply several methods for feature extraction in order to keep only useful attributes for classification models. Finally, each algorithm is trained by some train data and then, used to predict the class of each text of test set. For the evaluation of the model, the total amount of correct predictions are taken into consideration.

Beyond the experimental part of this diploma thesis, the scientific fields, in which the present thesis is included, and the algorithms it contains, are presented.

Keywords: Categorization, Machine Learning, Naive Bayes, Decision Tree, Natural Language Processing, Clustering

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. ΕΙΣΑΓΩΓΗ

1.1 Γενική Εισαγωγή

Η Διοίκηση Ανθρώπινου Δυναμικού είναι μια οργανωτική λειτουργία που σχετίζεται με την προμήθεια και τη διατήρηση ταλαντούχων υπαλλήλων. Αποτελεί αντικείμενο συνεχούς έρευνας και ένα από τα βασικότερα τμήματα μιας εταιρείας, που στοχεύει στην εκτέλεση των απαραίτητων ενεργειών για την εργασιακή και προσωπική ικανοποίηση των εργαζομένων, όπως και την ανάπτυξη καλών σχέσεων μεταξύ τους.

Ένας από τους κυριότερους στόχους της διοίκησης ανθρωπίνου δυναμικού, επίσης, είναι η συνεχώς αυξανόμενη απόδοση των εργαζομένων η οποία επιτυγχάνεται ως επί το πλείστον με παροχή κινήτρων σε αυτούς.

Στην πλειονότητα των οργανισμών, οι εργαζόμενοι είναι καθοριστικοί για τη δημιουργία ενός βιώσιμου ανταγωνιστικού πλεονεκτήματος.

1.2 Εισαγωγή στην Επιλογή Ανθρώπινου Δυναμικού

Κύριο μέλημα του τμήματος Ανθρωπίνου Δυναμικού κάθε οργάνωσης είναι η στελέχωσή της. Η λειτουργία της στελέχωσης αποτελείται από τρεις σχετικές δραστηριότητες: τη στρατολόγηση, την επιλογή και την τοποθέτηση.

Στρατολόγηση είναι η ανάπτυξη μιας δεξαμενής υποψηφίων για θέσεις εργασίας μέσα σε μία οργάνωση. Η στρατολόγηση μπορεί να γίνει εσωτερικά της οργάνωσης ή εξωτερικά. Η εσωτερική στρατολόγηση λαμβάνει υπόψη ήδη υπάρχοντες υπαλλήλους για προαγωγές ή μεταβιβάσεις. Το πλεονέκτημά της είναι ότι οι εργοδότες γνωρίζουν τους υπαλλήλους και οι υπάλληλοι γνωρίζουν την οργάνωση και τις ανάγκες της. Αντίθετα, η εξωτερική στρατολόγηση φέρνει νέους υποψηφίους σε μια εταιρεία και μπορεί να εμπνεύσει την καινοτομία. Ανάμεσα στις πηγές εξωτερικών υποψηφίων είναι οι πίνακες εργασίας στο διαδίκτυο, που χρησιμοποιούνται συχνότατα, οι ιστοσελίδες των εταιρειών, οι αναφορές υπαλλήλων, οι αγγελίες εργασίας και η στρατολόγηση σε πανεπιστήμια. Σημαντικό ποσοστό των θέσεων εργασίας που καλύπτονται, οφείλονται σε προτάσεις που υποβάλλουν υπάρχοντες υπάλληλοι και στους διαδικτυακούς πίνακες εργασίας. Στην πραγματικότητα, έρευνες δείχνουν ότι οι προφορικές συστάσεις από εργαζόμενους είναι ο τρόπος με τον οποίο καταλαμβάνονται οι περισσότερες θέσεις εργασίας λόγω του χαμηλού κόστους και της τάσης των εργαζομένων να γνωρίζουν ποιος είναι κατάλληλος για την εταιρεία. Οι πίνακες εργασίας στο Διαδίκτυο όπως οι CareerBuilder και Monster έχουν σημειώσει άνοδο δημοτικότητας ως εργαλείο στρατολόγησης επειδή είναι διαθέσιμοι σε μεγάλο αριθμό ανθρώπων που αναζητούν εργασία. Ωστόσο, για εξειδικευμένες θέσεις, μεγάλος αριθμός εταιρειών αναζητούν εργαζόμενους μέσω ιστοσελίδων επαγγελματικής δικτύωσης όπως το LinkedIn καθώς οι πίνακες εργασίας

δημιουργούν τεράστιες βάσεις δεδομένων οι οποίες είναι δύσκολο για την εταιρεία να τις επεξεργαστεί. Επίσης, πολλές εταιρείες επιλέγουν να δημοσιεύουν ανοιχτές θέσεις εργασίας στην επίσημη ιστοσελίδα τους ή να αναζητούν εργαζόμενους μέσω πανεπιστημιακών ιδρυμάτων. Οι περισσότερες εταιρείες χρησιμοποιούν κάποιο συνδυασμό από τις παραπάνω μεθόδους ανάλογα με το αντικείμενό τους και τη θέση εργασίας.

Η διαδικασία την επιλογής είναι πλέον σημαντική καθώς σε αυτό το στάδιο αποφασίζεται ποιος από τους κατάλληλους υποψηφίους θα προσληφθεί από την εταιρεία. Κατά τη διαδικασία της επιλογής χρησιμοποιούνται ορισμένα εργαλεία όπως οι αιτήσεις και τα βιογραφικά, και η δομημένη συνέντευξη.

Οι έντυπες αιτήσεις και τα βιογραφικά παρέχουν βασικές πληροφορίες στους εργοδότες. Για να κάνουν την πρώτη επιλογή ανάμεσα στους υποψηφίους, οι εργοδότες επιθεωρούν τα χαρακτηριστικά και τα στοιχεία των διαφόρων υποψήφιων εργαζομένων. Οι αιτήσεις και τα βιογραφικά περιλαμβάνουν πληροφορίες σχετικά με το μορφωτικό επίπεδο, την ειδικότητα, την υπηκοότητα, την εργασιακή εμπειρία, τα διαπιστευτήρια, τις δεξιότητες και άλλα στοιχεία του υποψηφίου.

Οι συνεντεύξεις αποτελούν το βασικότερο εργαλείο αξιολόγησης των υποψηφίων καθώς βοηθούν στη δημιουργία επαφής και παρέχουν μια αίσθηση της προσωπικότητας του υποψηφίου στον εργοδότη. Επίσης, ο εργοδότης έχει τη δυνατότητα να θέσει στον υποψήφιο ως ερώτημα την πιθανή αντιμετώπιση ορισμένων περιστάσεων ώστε να αντιληφθεί τις ικανότητές του. Σε ορισμένες συνεντεύξεις δίνεται ιδιαίτερη βαρύτητα σε θεωρητικές ή τεχνικές ερωτήσεις που αφορούν τον τομέα της θέσεως εργασίας ώστε να αξιολογηθούν οι γνώσεις και η εμπειρία του υποψηφίου.

Ένα σύνηθες βήμα στη διαδικασία επιλογής είναι ο έλεγχος των συστάσεων ώστε να διαπιστωθούν τα λεγόμενα του υποψηφίου σχετικά με τα διαπιστευτήρια, τις ικανότητες και την εργασιακή του εμπειρία.

Οι περισσότερες οργανώσεις χρησιμοποιούν τα παραπάνω εργαλεία για την επιλογή εργαζομένων σε μια συγκεκριμένη θέση εργασίας. Υπάρχουν και οργανώσεις οι οποίες χρησιμοποιούν επιπλέον παραδοσιακές μεθόδους όπως τα τεστ προσωπικότητας, εντιμότητας και γνωστικών ικανοτήτων, έλεγχο για ναρκωτικές ουσίες, τεστ απόδοσης και άλλα εργαλεία.

1.3 Εισαγωγή στην Εκπαίδευση και Ανάπτυξη Ανθρώπινου Δυναμικού

Το σημερινό ανταγωνιστικό περιβάλλον απαιτεί από τους διευθυντές να αναβαθμίζουν διαρκώς τις ικανότητες και την απόδοση των υπαλλήλων αλλά και τις δικές τους. Αυτή η συνεχής βελτίωση αυξάνει τόσο την προσωπική όσο και την επιχειρησιακή αποτελεσματικότητα. Τα μέλη της οργάνωσης γίνονται πιο χρήσιμα

στην εργασία που τους έχει ανατεθεί τη δεδομένη χρονική στιγμή και προετοιμάζονται ώστε να λάβουν νέες ευθύνες. Ως εκ τούτου, η οργάνωση δύναται να αντιμετωπίσει νέες προκλήσεις και να εκμεταλλευτεί τις νέες μεθόδους και τεχνολογίες που προκύπτουν.

Η ανάπτυξη του εργατικού δυναμικού περιλαμβάνει εκπαίδευση και δραστηριότητες εξέλιξης. Περιλαμβάνει επίσης την αξιολόγηση της απόδοσης των υπαλλήλων και την παροχή αποτελεσματικών πληροφοριών ώστε να έχουν κίνητρο να αποδώσουν κατά το μέγιστο βαθμό.

Η πρώτη φάση της εκπαίδευσης συνήθως αρχίζει με μία αξιολόγηση αναγκών. Οι διευθυντές διεξάγουν ανάλυση ώστε να εκτιμήσουν τις εργασίες, τους ανθρώπους, και τα τμήματα που χρειάζονται εκπαίδευση.

Η δεύτερη φάση περιλαμβάνει το σχεδιασμό των εκπαιδευτικών προγραμμάτων. Τα αποτελέσματα της αξιολόγησης αναγκών καθορίζουν το περιεχόμενο της εκπαίδευσης.

Η τρίτη φάση περιλαμβάνει αποφάσεις σχετικά με τις εκπαιδευτικές διαδικασίες. Συνήθεις εκπαιδευτικές μέθοδοι που εφαρμόζονται αρκετά χρόνια είναι οι διαλέξεις, τα παιχνίδια ρόλων, η προσομοίωση, η μοντελοποίηση συμπεριφοράς, τα συνέδρια, η πρακτική άσκηση σε προσομοιωμένο εργατικό περιβάλλον και οι μαθητείες. Η μέθοδος που ακολουθείται οφείλει να ανταποκρίνεται στις ανάγκες που προέκυψαν με βάση την έρευνα στην πρώτη φάση.

Τέλος, η τέταρτη φάση της εκπαίδευσης θα πρέπει να αξιολογήσει την αποτελεσματικότητας του εκπαιδευτικού προγράμματος. Κριτήριο και μέτρο αποτελεσματικότητας αποτελούν οι αντιδράσεις των υπαλλήλων, η μάθηση(τεστ), η βελτιωμένη συμπεριφορά στην εργασία και τα αποτελέσματα(π.χ. μια πιθανή αύξηση στην παραγωγή ή στις πωλήσεις ή μια μείωση στα ποσοστά ελαττωματικών προϊόντων).

1.4 Εισαγωγή στην Αξιολόγηση Ανθρώπινου Δυναμικού

Η αξιολόγηση απόδοσης (performance appraisal) αποτελεί μια από τις σημαντικότερες ευθύνες ενός διευθυντή καθώς η σωστή διεξαγωγή της οδηγεί τους υπαλλήλους σε βελτιωμένη συμπεριφορά, αμοιβή και συντελεί στην πιθανή προαγωγή τους. Επιπλέον, η αξιολόγηση της απόδοσης καλλιεργεί την επικοινωνία ανάμεσα σε διευθυντές και υπαλλήλους και τελικά αυξάνει την αποτελεσματικότητα των υπαλλήλων και της οργάνωσης.

Η αξιολόγηση της απόδοσης έχει διοικητικό και εξελικτικό ρόλο. Διοικητικά, παρέχει στους διευθυντές τις πληροφορίες ώστε να λάβουν αποφάσεις σχετικά με μισθούς, προαγωγές, απολύσεις και ταυτόχρονα βοηθάει τους υπαλλήλους να κατανοήσουν τη

βάση των αποφάσεων αυτών. Σε επίπεδο εξέλιξης της οργάνωσης, παρέχει πληροφορίες που αφορούν την αναγνώριση και το σχεδιασμό της εκπαίδευσης ή άλλων βελτιώσεων που χρειάζεται ο υπάλληλος ώστε να βελτιώσει την καθημερινή του απόδοση.

Οι αξιολογήσεις απόδοσης μπορούν να κρίνουν τρεις βασικές κατηγορίες της απόδοσης ενός υπαλλήλου: τα χαρακτηριστικά, τις συμπεριφορές, και τα αποτελέσματα. Η αξιολόγηση χαρακτηριστικών περιλαμβάνει υποκειμενικές κρίσεις για τα χαρακτηριστικά του υπαλλήλου που αφορούν την απόδοση. Τα χαρακτηριστικά αυτά σχετίζονται με την πρωτοβουλία, την ηγεσία και τη συμπεριφορά του υπαλλήλου. Η αξιολόγηση των αποτελεσμάτων είναι περισσότερο αντικειμενική και μπορεί να επικεντρωθεί σε ζητήματα παραγωγής, πωλήσεων, κερδών κλπ ανάλογα με το αντικείμενο της θέσης.

Η μη σωστή διεξαγωγή της αξιολόγησης έχει πολύ αρνητικό αντίκτυπο τόσο για το προσωπικό όσο και για τη διοίκηση και τελικά για την ομαλή και εξελισσόμενη πορεία της οργάνωσης. Μπορεί να προκαλέσει δυσαρέσκεια, να μειώσει το κίνητρο, να ελαχιστοποιήσει την απόδοση και ενδεχομένως να εκθέσει την οργάνωση σε νομικές κυρώσεις.

1.5 Εισαγωγή και Μορφή Βιογραφικών Σημειωμάτων

Το βιογραφικό σημείωμα αποτελεί το δημοφιλέστερο έγγραφο με το οποίο ένας υποψήφιος εργαζόμενος παρουσιάζει τον εαυτό του κατά την αίτησή του για μια θέση εργασίας. Από την πλευρά της διοίκησης, ο υπεύθυνος αξιολογεί τα βιογραφικά σημειώματα που λαμβάνει αποκτώντας μια πρώτη εικόνα για τους υποψηφίους σε ό,τι αφορά την καταλληλότητά τους για τη θέση. Δεδομένου ότι το βιογραφικό σημείωμα παρέχει την πρώτη εντύπωση για τον υποψήφιο και επομένως καθορίζει το ενδεχόμενο να κληθεί για το επόμενο στάδιο έως την πρόσληψή του, το έγγραφο αυτό οφείλει να περιλαμβάνει σημαντικές πληροφορίες για τη ζωή και το έργο του υποψηφίου που αφορούν τον εργοδότη. Τέτοιες πληροφορίες αφορούν την πανεπιστημιακή κατάρτιση, την εξειδίκευση σε κάποιο τομέα, την επαγγελματική εμπειρία, τις ομιλούμενες γλώσσες, τις θεωρητικές και τεχνικές δεξιότητες, τα συνέδρια και σεμινάρια που ο υποψήφιος έχει παρακολουθήσει, τις διακρίσεις και την εθελοντική του δράση, ορισμένες δημοσιεύσεις ή πιστοποιήσεις και οποιοδήποτε στοιχείο περιγράφει επαρκώς την προσωπικότητά του και τα επιτεύγματά του.

Τα βιογραφικά στην πλειονότητά τους είναι συνοπτικά έγγραφα που αναφέρουν επιγραμματικά τα χαρακτηριστικά του υποψηφίου χωρίς υπερφίαλες ή περιττές αναφορές. Συνήθως ο κάτοχος εισαγωγικά ξεκινάει με μια σύντομη περιγραφή του εαυτού του η οποία είναι μεστή και ταυτόχρονα ουσιαστική για την αίτηση.

Στα περισσότερα βιογραφικά τα γεγονότα παρατίθενται σε χρονολογική σειρά ξεκινώντας με τις προσωπικές πληροφορίες οι οποίες οφείλουν να περιλαμβάνουν το όνομα, την πόλη διαμονής, ορισμένα στοιχεία επικοινωνίας(τηλέφωνο, διεύθυνση email) και links του προφίλ στα social media.

Έπειτα ακολουθούν στοιχεία που αφορούν την εκπαίδευση(εκπαιδευτικό ίδρυμα, τίτλος πτυχίου, εξειδίκευση), την επαγγελματική εμπειρία(πρότερες επαγγελματικές θέσεις, καθήκοντα, ευθύνες και αποτελέσματα), τις δεξιότητες (hard skills, soft skills), τις διακρίσεις, τα σεμινάρια. Τα γεγονότα οφείλουν να είναι χωρισμένα σε κατηγορίες ώστε να διευκολυνθεί το έργο του εργοδότη ο οποίος καλείται να αξιολογήσει μεγάλο όγκο βιογραφικών σημειωμάτων έως ότου αποφασίσει ποιοι θα κληθούν για συνέντευξη.

Περισσότερες πληροφορίες παρέχονται στο βιβλίο (Γκάσης #) κεφ.10 (Η Διοίκηση Ανθρωπίνων Πόρων)

2. ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ (Machine Learning)

2.1 Εισαγωγή στη Μηχανική Μάθηση

Η **μηχανική μάθηση** (Machine Learning) είναι μια περιοχή της τεχνητής νοημοσύνης η οποία περιλαμβάνει αλγορίθμους και μεθόδους με βάση τις οποίες οι υπολογιστές "μαθαίνουν" μέσω της παράλληλης τροφοδότησής τους με δεδομένα. Οι αλγόριθμοι της μηχανικής μάθησης συνδυάζονται κατάλληλα και συνθέτουν μοντέλα τα οποία δέχονται ως είσοδο σύνολα δεδομένων ώστε να πραγματοποιούν προβλέψεις ή να παίρνουν αποφάσεις. Αυτό σημαίνει ότι οι αλγόριθμοι εκτελούνται με βάση την επεξεργασία δεδομένων και παράγουν αποτελέσματα σε μια συνεχή διαδικασία αυτοβελτίωσης από τα δεδομένα και χωρίς να εκτελούν ρητά κάποιες εντολές ενός προγραμματιστή. Η μηχανική μάθηση αποτελεί υποπεδίο της επιστήμης των υπολογιστών που αναπτύχθηκε από την αναγνώριση προτύπων και της υπολογιστικής θεωρίας μάθησης στην τεχνητή νοημοσύνη. Χαρακτηριστικά παραδείγματα εφαρμογών μηχανικής μάθησης αποτελούν τα φίλτρα της ανεπιθύμητης αλληλογραφίας (spam filtering), η οπτική αναγνώριση χαρακτήρων (OCR), οι μηχανές αναζήτησης και η όραση υπολογιστών. Η μηχανική μάθηση έχει κοινά στοιχεία με τη στατιστική η οποία επίσης αποτελεί εργαλείο για την πραγματοποίηση προβλέψεων. Σε ό,τι αφορά την ανάλυση δεδομένων, η μηχανική μάθηση χρησιμοποιείται για τη δημιουργία σύνθετων προβλεπτικών μοντέλων και αλγορίθμων. Σύμφωνα με αυτά τα αναλυτικά μοντέλα, οι επιστήμονες αναλυτές δεδομένων, μηχανικοί και ερευνητές καταλήγουν σε αποφάσεις και συγκρίσεις από τη μάθηση μέσω ιστορικών δεδομένων και τάσεων στα δεδομένα.

2.2 Ορισμοί Μηχανικής Μάθησης και Ιστορικά στοιχεία

Ως επιστημονικό εγχείρημα, η μηχανική μάθηση αναπτύχθηκε από την έρευνα στην τεχνητή νοημοσύνη. Τις τελευταίες δεκαετίες του 20ου αιώνα δόθηκαν αρκετοί ορισμοί για τη μηχανική μάθηση από επιστήμονες και πρωτοπόρους της βιομηχανίας. Το 1950, ο Alan Turing (1913-1954), ο οποίος θεωρείται "πατέρας" της τεχνητής νοημοσύνης, εμπνεύστηκε μια δοκιμασία η οποία έχει στόχο την αναγνώριση ευφυών μηχανών. Η δοκιμασία αυτή ονομάζεται Turing Test και αποτελείται από έναν κριτή, έναν άνθρωπο και έναν υπολογιστή. Κατά τη διάρκεια του Turing Test, ο κριτής υποβάλλει μια σειρά από ερωτήσεις στον άνθρωπο και στον υπολογιστή χωρίς να γνωρίζει σε ποιόν απευθύνεται κάθε φορά. Αν στο τέλος δεν καταφέρει να ξεχωρίσει τον άνθρωπο από τον υπολογιστή, τότε θεωρείται ότι ο υπολογιστής έχει κερδίσει. Το πείραμα αυτό συνεχίζεται μέχρι και σήμερα με τη μορφή διαγωνισμού (βραβείο Loebner) και απονέμει βραβείο στον <<ευφυή>> υπολογιστή του οποίου οι απαντήσεις δε μπορούν να διακριθούν από ενός ανθρώπου.

Το 1959, ο Αμερικανός πρωτοπόρος στον τομέα των υπολογιστών και της τεχνητής νοημοσύνης, Arthur Samuel(1901-1990), όρισε τη μηχανική μάθηση ως "*Ένα πεδίο μελέτης που δίνει την ικανότητα στους υπολογιστές να μαθαίνουν χωρίς να έχουν ρητά προγραμματιστεί*".

Το 1997, ο καθηγητής, επιστήμονας υπολογιστών και συγγραφέας Tom M. Mitchell(1951-) έδωσε έναν πιο επίσημο ορισμό για τη μηχανική μάθηση που χρησιμοποιείται ευρέως: "*Ένα πρόγραμμα υπολογιστή λέμε ότι μαθαίνει από ένα πρόγραμμα E ως προς μια κλάση εργασιών T και ένα μέτρο επίδοσης P, αν η επίδοσή του σε εργασίες της κλάσης T, όπως αποτιμάται από το μέτρο P, βελτιώνεται με την εμπειρία E*". Με βάση τον ορισμό αυτό εισάγεται ένα βασικό λειτουργικό πλαίσιο για τη μηχανική μάθηση θέτοντας τους συγκεκριμένους όρους γύρω από την έννοιά της:

- **Εργασίες (T)**
- **Επίδοση (P)**
- **Εμπειρία (E)**

Με τον τρόπο αυτό ενώ ο υπολογιστής εκτελεί μια σειρά από διεργασίες, αυξάνει την απόδοσή του μέσω της εμπειρίας που αποκτά.

2.3 Κατηγορίες Προβλημάτων και Τεχνικές Μηχανικής Μάθησης

Υπάρχει μεγάλο πλήθος αλγορίθμων μηχανικής μάθησης που μπορούν να χρησιμοποιηθούν για την επίλυση προβλημάτων. Αντίστοιχα, υπάρχει μεγάλη ποικιλία προβλημάτων με διαφορετικές απαιτήσεις και επιθυμητές εξόδους. Ανάλογα με τη φύση και τις ανάγκες του προβλήματος καθορίζεται και ένα σύνολο τεχνικών μηχανικής μάθησης που πρέπει να εφαρμοστούν. Οι εργασίες μηχανικής μάθησης ταξινομούνται σε τρεις βασικές κατηγορίες:

την επιβλεπόμενη μάθηση, τη μη επιβλεπόμενη μάθηση και την ενισχυτική μάθηση.

2.3.1 Επιβλεπόμενη Μάθηση (Supervised Learning)

Η λογική της επιβλεπόμενης μάθησης είναι πως ένας <<δάσκαλος>> εισάγει παραδειγματικές εισόδους και τις αντίστοιχες εξόδους σε μια υπολογιστική μηχανή προκειμένου να δημιουργηθεί ένας γενικός κανόνας ο οποίος θα αντιστοιχίζει τις επόμενες εισόδους στις αντίστοιχες εξόδους τους. Επι της ουσίας, ο αλγόριθμος δημιουργεί μια συνάρτηση για να καταφέρει να απεικονίσει τα δεδομένα από το σύνολο εκπαίδευσης στις εξόδους τους που είναι ήδη γνωστές. Μετέπειτα, γενικεύοντας τη συνάρτηση αυτή, έχει τη δυνατότητα να εκτελεί προβλέψεις για δεδομένα με άγνωστη εκ των προτέρων έξοδο (σύνολο ελέγχου).

Στην επιβλεπόμενη μάθηση, κατατάσσονται δύο βασικές υποκατηγορίες προβλημάτων: η ταξινόμηση (classification) και η παλινδρόμηση (regression).

Η ταξινόμηση (classification) αφορά στη δημιουργία μοντέλων πρόβλεψης διακριτών τάξεων, όπως για παράδειγμα την ομάδα αίματος ενός ανθρώπου ή το αποτέλεσμα μιας βιοψίας (καλοήθης ή κακοήθης όγκος) κλπ. Τα δεδομένα εισόδου χωρίζονται σε δύο ή περισσότερες κλάσεις και ο υπολογιστής πρέπει να δημιουργήσει ένα μοντέλο το οποίο θα τοποθετεί τα δεδομένα σε μια ή περισσότερες (multi-label classification) ομάδες (κλάσεις). Γνωστό παράδειγμα ταξινόμησης είναι το spam filtering κατά το οποίο το μοντέλο δέχεται ως είσοδο τα emails και τα ταξινομεί στις κλάσεις "spam" ή "not spam" ανάλογα με το περιεχόμενό τους. Στα περισσότερα προβλήματα το πλήθος των κλάσεων εξόδου είναι μικρό και διακριτό, ενώ συνήθως δεν ξεπερνά τις δύο.

Η παλινδρόμηση (regression) από την άλλη πλευρά αφορά στη δημιουργία προβλέψεων συνεχών αριθμητικών τιμών με βάση τα δεδομένα εισόδου. Το μοντέλο επιχειρεί να δημιουργήσει μια συνάρτηση που να αντιστοιχίζει τα δεδομένα που έχουμε στις εξόδους τους, ώστε να παράγει όσο το δυνατόν εγκυρότερες τιμές για τις μελλοντικές εξόδους. Η απλούστερη συνάρτηση που μπορεί να δημιουργήσει το μοντέλο είναι γραμμική της μορφής y=ax+b. Οι παράμετροι a,b υπολογίζονται ώστε να ανταποκρίνονται στα ήδη υπάρχοντα ζεύγη δεδομένων και έπειτα, δοθείσης μιας τιμής εισόδου x, προβλέπεται η έξοδος y.

Οι κυριότερες τεχνικές επιβλεπόμενης μηχανικής μάθησης είναι οι εξής:

- Μάθηση Εννοιών (Concept Learning)
- Δένδρα Απόφασης (Decision Trees)
- Μάθηση Κανόνων (Rule Learning)
- Μάθηση κατά Περίπτωση (Instance Based Learning)
- Μάθηση κατά Bayes
- Γραμμική Παρεμβολή (Linear Regression)
- Νευρωνικά Δίκτυα (Neural Networks)
- Μηχανές Διανυσμάτων Υποστήριξης (Support Vectors Machines)

2.3.2 Μη επιβλεπόμενη μάθηση (Unsupervised Learning)

Στα προβλήματα Μη επιβλεπόμενης Μάθησης ο υπολογιστής προσπαθεί μέσα από τη δομή των δεδομένων να βρει συσχετίσεις και ομάδες που υπάρχουν χωρίς να έχει καμία πρότερη εμπειρία. Σαν αποτέλεσμα (έξοδος) προκύπτουν πρότυπα (περιγραφές) καθένα εκ των οποίων περιγράφει ένα μέρος των δεδομένων που σχετίζονται μέσω κάποιας ιδιότητας. Τα δεδομένα που έχουμε στη διάθεσή μας στην περίπτωση αυτή, δεν έχουν κάποια γνωστή ετικέτα και επομένως δεν υπάρχει η δυνατότητα εκτίμησης πιθανού λάθους και συνεπώς αξιολόγησης της αποδοτικότητας του μοντέλου. Αυτή είναι η βασικότερη διαφοροποίηση μεταξύ επιβλεπόμενης και μη επιβλεπόμενης μάθησης. Η κατηγορία αυτή περιλαμβάνει και τεχνικές οι οποίες αναλύουν τα δεδομένα προσπαθώντας να εξάγουν και να συνοψίσουν ορισμένα χαρακτηριστικά τους. Τέτοιες τεχνικές χρησιμοποιούνται αρκετά στο κομμάτι της

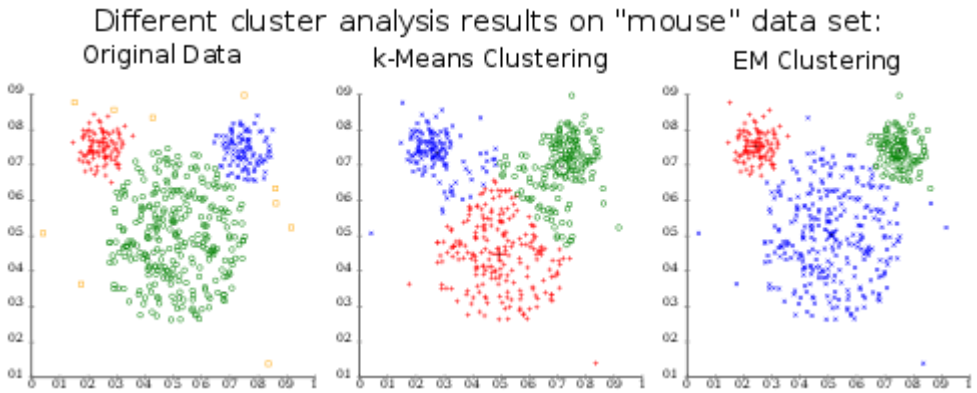
εξόρυξης δεδομένων (data mining) στη διαδικασία προεπεξεργασίας των δεδομένων (preprocessing).

Μία από τις κυριότερες προσεγγίσεις της μη επιβλεπόμενης μάθησης είναι η *Συσταδοποίηση (Clustering)*. Η συσταδοποίηση είναι η μέθοδος κατά την οποία τα αντικείμενα χωρίζονται σε ένα σύνολο από ομάδες (συστάδες). Η μέθοδος αυτή ομαδοποιεί τα δεδομένα με τέτοιο τρόπο ώστε τα αντικείμενα της ίδιας ομάδας (cluster) να παρουσιάζουν μεγαλύτερη ομοιότητα και περισσότερα κοινά στοιχεία σε σχέση με αντικείμενα που ανήκουν σε άλλη ομάδα. Οι ομάδες στις οποίες χωρίζονται τα δεδομένα δεν είναι γνωστές εξαρχής ούτε στο πλήθος ούτε στο περιεχόμενό τους. Για το λόγο αυτό, η τεχνική αυτή ανήκει στην κατηγορία της μη επιβλεπόμενης μάθησης.

Τα προβλήματα συσταδοποίησης μπορούν να επιλυθούν εφαρμόζοντας διάφορους αλγορίθμους οι οποίοι θα αναλυθούν στη συνέχεια.

Τα βασικά στάδια της διαδικασίας της συσταδοποίησης είναι τα εξής:

- **Επιλογή χαρακτηριστικών γνωρισμάτων**: Με την προεπεξεργασία των δεδομένων εξάγονται τα καταλληλότερα γνωρίσματα στα οποία πρόκειται να εφαρμοστεί η συσταδοποίηση ώστε να επιτευχθεί μεγαλύτερη ομοιογένεια σε κάθε ομάδα.
- **Επιλογή κατάλληλου αλγόριθμου συσταδοποίησης**: Ανάλογα το σύνολο δεδομένων που έχουμε στη διάθεσή μας, επιλέγουμε τον κατάλληλο αλγόριθμο ώστε να καθορίσει το καλύτερο σχήμα συσταδοποίησης. Για την επιλογή του αλγορίθμου χρησιμοποιείται το μέτρο γειτνίασης (proximity measure) και το κριτήριο συσταδοποίησης (clustering criterion).
- **Επικύρωση αποτελεσμάτων**: Σε περίπτωση που έχουμε κάποια ήδη γνωστά αποτελέσματα, στη φάση αυτή αξιολογούμε την ορθότητα των αποτελεσμάτων του αλγορίθμου. Ένας τρόπος αξιολόγησης είναι επίσης η σύγκριση των αποτελεσμάτων δύο ή περισσότερων συσταδοποιήσεων. Για παράδειγμα, στην παρακάτω εικόνα βλέπουμε τα αποτελέσματα των συσταδοποίησεων διαφορετικών αλγορίθμων.



Σχ2.1 Αποτελέσματα παραδείγματος συσταδοποίησης με διαφορετικές μεθόδους

Πηγή: https://commons.wikimedia.org/wiki/File:ClusterAnalysis_Mouse.svg

- **Ερμηνεία των αποτελεσμάτων**: Στο στάδιο αυτό οι αναλυτές καλούνται να εξάγουν γνώση από τις παραχθείσες συστάδες.

Υπάρχει ένα μεγάλο πλήθος αλγορίθμων συσταδοποίησης και ο καθένας εξ αυτών βασίζεται σε διαφορετική φιλοσοφία. Οι βασικές κατηγορίες αλγορίθμων είναι οι εξής:

1. K-means

Ο συγκεκριμένος αλγόριθμος χρησιμοποιείται στα περισσότερα προβλήματα, αποτελεί τη ρίζα για πολλούς άλλους και στοχεύει στη βελτιστοποίηση μιας συνάρτησης (συνάρτηση κόστους).

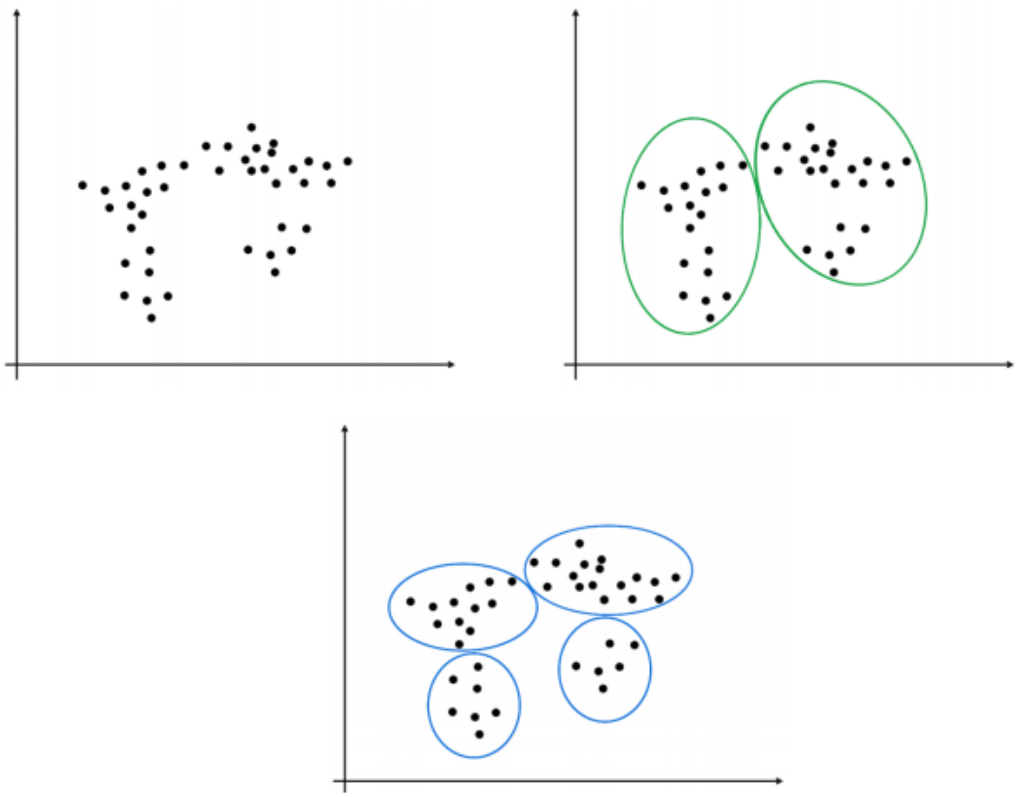
Αρχικά, έχουμε k ομάδες και η καθεμία αντιπροσωπεύεται από το μέσο διάνυσμα. Ο αλγόριθμος περιλαμβάνει τα εξής δύο βήματα:

1) Κατά τη φάση της διαμέρισης υπολογίζεται η ευκλείδεια απόσταση των διανυσμάτων που αναπαριστούν τα δεδομένα από το μέσο διάνυσμα καθεμιάς από τις k ομάδες $(d(x_i, C_j), \ \forall j = 1, 2, \ldots k)$. Δηλαδή, υπολογίζεται η απόσταση των δεδομένων από την κάθε ομάδα και προσδιορίζονται N αριθμοί q τέτοιοι ώστε $d(x_i, C_q) \leq d(x_i, C_j), \ \forall j = 1, 2, \ldots k$ με αποτέλεσμα να δημιουργούνται k σύνολα διανυσμάτων, ένα για κάθε ομάδα.
2) Ενημερώνονται τα μέσα διανύσματα των k ομάδων μετά τις προσθήκες των νέων διανυσμάτων του παραπάνω βήματος σε κάθε ομάδα.

Ο αλγόριθμος ολοκληρώνεται όταν οι ενημερώσεις που διενεργούνται στα μέσα διανύσματα των ομάδων, είναι πλέον αμελητέες.

Ένα βασικό μειονέκτημα του k-means αλγορίθμου είναι το γεγονός ότι δεν υπάρχει κάποιος συγκεκριμένος τρόπος να καθοριστεί ο αριθμός k των συστάδων. Αντίθετα, ο αριθμός k δίνεται ως είσοδος από το χρήστη και έγκειται στην δική του γνώση και εμπειρία η καταλληλότητα του αριθμού αυτού. Επιπλέον, δεν είναι γνωστό ούτε το χαρακτηριστικό της κλάσης των δεδομένων, οπότε απαιτείται εξερεύνηση των δεδομένων ώστε να εξαχθούν οι απαραίτητες πληροφορίες για τον καθορισμό του αριθμού των συστάδων.

Αρκετές φορές, ακόμα και η οπτικοποίηση των δεδομένων δεν είναι ιδιαίτερα βοηθητική για τον καθορισμό του αριθμού των συστάδων γιατί τα δεδομένα από τη φύση τους είναι διφορούμενα. Στο παρακάτω σχήμα φαίνεται μια τέτοια περίπτωση όπου δεν είναι ξεκάθαρες οι κλάσεις στις οποίες ανήκουν τα δεδομένα:



Σχ.2.2 Μη προφανείς συστάδες των δεδομένων

2. Ιεραρχικοί αλγόριθμοι

Οι ιεραρχικοί αλγόριθμοι συσταδοποίησης, όπως προδίδει το όνομά τους, δημιουργούν μια ιεραρχία εμφωλιασμένων συσταδοποιήσεων. Δηλαδή, κάθε

συστάδα περιέχει μεμονωμένα στοιχεία και άλλες συστάδες, οι οποίες με τη σειρά τους περιέχουν κι άλλες συστάδες χαμηλότερου επιπέδου. Με τον τρόπο αυτό, σχηματίζονται επίπεδα ιεραρχίας.

Οι ιεραρχικοί αλγόριθμοι χωρίζονται σε δύο κατηγορίες:

- Συσσωρευτικοί αλγόριθμοι
- Διαιρετικοί αλγόριθμοι

Οι συγκεκριμένοι αλγόριθμοι μπορούν να αναπαρασταθούν πλήρως με δενδροδιαγράμματα στα οποία φαίνεται η διάταξη των συστάδων. Πρακτικά, κάθε επίπεδο ενός δενδρικού διαγράμματος είναι ένα βήμα του αλγορίθμου. Κύριο πλεονέκτημα των αλγορίθμων αυτών είναι ότι μπορεί να επιτευχθεί συσταδοποίηση με οποιοδήποτε αριθμό συστάδων επιλέγοντας κάθε φορά το επιθυμητό επίπεδο ιεραρχίας στο δενδροδιάγραμμα.

2.1 Συσσωρευτικοί αλγόριθμοι

Η βασική ιδέα των συσσωρευτικών αλγορίθμων είναι το γεγονός ότι ξεκινούν με n δείγματα σε n διαφορετικές συστάδες. Δηλαδή κάθε δείγμα βρίσκεται σε μια ξεχωριστή συστάδα. Σε κάθε βήμα του αλγορίθμου ενώνονται οι δύο κοντινότερες συστάδες μειώνοντας το πλήθος των συστάδων κάθε φορά κατά ένα. Στο τελευταίο επίπεδο έχουμε καταλήξει να έχουμε μία μόνο συστάδα με όλα τα δείγματα n.

2.2 Διαιρετικοί αλγόριθμοι

Αντίθετα με τους συσσωρευτικούς αλγόριθμους, ένας διαιρετικός αλγόριθμος ξεκινάει με n δείγματα σε μία συστάδα. Σε κάθε βήμα, μία ομάδα διασπάται σε δύο έως ότου καταλήξουμε να έχουμε n διαφορετικές συστάδες. Οι τρόποι για να χωριστεί μία ομάδα n στοιχείων είναι $2^n - 1$ οπότε η πολυπλοκότητα των διαιρετικών αλγορίθμων είναι μεγαλύτερη από αυτή των συσσωρευτικών.

3. Αλγόριθμοι Ανταγωνιστικής Μάθησης

Οι συγκεκριμένοι αλγόριθμοι χρησιμοποιούν ένα σύνολο αντιπροσώπων w_i για i=1,2,....m . Αυτά δίνονται ως είσοδος μαζί με το μέγιστο αριθμό των συστάδων καθώς και το κριτήριο τερματισμού.

Κάθε διάνυσμα των δεδομένων εισόδου κατά την εμφάνισή του διεκδικείται από το σύνολο των αντιπροσώπων w_i, τότε αυτός μετακινείται προς το x_bενώ οι υπόλοιποι

δεν μετακινούνται ή μετακινούνται ελάχιστα. Ο νικητής νευρώνας προκύπτει με βάση την ευκλείδεια απόσταση. Δηλαδή προκύπτει από τη σχέση $|x - w_m|^2$.

2.4 Εφαρμογές Μηχανικής Μάθησης

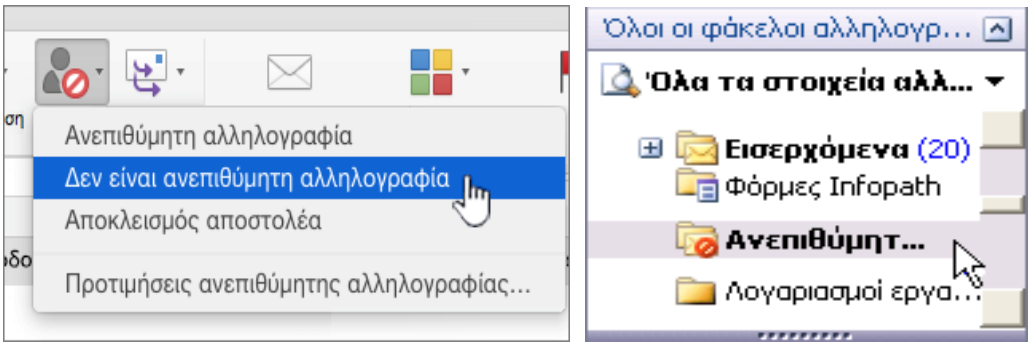
Η Μηχανική Μάθηση βρίσκει μεγάλη εφαρμογή σε πεδία που αφορούν την καθημερινή μας ζωή με αποτέλεσμα να βελτιώνεται η ποιότητα των υπηρεσιών στον κάθε τομέα. Ενδεικτικά αναφέρονται τα εξής:

Λογισμικό - Διαδίκτυο

Σκοπός της μηχανικής μάθησης σε ό,τι αφορά το διαδίκτυο είναι να βελτιώσει την εμπειρία των χρηστών μέσω της αποθήκευσης των προτιμήσεων και της ανάλυσης των δεδομένων που αντλούνται.

- Ανεπιθύμητη αλληλογραφία

Η εκπαίδευση του συστήματος να αντιλαμβάνεται την ανεπιθύμητη αλληλογραφία και να κατατάσσει τα ανάλογα μηνύματα στον αντίστοιχο φάκελο είναι τυπικό παράδειγμα των επιτευγμάτων της μηχανικής μάθησης και βελτιώνει την εμπειρία και την ποιότητα παροχών προς το χρήστη. Ο χρήστης με τη σειρά του έχει την ευκαιρία να αποφασίσει μετά από έλεγχο αν η ταξινόμηση της αλληλογραφίας έγινε σύμφωνα με τις προτιμήσεις του ή να απορρίψει την αρχειοθέτηση αυτή. Με τον τρόπο αυτό το σύστημα <<μαθαίνει>> και αυξάνονται οι πιθανότητες να κάνει σωστή πρόβλεψη αλληλογραφίας στο μέλλον.



Σχ 2.3 Ανεπιθύμητη Αλληλογραφία

- Διαφήμιση - Cookies

Τις τελευταίες δεκαετίες όλο και περισσότερες επιχειρήσεις επιλέγουν να ενημερώσουν και να επηρεάσουν το καταναλωτικό κοινό μέσω διαδικτύου. Η αποθήκευση των προτιμήσεων (αναζητήσεων) των χρηστών είναι μια διαδικασία η οποία δρα ευεργετικά στην προσπάθεια των εταιρειών να διαφημίσουν τα προϊόντα ή τις υπηρεσίες τους. Τα δεδομένα που συλλέγονται συμβάλλουν στην όσο το δυνατό

ακριβέστερη ομαδοποίηση του κοινού και στην προβολή διαφημίσεων ανάλογα με τις προτιμήσεις του.

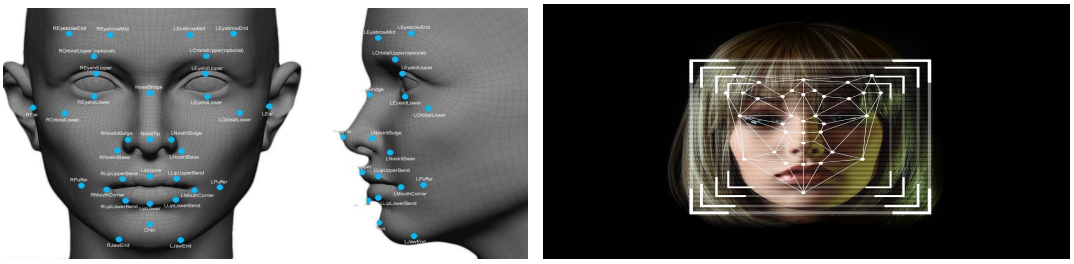
Σχ.2.4 Εφαρμογές που περιέχουν διαφημίσεις στο διαδίκτυο

- Αναγνώριση Προτύπων

Το πρόβλημα της αναγνώρισης προτύπων αφορά την απόδοση τιμών σε κωδικοποιημένα δεδομένα τα οποία, για παράδειγμα, μπορεί να είναι εικόνες ή βίντεο. Για παράδειγμα, το πρόβλημα της αναγνώρισης ενός προσώπου για την είσοδο σε μια υπηρεσία ή για την ταυτοποίηση ενός προσώπου ή για το ξεκλείδωμα ενός κινητού τηλεφώνου, απασχολεί την επιστημονική κοινότητα και αποτελεί αντικείμενο συνεχούς μελέτης προς βελτίωση των υπαρχόντων τεχνικών. Αναλυτικότερα, από μία φωτογραφία το σύστημα καλείται να αναγνωρίσει την ταυτότητα του ατόμου ή να κατηγοριοποιήσει το πρόσωπο σε μία ή περισσότερες κλάσες.

Άλλες εφαρμογές αφορούν την αναγνώριση του λόγου ή της φωνής όπως η φωνητική αναζήτηση στο διαδίκτυο, η φωνητική πληκτρολόγηση, οι εντολές σε μια ρομποτική μηχανή και πολλές ακόμα.

Η αναγνώριση προτύπων βρίσκει εφαρμογή σε πεδία όπως η εγκληματολογική έρευνα, η ασφάλεια των υπηρεσιών, ο έλεγχος στα αεροδρόμια, η ασφάλεια των αντικειμένων των χρηστών (ξεκλείδωμα οθόνης κλπ).



Σχ.2.5 Αναγνώριση προσώπου από τα χαρακτηριστικά

- Αυτόματη μετάφραση κειμένου

Πολλές προσπάθειες γίνονται ώστε η αυτόματη μετάφραση να γίνεται ολοένα και πιο ακριβής και συντακτικά ορθότερη. Στόχος είναι το αυτόματα μεταφρασμένο κείμενο να μη διαφέρει από το κείμενο που προκύπτει από ανθρώπινη μετάφραση με βάση τους γραμματικούς, λεξιλογικούς και συντακτικούς κανόνες των γλωσσών στις οποίες αφορά η μετάφραση. Η λογική είναι το γεγονός ότι χρησιμοποιούνται ήδη μεταφρασμένα κείμενα ώστε να εκπαιδεύεται το σύστημα να μεταφράζει προτάσεις και τελικά ολόκληρο το κείμενο. Το πρόβλημα γίνεται δυσκολότερο όταν το κείμενο που δίνεται αρχικά περιέχει εκφράσεις δύσκολες στην κατανόηση ή συντακτικές αστοχίες.

Σχ. 2.6 Μετάφραση της Google

<u>Υγεία - Ιατρική</u>

Τα τελευταία χρόνια η μηχανική μάθηση παίζει σπουδαίο ρόλο στον τομέα της υγειονομικής περίθαλψης. Η παρουσία της είναι εμφανής και εκτεταμένη σε πτυχές όπως η ανάπτυξη νέων ιατρικών διαδικασιών, η αντιμετώπιση δεδομένων ασθενών και η πρόβλεψη αποτελεσμάτων θεραπειών και χειρουργικών επεμβάσεων. Πιο συγκεκριμένα:

- Πρόβλεψη των αποτελεσμάτων νόσων

Σύμφωνα με μελέτη στο Journal of the American Medical Association (JAMA), ερευνητές από το Ιατρικό Κέντρο του Πανεπιστημίου του Ρότσεστερ και το Πανεπιστήμιο του Ιλλινόις αναφέρουν ότι η μηχανική μάθηση δίνει τη δυνατότητα στους θεράποντες γιατρούς να κάνουν ακριβείς προβλέψεις για την πορεία και τα αποτελέσματα της νόσου του ασθενούς. Οι ερευνητές, με επικεφαλής τον Paul Nierenberg, έχουν αναπτύξει έναν αλγόριθμο που χρησιμοποιεί μια συλλογή δεδομένων, συμπεριλαμβανομένων δεδομένων ασθενών, για την πρόβλεψη της νόσου και των αποτελεσμάτων χειρουργικών επεμβάσεων αξιοποιώντας δεδομένα ασθενών και προγράμματα υπολογισμού των πιθανοτήτων του αποτελέσματος

- Πρόβλεψη αποτελεσμάτων χειρουργικών επεμβάσεων

Η έρευνα για την ανάπτυξη μεθόδων πρόβλεψης αποτελεσμάτων και επιπλοκών χειρουργικών επεμβάσεων καθίσταται πλέον σημαντική καθώς ανοίγει το δρόμο για την εισαγωγή επιπλέον διαδικασιών σε προεγχειρητικό στάδιο ώστε να μειωθούν οι επιπλοκές στην ιατρική κατάσταση του ασθενή.

- Ανάπτυξη νέων φαρμάκων

Η χρήση της μηχανικής μάθησης στην έρευνα ανάπτυξης φαρμάκων κατά το προκαταρκτικό στάδιο δίνει πολλές δυνατότητες από τη εξέταση ενώσεων των φαρμάκων μέχρι το προβλεπόμενο ποσοστό επιτυχίας με βάση βιολογικούς παράγοντες. Μπορεί επίσης δυνητικά να βοηθήσει στην πρόβλεψη της αποτελεσματικότητας και της ασφάλειας νέων φαρμάκων και να παρέχει καλύτερη εικόνα για τις πιθανές ουσίες που μπορούν να χρησιμοποιηθούν για τους εκάστοτε επιθυμητούς σκοπούς.

- Διάγνωση ασθενειών

Δεν είναι λίγες οι περιπτώσεις στις οποίες ένας υπολογιστής προβλέπει το είδος της νόσου ενός ασθενή, τη σοβαρότητά της και την εξέλιξή της. Πολλές φορές χρησιμοποιώντας δεδομένα ασθενών με καρκίνο, προβλέφθηκε με ακρίβεια η καλοήθεια ή μη ενός ευρήματος κατά τη διεξαγωγή μιας βιοψίας. Στην ίδια φιλοσοφία είναι και η διάγνωση καρδιακών ή εγκεφαλικών επεισοδίων με την παρατήρηση, τη σύγκριση και την ανάλυση εξετάσεων (ακτινογραφιών, αξονικών και υπερήχων) από ένα νευρωνικό δίκτυο.

Ρομποτική

Η διαδικασία εκμάθησης (meta learning) μαθαίνει στη μηχανή (robot) να αναπτύσσει δεξιότητες μέσω επαγωγικών μεθόδων βασιζόμενη στην εμπειρία. Η Αναπτυξιακή Μάθηση (Developmental Robotics), η οποία χρησιμοποιείται για την εκπαίδευση μηχανών (robots), δημιουργεί μια ακολουθία μαθησιακών καταστάσεων ώστε το

ρομπότ να αποκτά σταδιακά ποικίλες δεξιότητες μέσω της εξερεύνησης και της κοινωνικής αλληλεπίδρασης με ανθρώπους.Αυτή η εκπαίδευση επιτυγχάνεται με καθοδήγηση η οποία είναι συνήθως ενεργητική μάθηση, η ωρίμανση και η μίμηση. Το ρομπότ μέσω της εκπαίδευσής του μπορεί να εκτελεί κινήσεις, να μετακινείται και να λαμβάνει σήματα.

Οικονομικά

Ο κλάδος των οικονομικών έχει ωφεληθεί ποικιλοτρόπως με την εισαγωγή μεθόδων μηχανικής μάθησης σε πολλές εφαρμογές. Χαρακτηριστικά παραδείγματα είναι οι online πλατφόρμες εμπορίου με στόχο την καλύτερη εξυπηρέτηση των πελατών, η διεξαγωγή προβλέψεων για μελλοντικές οικονομικές καταστάσεις, η πρόβλεψη για την ανάκαμψη ή την πτώχευση μιας εταιρείας, η άνοδος ή η πτώση μιας μετοχής στο χρηματιστήριο, η χαμηλή ή υψηλή ζήτηση ενός προϊόντος ή μιας υπηρεσίας και οι τάσεις του καταναλωτικού κοινού. Στο ηλεκτρονικό εμπόριο, μέσα από την ανάλυση δεδομένων των καταναλωτών παρέχονται πολλές φορές <<κάρτες αφοσίωσης>> σε αυτούς. Μέσα από τη μελέτη των επισκέψεων και των αναζητήσεων στις online πλατφόρμες των εταιρειών, παρέχονται πληροφορίες για τις καταναλωτικές τάσεις. Αυτό έχει ως αποτέλεσμα μέσω αυτών των δεδομένων να προγραμματίζεται η παραγωγή και να προσαρμόζεται η στρατηγική της εταιρείας στην αγορά. Επίσης, με βάση τις προτιμήσεις του κοινού, δημιουργείται ένα σύνολο προτάσεων στους χρήστες με στόχο την καλύτερη ενημέρωση και εξυπηρέτησή τους.

Η χρήση της μηχανικής μάθησης στο χρηματιστήριο, από την άλλη πλευρά, καθίσταται πολύ σημαντική καθώς από την παρατήρηση της εξέλιξης των μετοχών σχεδιάζονται επενδυτικά προγράμματα και λαμβάνονται σημαντικές επιχειρηματικές, οικονομικές και πολιτικές αποφάσεις.

Ηλεκτρονικά Παιχνίδια

Είναι ίσως η δημοφιλέστερη εφαρμογή της τεχνητής νοημοσύνης και της μηχανικής μάθησης. Σύγχρονα ηλεκτρονικά παιχνίδια έχουν τη δυνατότητα να προσαρμόζουν τη συμπεριφορά του παιχνιδιού ανάλογα με τον τρόπο αντίδρασης του παίκτη στο παρελθόν. Τα ηλεκτρονικά παιχνίδια εκτός από τον ψυχαγωγικό του ρόλο, αποτελούν μελετημένα μοντέλα του πραγματικού μας κόσμου. Πολλές τεχνικές έχουν αναπτυχθεί οι οποίες παράγουν προβλέψεις για την έκβαση ενός παιχνιδιού ανάλογα με τις κινήσεις του <<ήρωα>> λαμβάνοντας υπόψη το περιβάλλον, τη συνάρτηση ανταμοιβής και πολλούς άλλους παράγοντες.

Βιομηχανία

Η μηχανική μάθηση χρησιμοποιείται στη βιομηχανία με την έννοια της εκμάθησης μηχανών. Νέες τεχνολογίες κάνουν αισθητή την παρουσία τους στο βιομηχανικό αυτοματισμό και στόχος είναι να ενταχθεί η μηχανική μάθηση σε εργασίες όπως η προγνωστική συντήρηση ώστε να περιορίζεται ο ανθρώπινος χειρισμός. Η βελτιστοποίηση είναι επίσης ένας τομέας όπου η εισαγωγή μηχανικής μάθησης πρόκειται να επιφέρει επανάσταση. Προσπάθειες γίνονται να μοντελοποιηθούν οι διαδικασίες ώστε με χρήση μεθόδων μηχανικής μάθησης να γίνονται προβλέψεις συντήρησης του εξοπλισμού, βελτιστοποίησης της παραγωγικότητας και της αποδοτικότητας μιας μηχανής και ελέγχου.

3. ΕΠΕΞΕΡΓΑΣΙΑ ΚΕΙΜΕΝΟΥ

3.1 Επεξεργασία Φυσικής Γλώσσας (NLP)

Η επεξεργασία της φυσικής γλώσσας είναι μια από τις πιο δημοφιλείς εφαρμογές της τεχνητής νοημοσύνης. Η δυνατότητα επικοινωνίας του ανθρώπου με τον υπολογιστή σε ανθρώπινη γλώσσα είναι ιδιαίτερη πρόκληση εδώ και πολλά χρόνια. Ωστόσο η φύση της γλώσσας την καθιστά πολλές φορές διφορούμενη με αποτέλεσμα να είναι εξαιρετικά δύσκολη η κατανόησή της και η εξαγωγή νοήματος από αυτή. Συνεπώς η επεξεργασία της φυσικής γλώσσας αποτελεί πρόκληση για την εξέλιξη της τεχνητής νοημοσύνης και πολλές μέθοδοι έχουν επινοηθεί ώστε να αντιμετωπιστούν οι δυσκολίες στην κατανόηση και την ερμηνεία της.

Η χρήση της επεξεργασίας φυσικής γλώσσας εντοπίζεται σε διάφορους τομείς όπως η επικοινωνία ανθρώπου-μηχανής (human - computer interaction). Οι χρήστες μπορούν να επικοινωνούν με τη μηχανή στη γλώσσα τους (ή στην αγγλικη) με τη μηχανή και όχι μέσω κάποιας γλώσσας προγραμματισμού ή μενού επιλογών. Μία άλλη περιοχή που χρησιμοποιεί την επεξεργασία φυσικής γλώσσας είναι η διαχείριση πληροφορίας (information management) όπου επεξεργάζονται και ερμηνεύονται οι πληροφορίες που αντλούνται ώστε να εξάγει συμπεράσματα. Για παράδειγμα, η αρχειοθέτηση ή η ταξινόμηση εγγράφων είναι εφαρμογές στις οποίες χρησιμοποιείται η επεξεργασία φυσικής γλώσσας. Τρίτη περιοχή είναι η αναζήτηση σε βάσεις δεδομένων (database searching). Οι πληροφορίες που αναζητούνται σε βάσεις δεδομένων εκφράζονται με μενού επιλογών, με λίστες ή με μορφές σε δομημένη τεχνητή γλώσσα. Το γεγονός αυτό δίνει τη δυνατότητα ανάπτυξης πολλών μηχανών αναζήτησης, αλλά απαιτεί την εξοικείωση του χρήστη με την τεχνητή γλώσσα και τη δομή της βάσης. Αντίθετα, οι χρήστες είναι περισσότερο εξοικειωμένοι με το περιεχόμενο της βάσης και όχι τόσο με τη δομή της. Επομένως, με την επεξεργασία φυσικής γλώσσας τα αιτήματα αναζήτησης προσανατολίζονται στο περιεχόμενο της βάσης.

Η διφορούμενη φύση της φυσικής γλώσσας την καθιστά πολύ δύσκολη στην κατανόηση και την ερμηνεία καθώς δημιουργεί σημαντικές ασάφειες. Οι ασάφειες αυτές εντοπίζονται σε πολλά επίπεδα. Ενδεικτικά αναφέρονται τα εξής:

- Συντακτικές ασάφειες: Μια συντακτικά ορθή πρόταση επιδέχεται πολλές πιθανές ερμηνείες όπως για παράδειγμα η πρόταση:

  Χτύπησα τον κλέφτη με το τσεκούρι

  Δεν είναι σαφές αν χρησιμοποίησα το τσεκούρι ως όπλο ή αν ο κλέφτης κρατούσε τσεκούρι.

- Λεξιλογικές ασάφειες: Στην πρόταση *Το πρώτο γράμμα του Γιώργου* η λέξη <<γράμμα>> μπορεί να αναφέρεται είτε στο γράμμα που γράφει ο Γιώργος είτε στο σύμβολο του αλφαβήτου με το οποίο αρχίζει το όνομα Γιώργος.
- Αναφορικές ασάφειες: Τέτοιου είδος ασάφεια παρατηρείται όταν δεν είναι ξεκάθαρο σε ποιόν αναφέρεται το καθετί. Για παράδειγμα στην πρόταση:

  Ο Γιάννης χτύπησε το Γιώργο γιατί του αρέσει η Μαίρη.

  Δεν είναι σαφές αν η Μαίρη αρέσει στο Γιάννη ή στο Γιώργο.

- Σημασιολογικές ασάφειες: Μια πρόταση με την ίδια σύνταξη επιδέχεται δύο ή περισσότερες διαφορετικές ερμηνείες. Για παράδειγμα η λέξη <<*φακός*>> μπορεί να αναφέρεται είτε στη Φωτογραφία, είτε στη Φυσική είτε στην Ανατομία.
- Πραγματολογικές ασάφειες: Για τη διερμηνεία μιας πρότασης πολλές φορές είναι απαραίτητο να λάβουμε υπόψη και το κείμενο το οποίο την περιέχει. Για παράδειγμα στη φράση:

  Οι δεινόσαυροι εξαφανίστηκαν πριν πολλά χρόνια.

  Δεν είναι σαφές πόσα είναι τα χρόνια εξαφάνισης των δεινοσαύρων.

Οι ασάφειες που υπάρχουν στις προτάσεις της φυσικής γλώσσας αποτελούν σημαντικό πρόβλημα στην εξέλιξη της αυτόματης αναγνώρισης όρων και επομένως στην επεξεργασία της φυσικής γλώσσας. Για το λόγο αυτό απασχολούν ιδιαίτερα τους επιστήμονες που ασχολούνται με την αυτόματη αναγνώριση όρων.

Περισσότερες πληροφορίες παρέχονται στο άρθρο ("Επεξεργασία και Κατανόηση Φυσικής Γλώσσας")

3.2 Ιστορικά στοιχεία

Κατά τη δεκαετία 1940-1950 η έρευνα επικεντρώθηκε στην ανάπτυξη μεθόδων αυτοματοποίησης και χρήσης πιθανολογικών μοντέλων. Τη δεκαετία του 1950 ο Turing με το μοντέλο του για αλγοριθμικούς υπολογισμούς εισήγαγε την

αυτοματοποίηση ως διαδικασία. Μετά τον Turing ακολούθησαν οι Mc Culloch Pitts νευρώνες οι οποίοι μπορούσαν να περιγράψουν εκφράσεις προτασιακής λογικής. Έπειτα ο Kleene στην ίδια λογική δημιούργησε τα πεπερασμένα αυτόματα και τις κανονικές εκφράσεις. Ο Shannon με τη σειρά του χρησιμοποίησε τα πιθανολογικά μοντέλα των διαδικασιών Markov ώστε να μετατρέψει την επεξεργασία φυσικής γλώσσας σε αυτοματοποιημένη διαδικασία. Ο Chomsky βασιζόμενος στη μελέτη του Shannon εισήγαγε την ιδέα των μηχανών πεπερασμένων καταστάσεων για την αναγνώριση μιας γλώσσας. Οι μηχανές πεπερασμένων καταστάσεων ήταν ο τρόπος για να χαρακτηριστεί μια γραμματική και να οριστεί μια πεπερασμένων καταστάσεων γλώσσα ως γλώσσα που παράγεται από μια τέτοια γραμματική. Στα μοντέλα αυτά χρησιμοποιείται κυρίως άλγεβρα και θεωρία συνόλων για τον καθορισμό τυπικών γλωσσών ως ακολουθίες συμβόλων.

Αξιοσημείωτες είναι επίσης οι έρευνες γλωσσολόγων και επιστημόνων της πληροφορικής σχετικά με αλγορίθμους ανάλυσης της γλώσσας την ίδια δεκαετία. Τέτοιοι αλγόριθμοι χρησιμοποιούσαν λογικές όπως η από πάνω προς τα κάτω (top - down) και η από κάτω προς τα πάνω (bottom - up) και σε δεύτερο χρόνο η λογική του δυναμικού προγραμματισμού. Ένα από τα πρώτα ολοκληρωμένα συστήματα ανάλυσης λόγου ήταν το πρότζεκτ <<Μετασχηματισμοί και Ανάλυση Λόγου>> (Transformations and Discourse Analysis Project). Στα μέσα της δεκαετίας του 1960 αναπτύχθηκε το σύστημα ELIZA. Στο σύστημα αυτό ο Weizenbaum βασιζόμενος στα πεπερασμένα αυτόματα (Finite State Automata, FSA) πέτυχε την αναγνώριση λεκτικών οντοτήτων (λέξεων και άλλων στοιχείων του λόγου). Την ίδια δεκαετία ο Roger Shrank και η ομάδα του δημιούργησαν μια σειρά από μεθόδους κατανόησης ανθρώπινης γλώσσας που κεντρικό άξονα είχαν την ανθρώπινη εννοιολογική γνώση όπως οι ενέργειες, τα σχέδια, οι στόχοι και η οργάνωση της ανθρώπινης μνήμης. Τη διετία 1968-1970 παρουσιάστηκε έκρηξη στην έρευνα για την επεξεργασία φυσικής γλώσσας με την ανάπτυξη του SHRDLU. Το SHRDLU είναι ένα πρώιμο πρόγραμμα σε υπολογιστή για την κατανόηση φυσικής γλώσσας που αναπτύχθηκε από τον Terry Winograd στο MIT την περίοδο 1968-1970. Στο πρόγραμμα αυτό, ο χρήστης πραγματοποιεί μια συνομιλία με τον υπολογιστή μετακινώντας αντικείμενα. Γράφτηκε σε γλώσσα προγραμματισμού Lisp και αποτελούσε κυρίως πρόγραμμα ανάλυσης γλωσσών που επέτρεπε την αλληλεπίδραση των χρηστών χρησιμοποιώντας αγγλικούς όρους. Η επιτυχία του SHRDLU ήταν τόσο μεγάλη που έδειξε ότι η ανάλυση είναι αρκετά κατανοητή ώστε να δημιουργηθεί ενδιαφέρον για τη σημασιολογία και τα λεκτικά μοντέλα.

Στις αρχές του 1970 ο Montague μελέτησε την αρχή της συνθετικότητας για να αναλύσει τις προτάσεις της φυσικής γλώσσας επιτυχώς. Ήδη από τα τέλη του 19ου αιώνα έως και τις αρχές του 20ου αιώνα, ήταν γνωστή από το Frege η αρχή της συνθετικότητας των προτάσεων (Αρχή του Frege). Αυτό σημαίνει ότι η σημασία μιας πρότασης είναι συνθετική και αποτελείται από τη σημασία των επιμέρους εκφράσεων που περιέχει (compositional semantics).

Το 1977 οι Schank και Abelson έπαιξαν σημαντικό ρόλο στην κατανόηση της φυσικής γλώσσας εφόσον εισήγαγαν την αναπαράσταση κειμένου μέσω σεναρίων (scripts) τα οποία παρουσιάζουν την αιτιότητα και την αλληλουχία των γεγονότων μέσα σε αυτό. Η θεωρία των σεναρίων υποστηρίζει ότι η ανθρώπινη συμπεριφορά βασίζεται σε ορισμένα πρότυπα που ονομάζονται σενάρια καθώς ακολουθούν ένα γραμμένο σενάριο το οποίο παρέχει προγράμματα δράσης. Οι Schank και Abelson χρησιμοποιήσαν τη θεωρία αυτή στην τεχνητή νοημοσύνη ως μέθοδο αναπαράστασης διαδικαστικών γνώσεων. Ένα τυπικό παράδειγμα σεναρίου είναι η ακολουθία ενεργειών που λαμβάνουν χώρα όταν ένα άτομο επισκέπτεται ένα εστιατόριο ώστε να πιει ένα ποτό. Οι κινήσεις είναι: *εύρεση θέσης, ανάγνωση μενού, παραγγελία ποτού από τη σερβιτόρα.* Οι κινήσεις αυτές αποτελούν κατά βάση εννοιολογικές μεταβάσεις, όπως MTRANS και PTRANS οι οποίες μπορεί να είναι είτε νοητικές (πληροφοριών) είτε φυσικές (πραγμάτων). Το εγχείρημα των Schank και Abelson ήταν καινοτόμο για την περίοδο εκείνη και μείζονος σημασίας καθώς πραγματευόταν ένα από τα μεγαλύτερα προβλήματα της τεχνητής νοημοσύνης, αυτό της κατανόησης ιστορίας. Τελικά το εγχείρημα αυτό έληξε χωρίς απτή επιτυχία, ωστόσο στάθηκε πολύ σημαντικό για τη μετέπειτα έρευνα στον τομέα της επεξεργασίας φυσικής γλώσσας.

Στην σύγχρονη πληροφορική, η επεξεργασία φυσικής γλώσσας χρησιμοποιεί ως βασικό εργαλείο τα πιθανολογικά μοντέλα και τα μοντέλα δεδομένων. Πλέον, η ανάλυση και η επεξεργασία του λόγου έχουν εντάξει τις πιθανότητες και χρησιμοποιούν στρατηγικές εκπαίδευσης και αξιολόγησης οι οποίες είναι ήδη γνωστές από εφαρμογές στην αναγνώριση ομιλίας και της ανάκτησης πληροφοριών.

Τα αποτελέσματα της επεξεργασίας και κατανόησης του λόγου είναι εντυπωσιακά με τα σύγχρονα μοντέλα, αλλά στις απαρχές του κλάδου τα μοντέλα ήταν πολύ περιορισμένων δυνατοτήτων και τα αποτελέσματα ήταν στοιχειώδη. Για παράδειγμα, το πρόγραμμα ELIZA το οποίο προσομοιώνει την επικοινωνία ενός ψυχιάτρου με έναν ασθενή, πραγματοποιεί αναγνώριση μεμονωμένων λεκτικών μονάδων με αποτέλεσμα να προκαλεί τυποποιημένες απαντήσεις. Το ρόλο του ψυχιάτρου παίζει το πρόγραμμα ELIZA. Παρακάτω φαίνεται η συνομιλία αυτή.

Σχ.3.1 Πρόγραμμα ELIZA

Πηγή: https://en.wikipedia.org/wiki/ELIZA

Οι απαντήσεις (έξοδος) που παράγει το πρόγραμμα ELIZA βασίζονται σε ένα σύνολο περιγραμμάτων (templates). Για παράδειγμα, περιγράμματα μπορούν να είναι οι φράσεις: ΤΙ ΣΕ ΚΑΝΕΙ ΝΑ ΠΙΣΤΕΥΕΙΣ ΟΤΙ XXX, ΠΕΣ ΜΟΥ ΠΕΡΙΣΣΟΤΕΡΑ ΓΙΑ XXX, ΣΕ ΕΥΧΑΡΙΣΤΕΙ ΝΑ ΠΙΣΤΕΥΕΙΣ ΟΤΙ XXX;

Η πρόταση εισόδου εξετάζεται από πεπερασμένα αυτόματα ως προς ορισμένες λέξεις ή φράσεις κλειδιά που μπορεί να περιέχει.

Κάθε λέξη-κλειδί της εισόδου μέσα από ένα σύνολο κανόνων παράγει από τη φράση εισόδου μια φράση εξόδου. Για παράδειγμα, ένας τέτοιος κανόνας θα μπορούσε να είναι ο εξής:

ΕΣΥ ΔΕΝ XXX ΜΕ ΜΕΝΑ -> ΕΓΩ ΔΕΝ XXX ΜΕ ΣΕΝΑ

<u>Παράδειγμα εφαρμογής:</u>

- ΔΕ ΔΙΑΦΩΝΕΙΣ ΜΑΖΙ ΜΟΥ
- ΤΙ ΣΕ ΚΑΝΕΙ ΝΑ ΠΙΣΤΕΥΕΙΣ ΟΤΙ ΔΕ ΔΙΑΦΩΝΩ ΜΑΖΙ ΣΟΥ;

Περισσότερες πληροφορίες παρέχονται στα άρθρα: (“Επεξεργασία και Κατανόηση Φυσικής Γλώσσας”) (Νταλιακούρας) (“SHRDLU”)

3.3 Κατανόηση Φυσικής Γλώσσας

Η Κατανόηση Φυσικής Γλώσσας (NLU) ορίζεται από τον Gartner ως η κατανόηση από τους υπολογιστές της δομής και της έννοιας της ανθρώπινης γλώσσας, επιτρέποντας στους χρήστες να επικοινωνούν με τον υπολογιστή με ανθρώπινες εκφράσεις. Επομένως, η NLU είναι η Τεχνητή Νοημοσύνη η οποία χρησιμοποιεί λογισμικό για την ερμηνεία κειμένου και κάθε είδους μη δομημένων δεδομένων. Η NLU μπορεί να αφομοιώσει ένα κείμενο, να το μετατρέψει σε γλώσσα υπολογιστή και να παράγει μια έξοδο οικεία στο χρήστη.

Η κατανόηση φυσικής γλώσσας ερμηνεύει την έννοια την οποία ο χρήστης επικοινωνεί και την ταξινομεί σε κατάλληλες προθέσεις. Για παράδειγμα, στην επικοινωνία μεταξύ ανθρώπων που μιλούν την ίδια γλώσσα είναι εύκολο να καταλάβει ο ένας τον άλλον, παρά τις διαφορές στο λεξιλόγιο ή στις εκφράσεις, τα οποία εισάγουν ασάφειες. Ωστόσο στην επικοινωνία ανθρώπου - υπολογιστή, η διαδικασία αυτή δεν είναι το ίδιο εύκολη. Η ερμηνεία του λόγου στην περίπτωση αυτή είναι αρμοδιότητα του NLU εφαρμόζοντας μια σειρά διαδικασιών όπως κατηγοριοποίηση κειμένου, ανάλυση περιεχομένου και ανάλυση συναισθημάτων.

Η κατανόηση φυσικής γλώσσας στοχεύει:

α) στη θεωρητική έρευνα της γλώσσας, δηλαδή η εξέταση των λεπτομερειών της γλώσσας για εφαρμογή από τον υπολογιστή

β) στην επίτευξη όλο και πιο βελτιωμένης επαφής ανθρώπου- υπολογιστή

Στο παρακάτω σχήμα φαίνεται σε διάγραμμα η παραπάνω διαδικασία:

Σχ.3.2 Επικοινωνία ανθρώπου - υπολογιστή

Πηγή: https://repository.kallipos.gr/bitstream/11419/3385/1/02_chapter_07.pdf

Η κατανόηση φυσικής γλώσσας εξαρτάται σε μεγάλο βαθμό από τον τρόπο με τον οποίο αναπαρίσταται και εκφράζεται η πληροφορία. Τα στάδια της κατανόησης είναι κυρίως τα εξής:

- Αναγνώριση φωνής
- Συντακτική Ανάλυση
- Σημασιολογική Ανάλυση
- Πραγματολογική Ανάλυση
- Παραγωγή φυσικής γλώσσας

Κάποιες προτάσεις είναι καλοσχηματισμένες και κάποιες είναι κακοσχηματισμένες σε συντακτικό, σημασιολογικό ή πραγματολογικό επίπεδο. Το γεγονός αυτό καθιστά τη διαδικασία επεξεργασίας της φυσικής γλώσσας ιδιαίτερα δύσκολη. Παραδείγματα τέτοιων προτάσεων είναι τα εξής:

Υποθέτοντας ότι έχουμε την ερώτηση *<<Που βρίσκεται το σχολείο;>>* . Πιθανές απαντήσεις που μπορεί κανείς να λάβει είναι οι εξής:

- *Το σχολείο βρίσκεται πίσω από το Δημαρχείο.*

  Η πρόταση αυτή είναι συντακτικά, σημασιολογικά και πραγματολογικά άρτια.

- *Το σχολείο βρίσκεται στο τέλος του ραντεβού*.

  Η πρόταση μπορεί να είναι συντακτικά ορθή, όμως δε μπορεί να σταθεί σημασιολογικά γιατί το περιεχόμενό της δεν έχει κανένα νόημα.

- *Το σχολείο δημαρχείου είναι.*

  Η παραπάνω πρόταση είναι συντακτικά λανθασμένη καθώς δεν πληροί τους κανόνες σύνταξης.

- *Βεβαίως.*

  Η παραπάνω περιεκτική πρόταση μπορεί να μην παραβιάζει κάποιο συντακτικό ή σημασιολογικό κανόνα, ωστόσο είναι πραγματολογικά λανθασμένη διότι δεν απαντάει στο αρχικό ερώτημα.

Το σύστημα χρησιμοποιεί το λεξικό της γλώσσας για την επεξεργασία της. Λόγω του μεγάλου πλήθους των λέξεων, το σύστημα συγκρατεί μόνο τη ρίζα κάθε λέξης και δημιουργεί ή αναγνωρίζει τις υπόλοιπες λέξεις με τη βοήθεια της μορφολογικής ανάλυσης.

3.3.1 Συντακτική Ανάλυση

Η συντακτική ανάλυση (syntactic analysis) ασχολείται με τη δομή μιας πρότασης και σκοπεύει να αποφανθεί μέσα από κανόνες αν η πρόταση είναι ορθή ή λανθασμένη.

Μια απλή και λογική προσέγγιση, την οποία χρησιμοποίησε και το ELIZA, είναι η χρήση <<ταιριάσματος>> προτύπων. Ένα τέτοιο πρότυπο για παράδειγμα θα

μπορούσε να είναι : << *Ο XXX γράφει YYY*>> , όπου *XXX, YYY* οποιεσδήποτε μεταβλητές. Ωστόσο η λογική αυτή δεν απέδιδε ικανοποιητικά γιατί αδυνατούσε να εντοπίσει προτάσεις σημασιολογικά λανθασμένες. Για παράδειγμα η έκφραση <<*Το αυτοκίνητο γράφει στον τοίχο*>> με βάση τη λογική των προτύπων είναι αποδεκτή, όμως είναι σημασιολογικά κακοσχηματισμένη.

Περισσότερο αποτελεσματική μέθοδος αποδείχθηκε η **γραμματική ανάλυση** με βάση την οποία η πρόταση μετατρέπεται σε ιεραρχική δομή η οποία δηλώνει τα συστατικά της. Η μέθοδος αυτή περιέχει:

- Μια γραμματική η οποία παρέχει μια αναπαράσταση των συντακτικών στοιχείων της γλώσσας και
- Το συντακτικό αναλυτή ο οποίος συγκρίνει τη λέξη που λαμβάνει με τη γραμματική.

Εκτός από τη γραμματική ανάλυση, η συντακτική ανάλυση μπορεί να επιτευχθεί με διαγράμματα μετάβασης (transition networks) και επαυξημένα διαγράμματα μετάβασης (augmented transition netwotks).

Αν ορισμένες λέξεις ή φράσεις είναι διφορούμενες τότε πραγματοποιείται συντακτική ανάλυση όλων των πιθανών ερμηνειών των λέξεων από το λεξικό καταλήγοντας σε όσες προσεγγίζουν επαρκώς τη συντακτική, τη σημασιολογική και την πραγματολογική ανάλυση της πρότασης.

Γραμματική

Μια γραμματική προσδιορίζει τις αποδεκτές μορφές μιας γλώσσας. Περιλαμβάνει ένα σύνολο κανόνων της μορφής A → B, με την έννοια ότι κάθε στοιχείο A μπορεί να αντικατασταθεί από το στοιχείο B. Μια γραμματική αποτελείται από τα εξής στοιχεία:

- τα τερματικά σύμβολα
- τα μη τερματικά σύμβολα
- τους κανόνες παραγωγής

Τα τερματικά σύμβολα είναι οι λέξεις που περιέχει η γλώσσα των οποίων προσδιορίζεται υποχρεωτικά και ο τύπος. Για παράδειγμα η λέξη (τερματικό) <<αυτοκίνητο>> είναι ουσιαστικό.

Τα μη τερματικά σύμβολα είναι ειδικά σύμβολα τα οποία δηλώνουν τις δομές μιας γλώσσας. Υπάρχουν τρεις τύποι μη τερματικών συμβόλων:

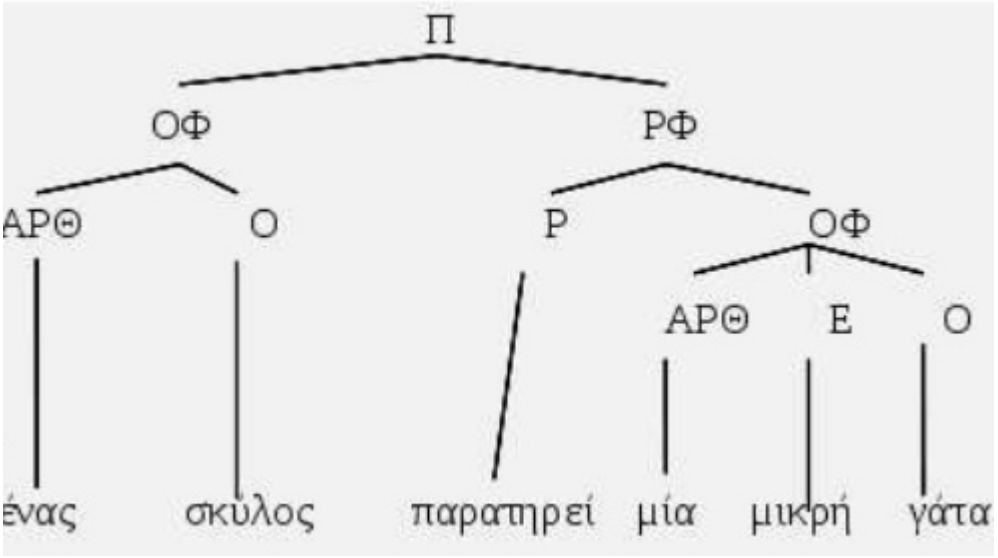
- ❏ Οι λεκτικές κατηγορίες (πχ ρήμα, ουσιαστικό κλπ.)
- ❏ Οι συντακτικές κατηγορίες (πχ αντικειμενική πρόταση, κατηγορηματική πρόταση)

- ❑ Το αρχικό σύμβολο το οποίο αντιπροσωπεύει την αρχική πρόταση που περιγράφεται γραμματικά.

Οι κανόνες παραγωγής είναι το σύνολο των κανόνων που αντιπροσωπεύουν τους κατάλληλους συνδυασμούς λέξεων για την παραγωγή μιας πρότασης και έχουν τη μορφή Α → ΟΦ/ΡΦ , όπου Α το αρχικό σύμβολο που ξεκινάει η πρόταση, ΟΦ η ονοματική φράση (πχ Ο Κώστας) και ΡΦ η ρηματική φράση (πχ γράφει την επιστολή).

<u>Συντακτικά δένδρα</u>

Ένα συντακτικό δένδρο αντιπροσωπεύει την ιεραρχική διάσπαση της πρότασης στα συστατικά της μέρη. Η ρίζα του συντακτικού δένδρου είναι το αρχικό σύμβολο, κάθε ενδιάμεσος κόμβος παριστάνει ένα μη τερματικό σύμβολο της γραμματικής και κάθε φύλλο επισημειώνεται με τερματικό σύμβολο. Ένα παράδειγμα συντακτικού δέντρου φαίνεται παρακάτω:



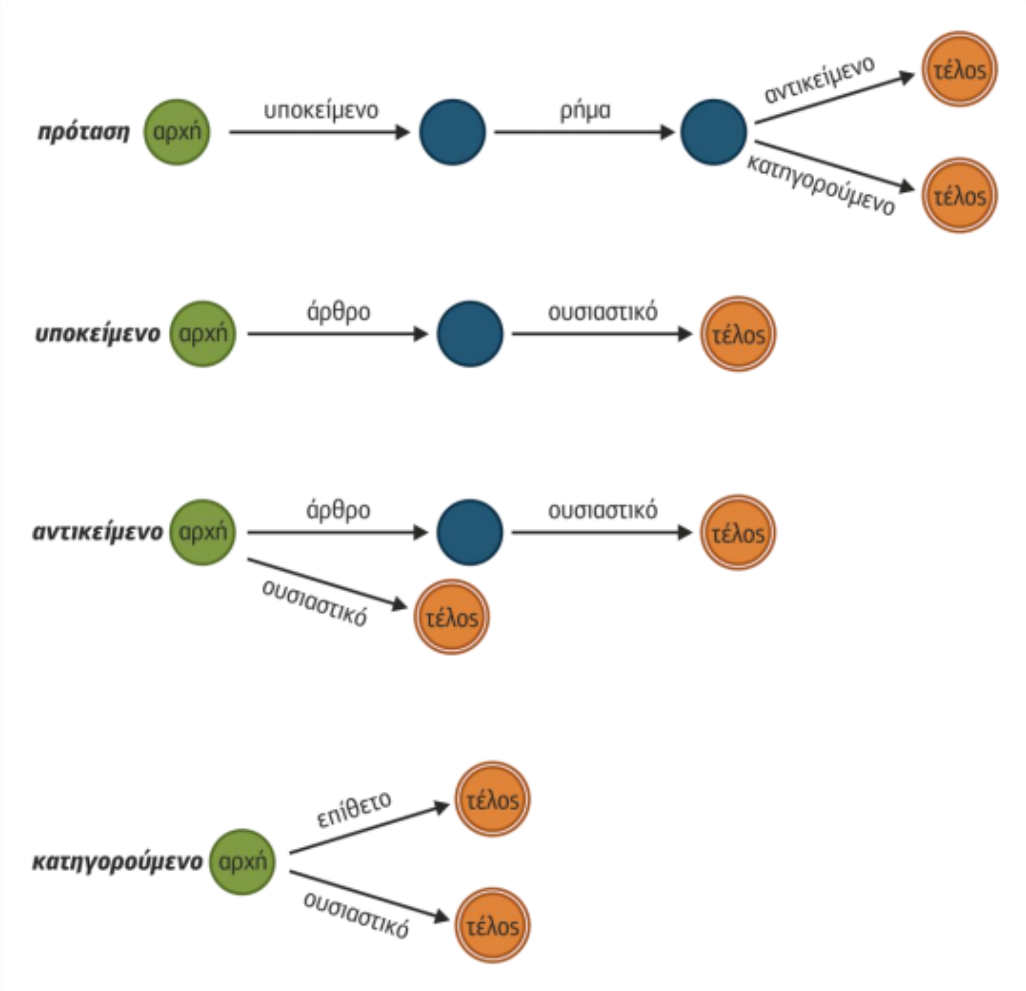
Σχ.3.3 Παράδειγμα Συντακτικού Δέντρου

Πηγή: https://www.slideshare.net/ourpal/ss-43255432

Ο συντακτικός αναλυτής αναπτύσσει το δέντρο με τη βοήθεια των κανόνων παραγωγής της γραμματικής μέχρι να φτάσει στα φύλλα, δηλαδή σε τερματικά σύμβολα.

Διαγράμματα μεταβάσεων

Τα διαγράμματα μετάβασης (transition networks) εκφράζουν τη γραμματική σαν ένα σύνολο μηχανών πεπερασμένων καταστάσεων. Οι μηχανές αυτές ονομάζονται αυτόματα πεπερασμένων καταστάσεων, στα οποία κάθε κόμβος αναπαριστά μια εσωτερική κατάσταση και κάθε βέλος τον τρόπο με τον οποίο η μηχανή μεταβαίνει από τη μία κατάσταση στην άλλη. Στην ανάλυση μιας γλώσσας, τα βέλη αντιπροσωπεύουν είτε τερματικά είτε μη τερματικά σύμβολα, ενώ οι κανόνες της γραμματικής αντιστοιχούν σε μία διαδρομή μέσα στο διάγραμμα μεταβάσεων. Για παράδειγμα, για τη συντακτική ανάλυση μιας πρότασης δημιουργούνται τα παρακάτω διαγράμματα μεταβάσεων τα οποία παριστάνουν τους κανόνες της γραμματικής:



Σχ.3.4 Διαγράμματα μεταβάσεων συντακτικής ανάλυσης πρότασης

Πηγή: https://repository.kallipos.gr/bitstream/11419/3385/1/02_chapter_07.pdf

Αν πρόκειται να αναλυθεί η πρόταση *Ο Κώστας πίνει τσάι,* τότε το σύστημα ξεκινάει την ανάλυση από το αρχικό σύμβολο, δηλαδή την πρόταση. Τότε, περιμένει να δει άρθρο ώστε να προχωρήσει στον επόμενο κόμβο. Το τερματικό σύμβολο "ο" είναι άρθρο, οπότε προχωράει με τη διάσχιση στον επόμενο κόμβο όπου περιμένει να δει ουσιαστικό. Το σύμβολο "Κώστας" αναγνωρίζεται ως ουσιαστικό οπότε η διάσχιση του υποκείμενου ολοκληρώνεται με επιτυχία και η διάσχιση συνεχίζεται από το αρχικό διάγραμμα. Ακολουθεί η αναζήτηση του μη τερματικού συμβόλου του ρήματος. Η λέξη "πίνει" αναγνωρίζεται ως ρήμα με βάση το λεξικό και η διάσχιση συνεχίζεται αναζητώντας αντικείμενο ή κατηγορούμενο. Τελικά με το ουσιαστικό "τσάι" διασχίζεται το διάγραμμα του αντικειμένου και η ανάλυση της πρότασης ολοκληρώνεται.

3.3.2 Σημασιολογική Ανάλυση

Κατά τη σημασιολογική ανάλυση (semantic analysis) οι προτάσεις μετατρέπονται σε εσωτερικές δομές αναπαράστασης της γνώσης στις οποίες χρησιμοποιείται το νόημα των λέξεων. Για την ανάλυση αυτή χρησιμοποιούνται εξελιγμένες γραμματικές οριστικών προτάσεων.

Το ζήτημα της ασάφειας παρουσιάζεται έντονα και σε αυτό το είδος της ανάλυσης καθώς οι περισσότερες λέξεις της γλώσσας μπορούν να λάβουν πολλαπλές ερμηνείες (πολυσημία). Μια πιθανή λύση στο πρόβλημα αυτό είναι να λαμβάνονται υπόψη τα χαρακτηριστικά των ουσιαστικών και των ρημάτων της πρότασης. Για παράδειγμα, στα ουσιαστικά απαιτείται η εξέταση του γένους, του αριθμού και της πτώσης και στα ρήματα το πρόσωπο.

Ένα παράδειγμα μιας εξελιγμένης γραμματικής η οποία λαμβάνει υπόψη τα χαρακτηριστικά των λέξεων είναι η εξής:

πρόταση → υποκείμενο(γένος, αριθμός, ονομαστική) ρήμα(αριθμός) αντικείμενο(γένος, αριθμός, αιτιατική)

πρόταση → υποκείμενο(γένος, αριθμός, ονομαστική) ρήμα(αριθμός) κατηγορούμενο(γένος, αριθμός, ονομαστική)

υποκείμενο → άρθρο(γένος, αριθμός,πτώση) ουσιαστικό(γένος, αριθμός, πτώση)

κατηγορούμενο → επίθετο(γένος, αριθμός, πτώση)

Οι κανόνες οι οποίοι θα μπορούσαν να διέπουν μια τέτοια γραμματική θα μπορούσαν να είναι:

- Ο αριθμός του ουσιαστικού του υποκειμένου να συμφωνεί με την κατάληξη του ρήματος.
- Το γένος, η πτώση και ο αριθμός του άρθρου του υποκειμένου να συμφωνεί με τα χαρακτηριστικά του ουσιαστικού του υποκειμένου.

Επίσης, κανόνες υπάρχουν και για την εν μέρη αντιμετώπιση της ασάφειας είτε σε σημασιολογικό είτε σε πραγματολογικό επίπεδο χωρίς όμως αυτοί να επιλύουν εξ ολοκλήρου το πρόβλημα.

Για παράδειγμα ο κανόνας ότι ο χρονικός προσδιορισμός συνδέεται με το κοντινότερο ρήμα μπορεί να άρει την ασάφεια στην παρακάτω πρόταση:

Η Μαίρη μίλησε στη μητέρα της για την εργασία που πρέπει να παραδώσει σήμερα.

Τέλος, η πολυσημία των λέξεων δημιουργεί αμφιβολία στην ερμηνεία μιας πρότασης. Κατά κανόνα, μεγαλύτερη πιθανότητα έχει μια λέξη να εμφανιστεί με την πρώτη της σημασία σε μία πρόταση. Επομένως, κύριο μέλημα του συστήματος είναι να λάβει υπόψη τις πιθανότητες εμφάνισης των λέξεων με την κάθε τους σημασία.

3.3.3 Πραγματολογική Ανάλυση

Στόχος της πραγματολογικής ανάλυσης είναι η κατανόηση κειμένων και ο χειρισμός διαλόγων. Κατά τη διαδικασία αυτή επιχειρείται η σύνδεση της κάθε πρότασης με το γενικότερο νόημα του περιεχομένου των συμφραζομένων. Συνήθως δυσκολίες προκύπτουν όταν υπάρχουν αντωνυμίες, όπως για παράδειγμα:

Την κάλεσαν….

Σε ένα κείμενο επίσης δυσκολίες σε πραγματολογικό επίπεδο προκύπτουν από τμήματα αντικειμένων που λείπουν όπως στην πρόταση:

Η Ελένη άνοιξε το βιβλίο που μόλις είχε αγοράσει. Η πρώτη σελίδα ήταν σκισμένη.

Εναλλακτικά μπορεί να λείπουν τμήματα ενεργειών όπως στην πρόταση:

Ο Γιάννης πήγε ταξίδι στην Κρήτη. Έφυγε με την πρωινή πτήση.

Για να λυθούν τα παραπάνω προβλήματα πρέπει το σύστημα να έχει τη δυνατότητα να αντιλαμβάνεται το περιεχόμενο και το νόημα του κειμένου που λαμβάνει. Πρέπει επίσης να λαμβάνει υπόψη τις πιθανότητες που έχουν κάποια γεγονότα να εμφανιστούν ή ορισμένα πιθανά σενάρια (scripts) που μπορεί να συμβούν.

Με βάση τα παραπάνω που λαμβάνει υπόψη το σύστημα μπορεί να καταλήξει σε συμπεράσματα τα οποία ανταποκρίνονται σε μεγάλο βαθμό στην πραγματικότητα. Η διαδικασία κατανόησης του νοήματος του κειμένου από το σύστημα είναι απολύτως απαραίτητη για τη πραγματολογική ανάλυση.

Περισσότερες πληροφορίες παρέχονται στο άρθρο (“Επεξεργασία και Κατανόηση Φυσικής Γλώσσας” #)

3.4 Πεδία Έρευνας της Επεξεργασίας Φυσικής Γλώσσας

Η ανάγκη επίλυσης ζητημάτων σε πολλούς τομείς που σχετίζονται με την χρήση δεδομένων κειμένων οδήγησε στην εφαρμογή της επεξεργασίας φυσικής γλώσσας. Προβλήματα που απαιτούν επίλυση μπορεί να προέρχονται από την καθημερινή ζωή ή από την επίλυση μεγαλύτερων και περισσότερο σύνθετων προβλημάτων. Για το λόγο αυτό διακρίνονται ορισμένα πεδία έρευνας της NLP τα οποία είναι διακριτά ως προς τη φύση των δεδομένων που χρησιμοποιούν, τους τρόπους αξιολόγησης και τα ζητήματα που τίθενται προς επίλυση. Ορισμένα πεδία έρευνας της επεξεργασίας φυσικής γλώσσας είναι τα εξής:

- Ανάλυση Λόγου: Στο συγκεκριμένο πεδίο έρευνας πραγματοποιούνται ποικίλες μελέτες. Ορισμένες αφορούν τη δομή του λόγου ενός κειμένου, δηλαδή τις σχέσεις μεταξύ των προτάσεων, και ορισμένες μπορεί να αφορούν την αναγνώριση και την κατηγοριοποίηση γλωσσικών πράξεων σε ένα κομμάτι κειμένου.
- Αυτόματη αναγνώριση ομιλίας: Ο εντοπισμός του ανθρώπινου λόγου από τον υπολογιστή και η μετατροπή του σε κείμενο.
- Αυτόματη ερωταπόκριση: Η αναζήτηση της σωστής απάντησης σε ερώτηση που πραγματοποιήθηκε από το χρήστη σε ανθρώπινη γλώσσα.
- Αυτόματη μορφολογική τεμαχιοποίηση: Η κατάτμηση των λέξεων στα μορφήματά τους καθώς και η αναγνώριση και η κατηγοριοποίηση των μορφημάτων αυτών. Τα μορφήματα ορίζονται ως η μονάδα ελάχιστης σημασίας μιας γλώσσας που αν διαιρεθεί περαιτέρω δε θα έχει πλέον κανένα νόημα. Κάθε ανθρώπινη γλώσσα έχει τα δικά της μορφήματα.
- Αυτόματη περίληψη: Η παραγωγή ενός κειμένου σε αναγνώσιμη μορφή το οποίο περιέχει το νόημα ενός μεγαλύτερου κειμένου συμπυκνωμένο.
- Εξαγωγή πληροφοριών: Η απόκτηση γνώσης από δομημένα ή μη δομημένα δεδομένα μέσω της ανάλυσης των πληροφοριών που παρέχουν.
- Επίλυση σχέσεων συναναφοράς: Το ζήτημα των αναφορών αποτελεί πολλές φορές αιτία για ασάφειες και παρερμηνείες σε μια πρόταση. Η έρευνα στο αντικείμενο αυτό επιχειρεί να προσδιορίσει ποιες λέξεις αναφέρονται σε ποια υποκείμενα σε μια πρόταση ή σε ένα κείμενο. Επομένως, κύριο μέλημα είναι η αντιστοίχιση των αντωνυμιών με τα ουσιαστικά ή τα ονόματα στα οποία αναφέρονται.

- Επισήμανση των μερών του λόγου: Η αυτόματη αναγνώριση και κατηγοριοποίηση των μερών του λόγου των περιεχόμενων σε μια δεδομένη πρόταση τόσο σε γραμματικό όσο και σε συντακτικό επίπεδο.
- Κατανόηση φυσικής γλώσσας: Η μετατροπή των κειμένων και γενικότερα του γραπτού λόγου σε λογικές ακολουθίες οι οποίες έχουν νόημα για τον υπολογιστή.
- Μηχανική μετάφραση: Η αυτόματη μετάφραση κειμένων από μια γλώσσα σε άλλη. Η πρόκληση στο κομμάτι αυτό έγκειται στη συντακτική ορθότητα και στην ικανότητα της μηχανής να αναγνωρίζει και να παραβλέπει ασάφειες του κειμένου.
- Οπτική αναγνώριση χαρακτήρων: Η αυτόματη αναγνώριση χαρακτήρων από κείμενο το οποίο παρουσιάζεται τυπωμένο πάνω σε κάποια εικόνα .
- Παραγωγή φυσικής γλώσσας: Η μετατροπή πληροφοριών από μη δομημένα δεδομένα σε αναγνώσιμο ανθρώπινο λόγο.
- Σύνθεση ομιλίας: Η αυτόματη παραγωγή ανθρώπινου λόγου από τους υπολογιστές.
- Συντακτική ανάλυση: Ο αυτόματος καθορισμός συντακτικού δέντρου μιας πρότασης και η επίλυση ασαφειών.

Περισσότερες πληροφορίες παρέχονται στο άρθρο (“Επεξεργασία φυσικής γλώσσας”)

4. ΕΞΟΡΥΞΗ ΚΕΙΜΕΝΟΥ ΜΕ ΤΗ ΧΡΗΣΗ ΤΗΣ PYTHON

4.1 Εισαγωγή στην Εξόρυξη Κειμένου (Text mining)

Με τον όρο **εξόρυξη κειμένου (Text mining)** εννοείται η παραγωγή χρήσιμης πληροφορίας από μεγάλο όγκο δεδομένων σε μορφή κειμένου. Περιλαμβάνει την απόκτηση πληροφορίας από τον υπολογιστή μέσω της αυτόματης εξόρυξης πληροφορίας από διαφορετικές πηγές όπως ιστοσελίδες, βιβλία, ηλεκτρονική αλληλογραφία, κριτικές και άρθρα. Βασικό εργαλείο για την εξόρυξη πληροφορίας από κείμενο είναι η εκμάθηση στατιστικών προτύπων. Σύμφωνα με τον Andreas Hotho, επιστήμονα στον τομέα της πληροφορικής και καθηγητή επιστήμης δεδομένων στο Πανεπιστήμιο του Wurzburg, διακρίνονται τρεις διαφορετικές προοπτικές της εξόρυξης κειμένου: η εξαγωγή πληροφοριών (information extraction), η εξόρυξη δεδομένων (data mining) και η KDD διαδικασία (Knowledge Discovery in Databases). Η εξόρυξη κειμένου συνήθως περιλαμβάνει την ανάλυση της δομής ενός κειμένου με την προσθήκη γλωσσικών χαρακτηριστικών ή την αφαίρεση άλλων. Μέσω της ανάλυσης της δομής αυτής ανακαλύπτονται μοτίβα που υπάρχουν στα δομημένα δεδομένα και τελικά αξιολογείται και ερμηνεύεται το αποτέλεσμα. Η ποιότητα της πληροφορίας που αντλήθηκε εξαρτάται από παράγοντες όπως η συνάφεια, η καινοτομία και το ενδιαφέρον της πληροφορίας αυτής. Εργασίες που περιλαμβάνουν τη διαδικασία εξόρυξης κειμένου είναι η **κατηγοριοποίηση κειμένου (text classification), η ομαδοποίηση κειμένου (text clustering),** η εξαγωγή νοήματος (concept extraction), η ανάλυση συναισθήματος, η περίληψη κειμένου και η μοντελοποίηση συσχετίσεων οντοτήτων (ER modeling). Βασικός στόχος της εξόρυξης κειμένου είναι η μετατροπή κειμένων σε δεδομένα προς ανάλυση με τη διαδικασία επεξεργασίας φυσικής γλώσσας (NLP) η οποία περιλαμβάνει διαφορετικούς αλγορίθμους και αναλυτικές μεθόδους.

Το κυριότερο συστατικό της διαδικασίας αυτής είναι το έγγραφο. Ως έγγραφο εννοείται ένα σύνολο κειμένων τα οποία υπάρχουν σε διάφορες συλλογές.

Η εξόρυξη κειμένων χρησιμοποιείται ευρέως σε οργανισμούς που βασίζονται στη γνώση και επίσης βοηθά στην απάντηση πολλών ερευνητικών ερωτημάτων. Ανακαλύπτει συσχετίσεις και γεγονότα που βρίσκονται <<κρυμμένα>> σε όγκο μεγάλων κειμένων και χωρίς αυτήν οι πολύτιμες αυτές πληροφορίες θα παρέμεναν ανεκμετάλλευτες.

Τα δομημένα δεδομένα, που προκύπτουν από την εξόρυξη κειμένου, ενσωματώνονται σε βάσεις δεδομένων, αποθήκες δεδομένων ή πίνακες εργαλείων επιχειρηματικής ευφυΐας ώστε να χρησιμοποιηθούν για περιγραφικές, προδιαγραφικές και προγνωστικές αναλύσεις.

Σε συνδυασμό με τη χρήση εργαλείων οπτικοποίησης των δεδομένων, η τεχνική αυτή δίνει τη δυνατότητα στις εταιρείες να ερμηνεύουν τη φύση των αποτελεσμάτων και να λαμβάνουν συνεχώς βελτιωμένες αποφάσεις.

4.2 Ανάλυση Κειμένου (Text Analytics) και Επεξεργασία Φυσικής Γλώσσας (NLP)

Η γραπτή επικοινωνία είναι η πιο διαδεδομένη μορφή ανθρώπινης επικοινωνίας στο σύγχρονο κόσμο του διαδικτύου και των social media. Οι χρήστες συνομιλούν μέσω chat, γραπτών μηνυμάτων, twitter, ηλεκτρονικής αλληλογραφίας, κοινοποιούν καταστάσεις ή ανταλλάσσουν απόψεις σε διάφορα forum σε καθημερινή βάση. Οι δραστηριότητες αυτές δημιουργούν μεγάλο όγκο πληροφοριών σε μορφή κειμένου, οι οποίες τις περισσότερες φορές βρίσκονται σε αδόμητη μορφή. Λόγω της συνεχούς εξέλιξης του ηλεκτρονικού εμπορίου και των social media, καθίσταται απαραίτητη η ανάλυση αυτών των πληροφοριών ώστε να κατανοείται η γνώμη των χρηστών.

Η επεξεργασία φυσικής γλώσσας (NLP) βοηθάει τον υπολογιστή να επικοινωνεί με τους ανθρώπους με φυσικό τρόπο.

4.3 Ανάλυση Κειμένου (Text Analytics) με χρήση της Python

Η Python είναι μια από τις πιο δημοφιλείς γλώσσες προγραμματισμού στον κόσμο εξαιτίας της ικανότητάς της να ενσωματώνεται με άλλες γλώσσες προγραμματισμού και ταιριάζει απόλυτα σε καινοτόμες ιδέες για νέα projects.

Η χρήση της Python είναι ευρέως διαδεδομένη και αναγνωρισμένη στο χώρο της Τεχνητής Νοημοσύνης, πράγμα το οποίο την καθιστά την πλέον κατάλληλη για projects σε τομείς όπως το Soft Computing, το Machine Learning, το NLP και πολλούς άλλους.

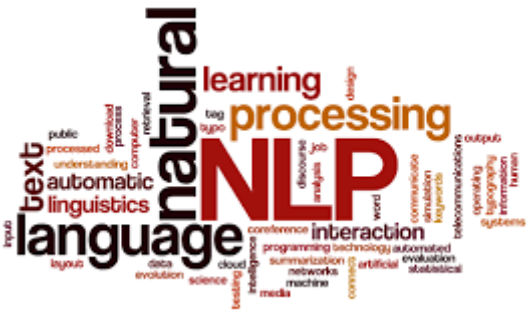
Σχ.4.1 Η γλώσσα προγραμματισμού Python

Πηγή:
https://medium.com/swlh/5-free-python-courses-for-beginners-to-learn-online-e1ca90687caf

Όπως αναφέρθηκε και σε προηγούμενα κεφάλαια, η ανάλυση κειμένου είναι η διαδικασία εξαγωγής νοήματος από κείμενο. Στη σύγχρονη εποχή του διαδικτύου και των διαδικτυακών υπηρεσιών, τα δεδομένα παράγονται με ιλιγγιώδη ταχύτητα και σε μεγάλη ποσότητα. Γενικά, ο επιστήμονας αναλυτής δεδομένων, ο μηχανικός και οι ερευνητές ασχολούνται με κειμενικά ή πινακοειδή δεδομένα. Όπως αναφέρθηκε, οι διαδικτυακές δραστηριότητες όπως τα άρθρα, τα κείμενα σε ιστοσελίδες, οι δημοσιεύσεις σε διάφορα blogs, οι δημοσιεύσεις στα μέσα κοινωνικής δικτύωσης δημιουργούν μη δομημένα δεδομένα σε μορφή κειμένου. Οι επιχειρήσεις καλούνται να αναλύουν τα δεδομένα αυτά ώστε να κατανοήσουν τις δραστηριότητες, τις απόψεις και το feedback των πελατών προς όφελός τους. Για παράδειγμα, μια τέτοια ανάλυση μπορεί να διεξάγεται σε κείμενα πελατών στο πλαίσιο μιας έρευνας αγοράς με σκοπό να παρατηρηθούν οι προτιμήσεις των πελατών και το συναίσθημά τους, δηλαδή αν κρίνουν θετικά ή αρνητικά τις υπηρεσίες. Η ιδέα να έχει ο κάθε οργανισμός το απαραίτητο feedback από τους πελάτες έχει καθοριστική σημασία για τη διαμόρφωση της στρατηγικής του, ώστε να βελτιώνει συνεχώς την εμπειρία των πελατών.

Η ανάλυση κειμένου βρίσκει πολλές εφαρμογές στο σύγχρονο διαδικτυακό κόσμο. Για παράδειγμα, η ανάλυση των δημοσιεύσεων των χρηστών στο Twitter παρέχει πληροφορίες σχετικά με τις απόψεις της κοινής γνώμης για κάποιο γεγονός. Η Amazon, με την ανάλυση των κριτικών από τους πελάτες, αποκτά μια εικόνα για την αρέσκεια ή δυσαρέσκειά τους για κάποιο προϊόν. Παρόμοια πλεονεκτήματα παρέχει και η ανάλυση των σχολίων των χρηστών σε εφαρμογές όπως το YouTube, η Imdb ή το BookMyShow.

Για την καλύτερη απόδοση της διαδικασίας αυτής, οι εταιρείες πλέον χρησιμοποιούν λογισμικά ανάλυσης κειμένου επιστρατεύοντας τη μηχανική μάθηση και αλγορίθμους επεξεργασίας φυσικής γλώσσας (NLP) ώστε να εντοπιστεί το περιεχόμενο σε μεγάλη ποσότητα κειμένων. Το περιβάλλον της Python παρέχει πολλές και διαφορετικές βιβλιοθήκες (libraries) για επεξεργασία φυσικής γλώσσας (NLP) οι οποίες περιέχουν πληθώρα συναρτήσεων που αυτοματοποιούν τις διαδικασίες επεξεργασίας και συνεπώς βοηθούν σε μεγάλο βαθμό τον ερευνητή.



Σχ.4.2 Επεξεργασία φυσικής γλώσσας (NLP)

Πηγή: https://www.blumeglobal.com/learning/natural-language-processing/

4.4 Βιβλιοθήκες της Python για NLP

Η επεξεργασία φυσικής γλώσσας είναι ένα από τα πιο κρίσιμα προβλήματα της Τεχνητής Νοημοσύνης. Στο παρελθόν, μόνο ειδικοί με άριστη γνώση μαθηματικών, στατιστικής, γλωσσολογίας και μηχανικής μάθησης μπορούσαν να συμβάλλουν σε projects επεξεργασίας φυσικής γλώσσας. Πλέον, οι developers έχουν τη δυνατότητα να χρησιμοποιούν έτοιμα εργαλεία, τα οποία απλοποιούν σε μεγάλο βαθμό τη διαδικασία, ώστε να εστιάζουν μόνο στην ανάπτυξη κατάλληλων μοντέλων μηχανικής μάθησης.

Υπάρχει μεγάλος αριθμός εργαλείων και βιβλιοθηκών τα οποία δημιουργήθηκαν αποκλειστικά για χρήση σε NLP. Ορισμένα εργαλεία παρατίθενται στη συνέχεια.

4.4.1 Natural Language Toolkit (NLTK)

Το **Natural Language Toolkit (NLTK)** είναι μία σειρά βιβλιοθηκών και προγραμμάτων της Python για επεξεργασία φυσικής γλώσσας στα Αγγλικά. Το NLTK αναπτύχθηκε στο Τμήμα Επιστήμης Υπολογιστών και Πληροφοριών του Πανεπιστημίου της Πενσυλβάνιας και προορίζεται να υποστηρίξει την έρευνα στην επεξεργασία φυσικής γλώσσας και άλλων κοντινών περιοχών μελέτης όπως την Εμπειρική Γλωσσολογία, την Τεχνητή Νοημοσύνη, την Επιστήμη Γνώσης, την Ανάκτηση Πληροφορίας και τη Μηχανική Μάθηση. Υποστηρίζει διαδικασίες όπως η κατηγοριοποίηση (classification), η εύρεση της ρίζας των όρων (stemming), η ετικετοποίηση (tagging), η ανάλυση (parsing), ο χωρισμός του κειμένου σε οντότητες (tokenization) και άλλες. Περιλαμβάνει μεγάλη ποικιλία αλγορίθμων για επεξεργασία φυσικής γλώσσας. Τέτοιοι αλγόριθμοι είναι το tokenization, το stemming, η ανάλυση συναισθήματος, ο διαχωρισμός θεμάτων και η αναγνώριση οντοτήτων, τα οποία θα αναλυθούν στη συνέχεια. Το NLTK χρησιμοποιείται ευρέως για την προεπεξεργασία κειμένων μέσω των αλγορίθμων του. Επομένως, βοηθάει τον υπολογιστή να αναλύσει, να προεπεξεργαστεί, να προετοιμάσει και να καταλάβει το γραπτό κείμενο.

Σχ.4.3 Natural Language ToolKit in Python

Πηγή :
https://medium.com/towards-artificial-intelligence/text-mining-in-python-steps-and-examples-78b3f8fd913b

4.4.1.1 Προεπεξεργασία δεδομένων κειμένου με το NLTK

Η προεπεξεργασία των δεδομένων είναι το σύνολο των ενεργειών που πραγματοποιούν την προετοιμασία και τον καθαρισμό των δεδομένων και οι οποίες εκτελούνται πριν τη διαδικασία εξόρυξης της γνώσης. Η προεπεξεργασία των δεδομένων είναι απολύτως απαραίτητη καθώς τα δεδομένα που πρόκειται να χρησιμοποιηθούν πάσχουν από διαφόρων ειδών προβλήματα. Τέτοια προβλήματα

είναι η ύπαρξη αλληλοσυγκρουόμενων πληροφοριών, οι επαναλήψεις που καθιστούν τον όγκο των δεδομένων μη διαχειρίσιμο, η ύπαρξη ασυνεπειών ως προς την κωδικοποίηση, οι χαμένες τιμές και ο θόρυβος. Πολλά προβλεπτικά μοντέλα ανά τα χρόνια έχουν οδηγήσει σε λανθασμένα ή ανακριβή αποτελέσματα λόγω της μη ύπαρξης ή της κακής προεπεξεργασίας των δεδομένων. Με τη διαδικασία της προεπεξεργασίας, τα δεδομένα αποκτούν μια μορφή ικανή να αναλυθεί και να χρησιμοποιηθεί για προβλέψεις. Η προεπεξεργασία δεδομένων περιλαμβάνει ένα σύνολο από ενέργειες οι οποίες θα παρατεθούν αναλυτικότερα στη συνέχεια, ωστόσο, αξίζει να σημειωθεί ότι δεν υπάρχει συγκεκριμένη μεθοδολογία και σειρά διεξαγωγής των ενεργειών, καθώς κάθε πρόβλημα πιθανότατα απαιτεί διαφορετική προεπεξεργασία των υπαρχόντων δεδομένων.

Όπως αναφέρθηκε προηγουμένως, η βιβλιοθήκη NLTK παρέχει στους προγραμματιστές πολλές δυνατότητες προκειμένου να διαχειριστούν τα δεδομένα που διατίθενται σε μορφή κειμένου ώστε να εξαχθεί και να απομονωθεί η χρήσιμη πληροφορία. Παρακάτω παρατίθενται ορισμένες ενέργειες που μπορούν να διεξαχθούν μέσω του NLTK.

Tokenization

Ένα κείμενο ή ακόμα και μια πρόταση μπορεί να χωριστεί σε οντότητες με τη διαδικασία του tokenization. Το tokenization είναι συνήθως το πρώτο στάδιο της ανάλυσης κειμένου κατά την οποία μια παράγραφος διαιρείται σε μικρότερα μέρη. Το δείγμα (token) είναι μια μικρή, ξεχωριστή και αυτόνομη οντότητα η οποία αποτελεί το συστατικό στοιχείο των παραγράφων. Οντότητες, για παράδειγμα, μιας παραγράφου μπορεί να είναι οι λέξεις οι οποίες τη συνθέτουν. Η διαδικασία αυτή στο περιβάλλον της Python πραγματοποιείται ως εξής:

```
In [1]: import nltk
        text = "King Lear of Britain, elderly and wanting to retire from the duties of the monarchy, decides to divide his realm among hi

In [5]: tokens = nltk.word_tokenize(text)
        print(tokens)

['King', 'Lear', 'of', 'Britain', ',', 'elderly', 'and', 'wanting', 'to', 'retire', 'from', 'the', 'duties', 'of', 'the', 'mona
rchy', ',', 'decides', 'to', 'divide', 'his', 'realm', 'among', 'his', 'three', 'daughters']
```

Επίσης, η διαδικασία tokenization μπορεί να εφαρμοστεί χωρίζοντας μια παράγραφο σε επιμέρους προτάσεις (sentence tokenization) όπως φαίνεται παρακάτω.

In [8]:
```
from nltk.tokenize import sent_tokenize
paragraph = "King Lear of Britain, elderly and wanting to retire from the duties of the monarchy, \
decides to divide his realm among his three daughters, and declares he \
will offer the largest share to the one who loves him most. The eldest, Goneril, \
speaks first, declaring her love for her father in fulsome terms. Moved by her flattery \
Lear proceeds to grant to Goneril her share as soon as she has finished her declaration, \
before Regan and Cordelia have a chance to speak. He then awards to Regan her share as soon \
as she has spoken. When it is finally the turn of his youngest and favourite daughter, Cordelia, \
at first she refuses to say anything (Nothing, my Lord) and then declares there is nothing to compare her love to, \
no words to properly express it; she says honestly but bluntly that she loves him according to her bond, \
no more and no less, and will reserve half of her love for her future husband. Infuriated, Lear disinherits \
Cordelia and divides her share between her elder sisters."

tokenized_text = sent_tokenize(paragraph)
print(tokenized_text)
```

```
['King Lear of Britain, elderly and wanting to retire from the duties of the monarchy, decides to divide his realm among his th
ree daughters, and declares he will offer the largest share to the one who loves him most.', 'The eldest, Goneril, speaks firs
t, declaring her love for her father in fulsome terms.', 'Moved by her flattery Lear proceeds to grant to Goneril her share as
soon as she has finished her declaration, before Regan and Cordelia have a chance to speak.', 'He then awards to Regan her shar
e as soon as she has spoken.', 'When it is finally the turn of his youngest and favourite daughter, Cordelia, at first she refu
ses to say anything (Nothing, my Lord) and then declares there is nothing to compare her love to, no words to properly express
it; she says honestly but bluntly that she loves him according to her bond, no more and no less, and will reserve half of her l
ove for her future husband.', 'Infuriated, Lear disinherits Cordelia and divides her share between her elder sisters.']
```

Μέσω του **Tokenization** μπορούν να μετρηθούν οι συχνότητες εμφάνισης των όρων στα δείγματα με τη συνάρτηση **FreqDist** όπως φαίνεται παρακάτω:

In [9]:
```
from nltk.tokenize import word_tokenize
from nltk.probability import FreqDist

text = "King Lear of Britain, elderly and wanting to retire from the duties of the monarchy, \
decides to divide his realm among his three daughters"

tokens = nltk.word_tokenize(text)
print(tokens)

fdist = FreqDist(tokens)
print(fdist)
```

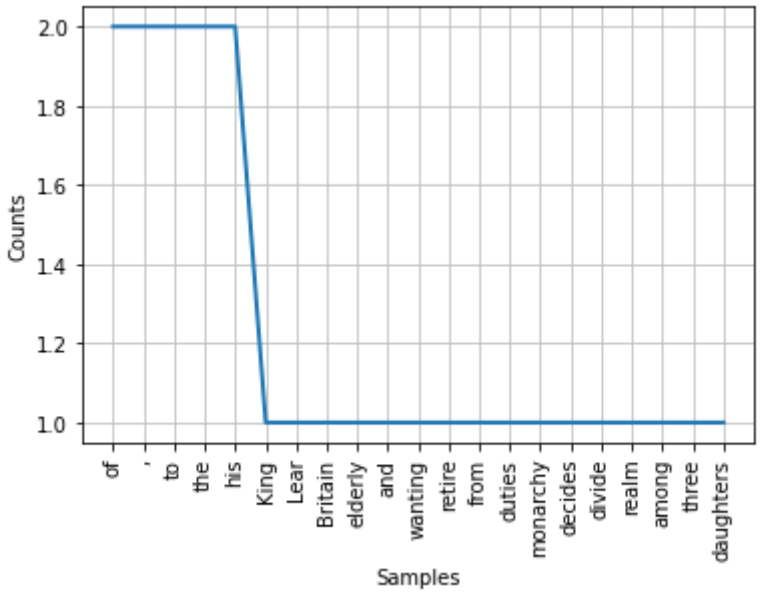
```
['King', 'Lear', 'of', 'Britain', ',', 'elderly', 'and', 'wanting', 'to', 'retire', 'from', 'the', 'duties', 'of', 'the', 'mona
rchy', ',', 'decides', 'to', 'divide', 'his', 'realm', 'among', 'his', 'three', 'daughters']
<FreqDist with 21 samples and 26 outcomes>
```

Από την κατανομή συχνοτήτων υπάρχει η δυνατότητα εύρεσης των περισσότερο εμφανιζόμενων όρων καθώς και η γραφική παράσταση της ίδιας της κατανομής συχνοτήτων.

In [10]:
```
fdist.most_common(3)
```

Out[10]:
```
[('of', 2), (',', 2), ('to', 2)]
```

In [12]:
```
#Frequency Distribution Plot

import matplotlib.pyplot as plt
fdist.plot(26, cumulative = False)
plt.show()
```



Σχ.4.4 Σχηματική παρουσίαση συχνότητας εμφάνισης όρων

Stopwords Removal

Η παρουσία ορισμένων λέξεων όπως is, a, an, to, of, and, the, κλπ, οι οποίες αποκαλούνται *stopwords*, στα δεδομένα δεν προσδίδει καμία χρήσιμη πληροφορία, αντιθέτως, εισάγει σημαντικό θόρυβο στα δεδομένα και αυξάνει κατά πολύ τον όγκο τους. Για το λόγο αυτό, κατά την προεπεξεργασία των δεδομένων ο αναλυτής δεν παραλείπει να "καθαρίσει" το κείμενο από τέτοιους όρους. Η βιβλιοθήκη NLTK παρέχει μεταξύ άλλων και αυτή τη δυνατότητα μέσω της δημιουργίας μιας λίστας από stopwords. Παρακάτω βλέπουμε τη λίστα που παρέχει το σώμα του NLTK για την αγγλική γλώσσα.

```
In [13]: from nltk.corpus import stopwords
         stop_words=set(stopwords.words("english"))
         print(stop_words)
```

```
{'no', 'did', 'm', 'but', 'ma', 'between', 'hers', 'his', 'were', 'an', 'above', 'too', 'wouldn', 'not', 'so', 't', 'needn', 'o
ut', 'whom', "weren't", 'to', 'won', "it's", 'by', 'it', 'from', 'again', 'each', 'their', 'have', 'same', "hasn't", 'being',
"shouldn't", 'isn', 'there', 'that', 'what', "couldn't", 'i', 'all', 'yourself', 'couldn', 'themselves', "that'll", 'further',
'don', 'if', 'mustn', 'or', 'your', 'down', 'be', 'yourselves', 'doing', 'having', 'should', 'myself', 'where', 'both', 'here',
'will', 'which', 'who', 'is', "don't", 'under', 'are', 'very', 'our', 'had', 'just', 'hasn', 'through', 'my', 'herself', "did
n't", 'during', 'them', "aren't", 'off', 'about', "wasn't", 'theirs', 'once', 'ours', 'of', 'him', 'the', 'for', 'he', 'this',
'against', "mustn't", "needn't", 'didn', "doesn't", 'mightn', 'll', 'shan', 'then', 'me', 'these', 'am', 'at', 'below', 'such',
'more', "should've", 'ourselves', 'up', 'as', "haven't", 'weren', 'over', 'doesn', 'hadn', 'they', 'why', 'himself', 'was', 'ot
her', 'and', "you'd", 'on', "hadn't", 'while', 'been', 'her', 'aren', 's', 'shouldn', 'you', 'in', 'nor', 'into', "you've", 'on
ly', 'most', 'after', 'a', 'does', "you're", 'do', 'its', 're', 'o', "she's", 'now', "won't", 'own', 'with', 'before', 'itsel
f', 'because', 'how', "wouldn't", 'any', 'few', 'when', "isn't", 'she', 'has', 'yours', 'we', "mightn't", 'until', "shan't", 'a
in', 've', 'd', 'y', "you'll", 'can', 'haven', 'than', 'wasn', 'some', 'those'}
```

Υποθέτοντας ότι τα δεδομένα μας περιλαμβάνουν κείμενα, στο παρακάτω παράδειγμα φαίνεται η εξαγωγή των stopwords από τέτοιου είδους δεδομένα.

```
In [14]: #Stopwords Removal

filtered_sentence = []
for i in tokens:
    if i not in stop_words:
        filtered_sentence.append(i)
print("Tokenized Sentence:", tokens)
print("Tokenized Sentence without stopwords:", filtered_sentence)
```

```
Tokenized Sentence: ['King', 'Lear', 'of', 'Britain', ',', 'elderly', 'and', 'wanting', 'to', 'retire', 'from', 'the', 'dutie
s', 'of', 'the', 'monarchy', ',', 'decides', 'to', 'divide', 'his', 'realm', 'among', 'his', 'three', 'daughters']
Tokenized Sentence without stopwords: ['King', 'Lear', 'Britain', ',', 'elderly', 'wanting', 'retire', 'duties', 'monarchy',
',', 'decides', 'divide', 'realm', 'among', 'three', 'daughters']
```

Είναι εμφανές ότι τα φιλτραρισμένα δεδομένα έχουν μικρότερο όγκο και περιέχουν την ίδια σημαντική πληροφορία με τα αρχικά χωρίς θόρυβο.

Lexicon Normalization

Η κανονικοποίηση λέξεων σε δεδομένα κειμένου επιλύει ένα άλλου είδους πρόβλημα των δεδομένων που εισάγει θόρυβο. Η παρουσία όρων οι οποίοι στα <<μάτια>> του υπολογιστή είναι διαφορετικοί και ανεξάρτητοι μεταξύ τους, ενώ ουσιαστικά είναι ομόρριζοι, προκαλεί δυσαρέσκεια στους αναλυτές. Για παράδειγμα, οι όροι *analysis, analyst, analyse, analysing* μπορούν να συνδεθούν και να περιγραφούν από το λήμμα analyse. Η διαδικασία της λεξιλογικής κανονικοποίησης μειώνει τις παράγωγες μορφές μιας λέξης σε μία κοινή ρίζα.

- **Stemming**

Η διαδικασία του Stemming παρέχεται ως δυνατότητα από τη βιβλιοθήκη NLTK και μειώνει τις ομόρριζες λέξεις του κειμένου συσχετίζοντας την καθεμία με τη ρίζα τους αγνοώντας τις καταλήξεις που μπορεί να έχουν.

Παρακάτω φαίνεται η διαδικασία stemming στο παράδειγμά μας.

```
In [15]: #Stemming

from nltk.stem import PorterStemmer

ps = PorterStemmer()

stemmed_words = []
for i in filtered_sentence:
    stemmed_words.append(ps.stem(i))

print("Filtered Sentence", filtered_sentence)
print("Stemmed Words", stemmed_words)
```

```
Filtered Sentence ['King', 'Lear', 'Britain', ',', 'elderly', 'wanting', 'retire', 'duties', 'monarchy', ',', 'decides', 'divid
e', 'realm', 'among', 'three', 'daughters']
Stemmed Words ['king', 'lear', 'britain', ',', 'elderli', 'want', 'retir', 'duti', 'monarchi', ',', 'decid', 'divid', 'realm',
'among', 'three', 'daughter']
```

- **Lemmatization**

Η διαδικασία Lemmatization διαμορφώνει τις λέξεις ώστε να επιστρέψουν στο αρχικό τους λήμμα. Η διαφορά του Lemmatization με το Stemming είναι ότι αυτή τη φορά ελέγχεται η γλωσσολογική προέλευση των λέξεων βάση του λεξικού ώστε να εξαχθεί το ζητούμενο λήμμα. Το Lemmatization είναι περισσότερο εξελιγμένη διαδικασία από το Stemming. Για παράδειγμα, το λήμμα της λέξης "*worse*" είναι το "*bad*". Το Stemming δε μπορεί να πραγματοποιήσει αυτή τη σύνδεση καθώς ασχολείται μόνο με τη λέξη χωρίς να αντλεί τις γλωσσολογικές της πληροφορίες. Στο παρακάτω παράδειγμα φαίνεται η εφαρμογή τόσο του lemmatization όσο και του stemming στους όρους "*studying*" και "*better*".

```
In [2]: #Stemming and Lemmatization

        from nltk.stem.wordnet import WordNetLemmatizer
        lem = WordNetLemmatizer()

        from nltk.stem.porter import PorterStemmer
        stem = PorterStemmer()

        print("Lemmatized Word :", lem.lemmatize("studying", pos='v'))
        print("Stemmed Word :", stem.stem("studying"))

        print("Lemmatized Word :", lem.lemmatize("better", pos='a'))
        print("Stemmed Word :", stem.stem("better"))
```

```
Lemmatized Word : study
Stemmed Word : studi
Lemmatized Word : good
Stemmed Word : better
```

- **POS Tagging (Επισήμανση μερών του λόγου)**

Σκοπός της διαδικασίας αυτής είναι η κατηγοριοποίηση ενός δοθέντος όρου σε ένα γραμματικό σύνολο ανάλογα με το περιεχόμενο του κειμένου. Η έξοδος του POS Tagging είναι η κατηγορία στην οποία ανήκει ο όρος, αν είναι δηλαδή ουσιαστικό, αντωνυμία, επίθετο, ρήμα, επίρρημα κλπ. Παρακάτω φαίνεται η διαδικασία αυτή σε ένα σύνολο λέξεων.

```
In [4]: #POS TAGGING

        import nltk
        text = "King Lear of Britain, elderly and wanting to retire from the duties of the monarchy, \
        decides to divide his realm among his three daughters"

        tokens = nltk.word_tokenize(text)
        nltk.pos_tag(tokens)
```

```
Out[4]: [('King', 'VBG'),
         ('Lear', 'NNP'),
         ('of', 'IN'),
         ('Britain', 'NNP'),
         (',', ','),
         ('elderly', 'RB'),
         ('and', 'CC'),
         ('wanting', 'VBG'),
         ('to', 'TO'),
         ('retire', 'VB'),
         ('from', 'IN'),
         ('the', 'DT'),
         ('duties', 'NNS'),
         ('of', 'IN'),
         ('the', 'DT'),
         ('monarchy', 'NN'),
         (',', ','),
         ('decides', 'VBZ'),
         ('to', 'TO'),
         ('divide', 'VB'),
         ('his', 'PRP$'),
         ('realm', 'NN'),
         ('among', 'IN'),
         ('his', 'PRP$'),
         ('three', 'CD'),
         ('daughters', 'NNS')]
```

- **Lowercasing**

Ίσως το απλούστερο βήμα του της προεπεξεργασίας δεδομένων (text preprocessing) είναι το lowercasing, δηλαδή η μετατροπή όλων των κεφαλαίων γραμμάτων ενός κειμένου σε πεζά καθώς ο υπολογιστής δε μπορεί να αντιληφθεί πως δύο γράμματα είναι τα ίδια αν το ένα είναι κεφαλαίο και το άλλο πεζό. Επομένως, ο αναλυτής προτού ξεκινήσει την επεξεργασία των δεδομένων, δεν παραλείπει να μετατρέψει όλα τα γράμματα του κειμένου σε πεζά. Παρακάτω φαίνεται ένα παράδειγμα μετατροπής των γραμμάτων των όρων σε πεζά σε μία πρόταση.

```
In [5]: text = "King Lear of Britain, elderly and wanting to retire from the duties of the monarchy, \
        decides to divide his realm among his three daughters"

        print(text.lower())
```

```
king lear of britain, elderly and wanting to retire from the duties of the monarchy, decides to divide his realm among his thre
e daughters
```

4.4.2 Spacy

Η Spacy είναι μια σχετικά νέα, ανοιχτού κώδικα βιβλιοθήκη λογισμικού που σχεδιάστηκε για επεξεργασία φυσικής γλώσσας. Είναι γραμμένη στις γλώσσες Python και Cython, δημοσιεύθηκε με άδεια του MIT και δημιουργήθηκε από τους προγραμματιστές Matthew Honnibal και Ines Montani. Είναι περισσότερο προσβάσιμη από τις υπόλοιπες βιβλιοθήκες της Python. Σε αντίθεση με το NLTK που έχει περισσότερο εκπαιδευτική και ερευνητική χρήση, η Spacy επικεντρώνει το λογισμικό της σε εργασίες που αφορούν την παραγωγή. Από την έκδοση 1.0, η Spacy υποστηρίζει ροές εργασίας βαθιάς μάθησης οι οποίες επιτρέπουν τη χρήση στατιστικών μοντέλων από άλλες δημοφιλείς βιβλιοθήκες όπως η Tensorflow, η Pytorch, η MXnet μέσω της δικής της βιβλιοθήκης Thinc. Με τη χρήση του Thinc ως backend, η Spacy διαθέτει μοντέλα νευρωνικών δικτύων που πραγματοποιούν κατηγοριοποίηση κειμένου, ανάλυση εξαρτήσεων, αναγνώριση οντότητας (NER) και επισήμανση μερικής ομιλίας. Η Spacy παρέχει έναν εξαιρετικά γρήγορο συντακτικό αναλυτή και, όπως έχει αποδειχθεί από τη χρήση της, είναι αρκετά αποτελεσματική. Το μειονέκτημα της Spacy είναι το γεγονός ότι υποστηρίζει μικρό αριθμό γλωσσών σε σχέση με τις άλλες βιβλιοθήκες. Ωστόσο, με την ανερχόμενη τάση της μηχανικής μάθησης και της επεξεργασίας φυσικής γλώσσας, υπάρχει ενδεχομένως η προοπτική να υποστηρίζει περισσότερες γλώσσες στο μέλλον. Η Spacy υποστηρίζει ένα σύνολο από διαδικασίες όπως:

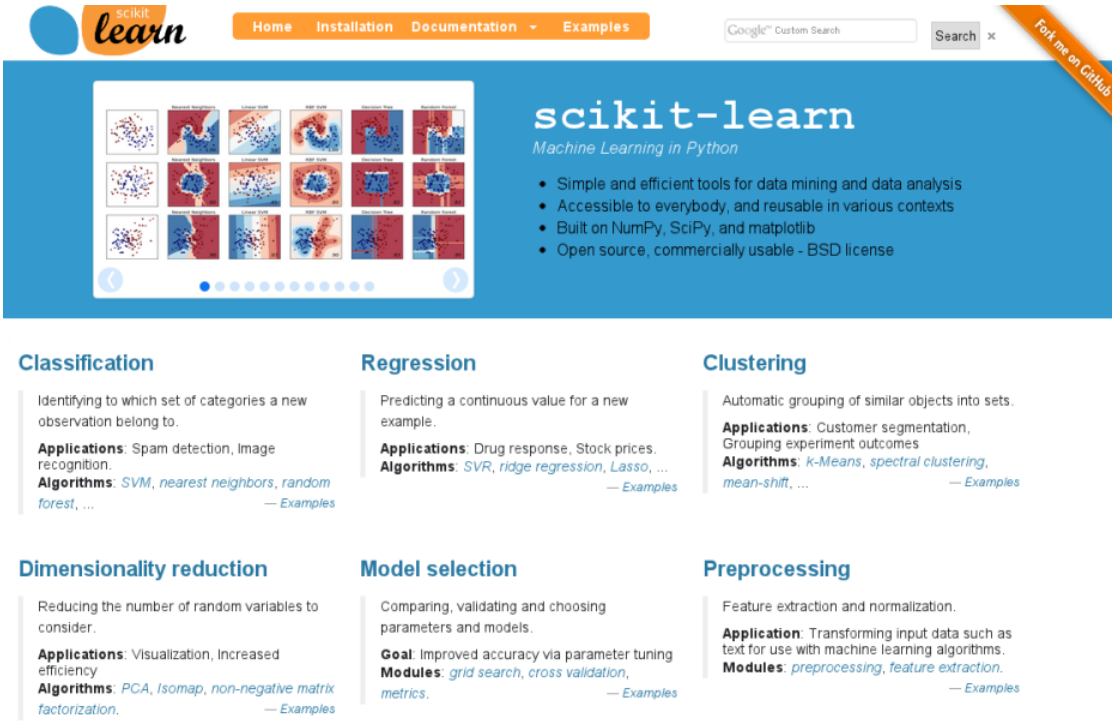
- Non - destructive tokenization
- Named Entity Recognition
- “Alpha Tokenization” για περισσότερες από 50 γλώσσες
- Στατιστικά μοντέλα για 11 γλώσσες
- Προεκπαιδευμένα διανύσματα λέξεων
- Part-of-speech tagging
- Labelled dependency parsing
- Κατηγοριοποίηση κειμένου
- Ενσωματωμένους οπτικοποιητές για συντακτικές και ονοματολογικές οντότητες
- Ενσωμάτωση βαθιάς μάθησης (deep learning integration)

Περισσότερες πληροφορίες παρέχονται στο άρθρο (“Spacy”)

4.4.3 Scikit - Learn

Η συγκεκριμένη βιβλιοθήκη είναι εξαιρετικά εύχρηστη και παρέχει στους προγραμματιστές μια μεγάλη γκάμα αλγορίθμων επιβλεπόμενης και μη επιβλεπόμενης μάθησης στο σταθερό interface της Python για τη δημιουργία στατιστικών μοντέλων. Προσφέρει επίσης πολλές λειτουργίες για τη χρήση της

μεθόδου bag-of-words για τη δημιουργία χαρακτηριστικών τα οποία βοηθούν στην επίλυση προβλημάτων ταξινόμησης. Το πλεονέκτημα της είναι οι διαισθητικές μέθοδοι κλάσεων. Η βιβλιοθήκη αυτή περιλαμβάνει ένα άρτια δομημένο documentation το οποίο βοηθά σε μεγάλο βαθμό τους χρήστες να αξιοποιήσουν στο έπακρο τις δυνατότητές του. Ωστόσο, η scikit - learn δεν υποστηρίζει νευρωνικά δίκτυα για προεπεξεργασία κειμένου. Διαθέτει διάφορους αλγορίθμους ταξινόμησης, παλινδρόμησης και ομαδοποίησης συμπεριλαμβανομένων των support vector models (SVM), random forests, k-means, gradient boosting και DBSCAN. Είναι σχεδιασμένη να λειτουργεί με την ταυτόχρονη χρήση των βιβλιοθηκών NumPy, SciPy. Αρχικά ξεκίνησε ως Scikits.learn το 2007, ένα έργο του David Cournapeau. Αργότερα, ο Matthieu Brucher, ποσοτικός αναλυτής, εντάχθηκε στο πρόγραμμα χρησιμοποιώντας το ως εργαλείο για τη διατριβή του. Το 2010 συμμετείχε και η INRIA (Εθνικό Ινστιτούτο Έρευνας στην Πληροφορική και στον Αυτοματισμό) και η πρώτη δημόσια κυκλοφορία (v0.1 beta) δημοσιεύτηκε στα τέλη Ιανουαρίου 2010. Το έργο του σχεδιασμού της βιβλιοθήκης αυτής έχει περισσότερους από 30 ενεργούς συνεργάτες και έχει λάβει υποτροφίες από εταιρείες όπως η INRIA, η Google, η Tinyclues, και η Python Software Foundation.



Σχ.4.5 Scikit - Learn Homepage

Πηγή :
https://machinelearningmastery.com/a-gentle-introduction-to-scikit-learn-a-python-machine-learning-library/

Διατίθεται με επιτρεπόμενη απλουστευμένη BSD άδεια και διανέμεται σε Linux ώστε να είναι δυνατή η εμπορική και η ακαδημαϊκή χρήση της. Η βιβλιοθήκη βασίζεται στο SciPy (Scientific Python) το οποίο περιλαμβάνει μια σειρά εργαλείων τα οποία πρέπει να εγκατασταθούν πριν χρησιμοποιηθεί το Scikit-Learn. Τα εργαλεία αυτά περιλαμβάνουν τα εξής:

- SciPy : Θεμελιώδης βιβλιοθήκη για scientific computing
- NumPy : Πακέτο πινάκων n-διαστάσεων
- Matplotlib : 2D/3D γραφική παρουσίαση δεδομένων
- IPython : Κονσόλα για διαδραστικό computing
- Sympy : Συμβολικά μαθηματικά
- Pandas : Δομές και αναλύσεις δεδομένων

Η βασική ιδέα της συγκεκριμένης βιβλιοθήκης είναι η διατήρηση και η εξέλιξη ενός επιπέδου ευρωστίας και υποστήριξης για την εκτεταμένη χρήση της στην παραγωγή. Αυτό συνεπάγεται τη συνεχή προσπάθεια για βελτίωση σε ζητήματα ευκολίας στη χρήση, ποιότητας του κώδικα, συνεργασίας, documentation και απόδοσης.

Η Scikit - Learn εστιάζει κυρίως στη μοντελοποίηση των δεδομένων παρά στη φόρτωση, τη διαχείριση ή τη σύνοψή τους. Για τις δυνατότητες αυτές καταλληλότερα εργαλεία είναι το NumPy και το Pandas.

Ορισμένα δημοφιλή μοντέλα που παρέχει η Scikit - Learn είναι τα παρακάτω:

- **Clustering** : Ομαδοποίηση σε μη ετικετοποιημένα δεδομένα (Kmeans)
- **Cross Validation** : Για εκτίμηση της απόδοσης των μοντέλων σε μη ελεγμένα δεδομένα
- **Datasets** : Για σύνολα δεδομένων ελέγχου και για τη δημιουργία συνόλων δεδομένων με συγκεκριμένα χαρακτηριστικά ώστε να εξεταστεί η συμπεριφορά των μοντέλων
- **Dimensionality Reduction** : Μείωση του συνόλου των χαρακτηριστικών με στόχο τη σύνοψη, την οπτικοποίηση και την επιλογή των βασικών χαρακτηριστικών των δεδομένων (feature selection)
- **Ensemble methods** : Για συνδυασμό των προβλέψεων που προέρχονται από διαφορετικά μοντέλα
- **Feature Extraction** : Για τον καθορισμό των σημαντικών χαρακτηριστικών εικόνων ή κειμένων τα οποία θα χρησιμοποιηθούν στις ταξινομήσεις
- **Feature Selection** : Η εύρεση των σημαντικών από τα χαρακτηριστικά των δεδομένων τα οποία θα χρησιμοποιηθούν στα μοντέλα επιβλεπόμενης μάθησης
- **Parameter Tuning** : Για να αξιοποιηθούν στο έπακρο τα εποπτευόμενα μοντέλα
- **Manifold Learning** : Για σύνοψη και απεικόνιση πολύπλοκων πολυδιάστατων δεδομένων

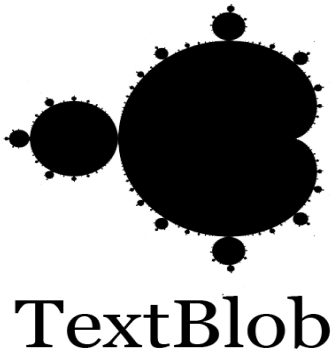
- **Supervised Models** : μια μεγάλη σειρά από γενικευμένα γραμμικά μοντέλα, discriminate analysis, naive bayes, lazy methods, νευρωνικα δίκτυα, support vector machines και δένδρα αποφάσεων.

Περισσότερες πληροφορίες παρέχονται στο άρθρο: ("A Gentle Introduction to Scikit-Learn: A Python Machine Learning Library")

4.4.4 TextBlob

Η βιβλιοθήκη TextBlob είναι ένα απαραίτητο εργαλείο για προγραμματιστές οι οποίοι ξεκινούν το ταξίδι τους στην επεξεργασία φυσικής γλώσσας (NLP) στην Python και επιθυμούν να αξιοποιήσουν σε μέγιστο βαθμό τις δυνατότητες του NLTK. Κυρίως βοηθάει τους αρχάριους να εξοικειωθούν με εργασίες όπως η ανάλυση συναισθήματος, η ετικετοποίηση (pos - tagging), η ταξινόμηση (classification), η μετάφραση και η εξαγωγή φράσεων ουσιαστικών (noun phrase extraction).Η συγκεκριμένη βιβλιοθήκη χρησιμοποιείται εκτενώς και επιτυχώς σε συνδυασμό με το NLTK και το Pattern. Αναλυτικότερα οι δυνατότητές της είναι οι εξής:

- Εξαγωγή φράσεων ουσιαστικών (Noun Phrase Extraction)
- Επισήμανση μερών του λόγου (Part - of - Speech Tagging)
- Ανάλυση συναισθήματος (Sentiment Analysis)
- Ταξινόμηση (Classification)
- Tokenization (διαχωρισμός κειμένου σε λέξεις ή προτάσεις)
- Υπολογισμός συχνοτήτων λέξεων ή φράσεων
- Ανάλυση (Parsing)
- n-grams (Εξαγωγή n οντοτήτων από κείμενο)
- Κλίση λέξεων (Ενικός και πληθυντικός αριθμός) και λημματοποίηση (lemmatization)
- Ορθογραφική διόρθωση (Spelling correction)
- Εισαγωγή νέων μοντέλων ή γλωσσών μέσω των επεκτάσεων
- Ενσωμάτωση WordNet



Σχ. 4.6 TextBlob

Πηγή: https://textblob.readthedocs.io/en/dev/

Περισσότερες πληροφορίες παρέχονται στο άρθρο : ("TextBlob: Simplified Text Processing")

4.4.5 Pattern

Ένα ακόμα εργαλείο της Python που χρησιμοποιούν οι προγραμματιστές για την επεξεργασία φυσικής γλώσσας είναι η βιβλιοθήκη pattern. Η συγκεκριμένη βιβλιοθήκη επιτρέπει εργασίες όπως η επισήμανση μερών του λόγου (Part-of-Speech tagging), η ανάλυση συναισθήματος, η μοντελοποίηση με διανύσματα (vector space modeling), το clustering, η αναζήτηση n οντοτήτων σε κείμενο (n-gram), το tokenization και το stemming. Επίσης, περιέχει APIs ώστε να αντλεί δεδομένα από ιστοτόπους όπως το Twitter, το Facebook, η Wikipedia και άλλους. Τέλος, διαθέτει μοντέλα μηχανικής μάθησης όπως perceptron, KKN, SVM τα οποία χρησιμοποιούνται σε προβλήματα ταξινόμησης, παλινδρόμησης και ομαδοποίησης.

Περισσότερες πληροφορίες παρέχονται στο άρθρο : ("Python for NLP: Introduction to the Pattern Library")

Ονομαστικά αναφέρονται επιπλέον οι βιβλιοθήκες της Python Polyglot, Gensim και CoreNLP οι οποίες ολοκληρώνουν το οπλοστάσιο της Python για σκοπούς επεξεργασίας φυσικής γλώσσας

4.5 Εφαρμογές της Εξόρυξης Κειμένου (Text Mining)

Η τεχνολογία της εξόρυξης κειμένου, ή όμοια της ανάλυσης κειμένου, εξυπηρετεί πληθώρα αναγκών με κυβερνητικό, ερευνητικό ή επιχειρηματικό ρόλο. Χρησιμοποιείται ώστε οι εκάστοτε υπηρεσίες να καταγράφουν τα δεδομένα από τις καθημερινές τους δραστηριότητες. Για παράδειγμα, κυβερνήσεις και στρατός χρησιμοποιούν την εξόρυξη κειμένου για ζητήματα εθνικής ασφάλειας και νοημοσύνης. Οι επιστημονικοί ερευνητές ενσωματώνουν την τεχνολογία της εξόρυξης κειμένου στην προσπάθεια να διαχειριστούν μεγάλους όγκους γραπτών κειμένων, στην κατανόηση των ιδεών από γραπτά κείμενα και στην έρευνα ζητημάτων βιοπληροφορικής. Στις επιχειρήσεις, η εξόρυξη κειμένου εφαρμόζεται στις αυτοματοποιημένες διαφημίσεις, στην ανταγωνιστική νοημοσύνη (competitive intelligence) και σε άλλες δραστηριότητες. Αναλυτικότερα τα πεδία εφαρμογής της εξόρυξης κειμένου παρουσιάζονται ως εξής:

Εφαρμογές Ασφάλειας (Security Applications)

Λογισμικά εξόρυξης κειμένου διατίθενται για ζητήματα ασφαλείας όπως η παρακολούθηση και η ανάλυση γραπτών κειμένων που βρίσκονται σε διάφορες πηγές στο διαδίκτυο όπως ειδησεογραφικές ιστοσελίδες, μέσα κοινωνικής δικτύωσης, blogs κλπ. Πολλές φορές οι παραπάνω τεχνικές εφαρμόζονται για την εθνική ασφάλεια των κρατών.

Βιοϊατρικές Εφαρμογές

Η παρουσία της εξόρυξης κειμένου στον τομέα αυτό υπόσχεται πολλές καινοτομίες στο εγγύς μέλλον. Ανάλυση κειμένου χρησιμοποιείται στις μελέτες για τη διερεύνηση των συνδέσεων και των αλληλεπιδράσεων των πρωτεϊνών και τη συσχέτιση μεταξύ ασθενειών και πρωτεϊνών. Επιπλέον, ο μεγάλος όγκος των κλινικών δεδομένων των ασθενών, τα δημογραφικά δεδομένα των πληθυσμών και η αναφορά ανεπιθύμητων ενεργειών από τους ασθενείς διευκολύνουν τις έρευνες στον τομέα της ιατρικής και της φαρμακολογίας και οδηγούν στην όλο και αυξανόμενη ακρίβεια της ιατρικής. Οι αλγόριθμοι εξόρυξης κειμένου εξυπηρετούν την καταγραφή και την εύρεση συγκεκριμένων κλινικών συμβάντων από μεγάλα σύνολα δεδομένων ασθενών με συμπτώματα, παρενέργειες από ηλεκτρονικά δεδομένα, αναφορές συμβάντων ή και αναφορές από διαγνωστικά τεστ. Μια διαδικτυακή εφαρμογή, η οποία συνδυάζει εξόρυξη κειμένου σε ιατρικά δεδομένα και οπτικοποιήσεις, είναι η PubGene. Η συγκεκριμένη εφαρμογή είναι μια μηχανή αναζήτησης δημοσίως προσβάσιμη.

Εφαρμογές Λογισμικού (Software Applications)

Εταιρείες λογισμικού όπως η IBM και η Microsoft έχουν αναπτύξει και ερευνήσει μεθόδους και λογισμικά εξόρυξης κειμένου ώστε να αυτοματοποιήσουν σε μεγαλύτερο βαθμό τις διαδικασίες εξόρυξης και ανάλυσης των δεδομένων. Ταυτόχρονα άλλες εταιρείες μέσω της εξόρυξης κειμένου ερευνούν και επενδύουν στη βελτίωση των αποτελεσμάτων τους. Επιπλέον, τα τελευταία χρόνια πραγματοποιούνται εντατικές προσπάθειες παρακολούθησης τρομοκρατικών ενεργειών. Σε επίπεδο έρευνας, το Weka Software παίζει σημαντικό ρόλο στον επιστημονικό κόσμο και αποτελεί εργαλείο εισαγωγής ιδιαιτέρως για αρχάριους.

Online Media Εφαρμογές

Μεγάλες εταιρείες όπως η Tribune Company χρησιμοποιούν την εξόρυξη κειμένου για να ταξινομήσουν σημαντικές πληροφορίες σε ιστοσελίδες με αποτέλεσμα να βελτιώσουν την εμπειρία των χρηστών. Στόχος είναι η αύξηση της επισκεψιμότητας των διαφόρων ιστοσελίδων και κατ επέκταση των εσόδων

Επιχειρηματικές και Εμπορικές Εφαρμογές

Η ανάλυση κειμένου χρησιμοποιείται από τις επιχειρήσεις στο πλαίσιο του Marketing προκειμένου να συσχετίζουν τις προτιμήσεις των πελατών αναζητώντας μεθόδους προσέλκυσής τους (customer relationship management). Επιπλέον, οι αλγόριθμοι ανάλυσης κειμένου συνθέτουν προβλεπτικά μοντέλα απώλειας πελατών και περίσσειας ή ελλείμματος αποθεμάτων.

Ανάλυση Συναισθήματος

Όπως έχει αναφερθεί και σε προηγούμενα κεφάλαια, η ανάλυση συναισθήματος σε γραπτά δεδομένα εκμεταλλεύεται τις μεθόδους της ανάλυσης κειμένου. Η ανάλυση συναισθήματος εφαρμόζεται για παράδειγμα σε κριτικές ταινιών που υπάρχουν αναρτημένες δημόσια σε διάφορες ιστοσελίδες προκειμένου να αποφανθούν αν η κριτική για την εκάστοτε ταινία είναι ευνοϊκή ή μη ευνοϊκή. Η διαδικασία αυτή απαιτεί ετικετοποιημένα δεδομένα ή καθορισμό της συναισθηματικότητας των λέξεων. Στις βάσεις δεδομένων Wordnet και Conceptnet έχουν δημιουργηθεί δεδομένα για τη συναισθηματικότητα των λέξεων και των εννοιών.

Περισσότερες πληροφορίες παρέχονται στο άρθρο (“Text mining”)

5. ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΚΕΙΜΕΝΟΥ (TEXT CLASSIFICATION)

5.1 Εισαγωγή

Τις τελευταίες δεκαετίες με την κατακόρυφη άνοδο της πληροφορικής και των τηλεπικοινωνιών, η διακίνηση των πληροφοριών διευκολύνθηκε σε τόσο μεγάλο βαθμό που διαμόρφωσε την καθημερινότητα και την εξέλιξη της σύγχρονης κοινωνίας. Με τη ραγδαία αύξηση του όγκου της πληροφορίας, καθίσταται αναγκαία η επιβολή μεθόδων για την αποτελεσματική διαχείρισή της. Πληροφορίες με τη μορφή μη δομημένων κειμένων υπάρχουν παντού, όπως στα μηνύματα ηλεκτρονικής αλληλογραφίας, σε συνομιλίες στα μέσα κοινωνικής δικτύωσης, σε ιστοτόπους. Όμως είναι εξαιρετικά δύσκολο να ανακτηθεί χρήσιμη πληροφορία από τέτοιου είδους κείμενα εκτός αν είναι οργανωμένα με ένα συγκεκριμένο τρόπο. Κάτι τέτοιο μέχρι πρότινος αποτελούσε μια αρκετά δύσκολη και δαπανηρή διαδικασία καθώς απαιτούσε χρόνο και πόρους για τη μη αυτόματη ταξινόμηση των πληροφοριών και τη δημιουργία από τον άνθρωπο κανόνων που δύσκολα διατηρούνται και προσαρμόζονται. Ένας επιστημονικός κλάδος που ασχολείται με τη διαχείριση των πληροφοριών και επιχειρεί την εύκολη πρόσβαση σε αυτές και την αποτελεσματική αξιοποίησής τους, είναι η Αυτόματη Κατηγοριοποίηση Κειμένου. Η Αυτόματη κατηγοριοποίηση κειμένου στοχεύει στη δυνατότητα των χρηστών να έχουν εύκολη πρόσβαση στην πληθώρα των δεδομένων με την έννοια ότι ασχολείται με την ανάθεση κειμένων, γραμμένων σε φυσική γλώσσα, σε ένα σύνολο από προκαθορισμένες κατηγορίες με βάση το περιεχόμενό τους. Με τη χρήση της μηχανικής μάθησης, το αντικείμενο έρευνας των επιστημόνων έχει στραφεί στη δημιουργία ταξινομητών οι οποίοι έχουν τη δυνατότητα να κατηγοριοποιήσουν κείμενα σε ηλεκτρονική μορφή αυτόματα μέσω της μάθησης από τα χαρακτηριστικά ήδη ταξινομημένων κειμένων. Οι ταξινομητές κειμένου με επεξεργασία φυσικής γλώσσας (NLP) έχει αποδειχθεί μια εξαιρετική λύση για τη διαδικασία δόμησης κειμένων με γρήγορο, αποτελεσματικό, οικονομικό και επεκτάσιμο τρόπο. Η ταξινόμηση κειμένων γίνεται όλο και σημαντικότερος κλάδος για τις επιχειρήσεις καθώς επιτρέπει την απόκτηση πληροφοριών από δεδομένα και την αυτοματοποίηση των επιχειρησιακών διαδικασιών. Μερικά από τα περισσότερο συνηθισμένα παραδείγματα και περιπτώσεις χρήσης της αυτόματης ταξινόμησης κειμένου είναι τα παρακάτω:

- Ανάλυση συναισθήματος: Η διαδικασία κατά την οποία ο ταξινομητής στοχεύει να αποφανθεί αν ένα κείμενο αναφέρεται θετικά ή αρνητικά σχετικά με ένα αντικείμενο ή ένα θέμα συζήτησης. Η ανάλυση συναισθήματος διευκολύνει σε μεγάλο βαθμό το έργο των επιχειρήσεων που επιθυμούν να λάβουν ανάδραση από το κοινό για τα προϊόντα τους ώστε να βελτιώσουν και να επαναπροσδιορίσουν τους στόχους τους.

- Ανίχνευση θέματος (Topic Detection): Η διαδικασία εντοπισμού του θέματος στο οποίο αναφέρεται ένα τμήμα κειμένου. Για παράδειγμα, η γνώση αν μια κριτική πελατών για ένα προϊόν αναφέρεται σε θέματα ευχρηστίας, κόστους ή εξυπηρέτησης από τους αρμόδιους.
- Ανίχνευση γλώσσας κειμένου (Language Detection): Η διαδικασία ανίχνευσης της γλώσσας στην οποία είναι γραμμένο ένα κείμενο. Για παράδειγμα αν ένα άρθρο είναι γραμμένο στα αγγλικά ή στα γερμανικά.

5.2 Ορισμός του προβλήματος

Η ταξινόμηση κειμένου αναφέρεται στη διαδικασία ανάθεσης μιας τάξης c_j σε ένα κείμενο d_i όπου $c_j \in C$ και $d_i \in \Delta$ με $C = \{c_1,..., c_j)$ το σύνολο των προκαθορισμένων τάξεων (κατηγοριών) και Δ το πεδίο των κειμένων. Βασικός στόχος είναι η δημιουργία μιας συνάρτησης $h: \Delta \to C$η οποία προσεγγίζει όσο το δυνατόν περισσότερο τη συνάρτηση στόχο $h': \Delta \to C$.Η συνάρτηση στόχος περιλαμβάνει τη γνώση σύμφωνα με την οποία ταξινομούνται τα κείμενα. Η συνάρτηση h αποτελεί ουσιαστικά τον ταξινομητή (classifier) των κειμένων. Ο ταξινομητής, επίσης, μπορεί να θεωρηθεί ως μια συνάρτηση$h(d, c) \to \{T, F\}$ η οποία απαντά με την τιμή T αν το κείμενο d ανήκει στην κατηγορία c και με την τιμή F αν το κείμενο d δεν ανήκει στην κατηγορία c. Ορισμένοι ταξινομητές επίσης παρέχουν στην έξοδό τους την πιθανότητα με την οποία ένα κείμενο d ανήκει στην τάξη c. Ανάλογα με τις απαιτήσεις του προβλήματος μπορεί σε ένα κείμενο να χρειαστεί να ανατεθούν περισσότερες από μια κατηγορίες c_j. Αυτό σημαίνει ότι μπορεί να χρειαστεί να ανατεθούν k τιμές από το σύνολο $C = \{c_1,..., c_j)$ σε ένα κείμενο. Στην ειδική περίπτωση όπου k=1 τότε η ταξινόμηση ονομάζεται *ταξινόμηση μονής ετικέτας (single-label classification)*. Μάλιστα όταν το σύνολο C περιέχει ακριβώς δύο κατηγορίες, δηλαδή $|C| = 2$, τότε πρόκειται για τη λεγόμενη *δυαδική ταξινόμηση*. Όταν $|C| > 2$τότε το πρόβλημα αναφέρεται ως ταξινόμηση πολλαπλών τάξεων (multi-class classification). Στη γενική περίπτωση, το πρόβλημα στο οποίο σε ένα κείμενο d θα μπορεί να ανατεθεί ένα σύνολο τάξεων Y_i όπου $\left|Y_i\right| \leq |C|$ ονομάζεται *ταξινόμηση πολλαπλών ετικετών (multi-label classification)*.

Πηγή (Κατάκης)

5.3 Επιλογή Γνωρισμάτων (Feature Selection) της επεξεργασίας για την ταξινόμηση των κειμένων

Κατά τη δημιουργία προβλεπτικών μοντέλων μηχανικής μάθησης σπανιότατα χρησιμεύουν όλες οι μεταβλητές των δεδομένων. Η εισαγωγή περιττών μεταβλητών μειώνει την ικανότητα γενίκευσης και τελικά την τελική απόδοση και ακρίβεια του

μοντέλου. Επομένως, η προσθήκη όλο και περισσότερων μεταβλητών στο μοντέλο καταλήγει στην αύξηση της πολυπλοκότητας του μοντέλου και δυσχεραίνει το έργο της μάθησης.

Στόχος της επιλογής γνωρισμάτων (feature selection) είναι η εύρεση ενός συνόλου γνωρισμάτων από τα δεδομένα που διευκολύνει τη δημιουργία αποδοτικών μοντέλων. Η κατάλληλη επιλογή γνωρισμάτων μπορεί να οδηγήσει στη μείωση του όγκου των δεδομένων εκπαίδευσης που απαιτούνται προκειμένου να επιτευχθεί καλύτερο αποτέλεσμα στο μοντέλο. Επίσης, μπορεί να βελτιώσει σε σημαντικό βαθμό την ακρίβεια της ταξινόμησης. Τα γνωρίσματα αυτά έχουν τη δυνατότητα να διατηρηθούν και να βοηθήσουν στη μελλοντική χρήση και εξέλιξη του ταξινομητή. Μια κακή επιλογή γνωρισμάτων, από την άλλη πλευρά, μπορεί να οδηγήσει σε περιορισμό της απόδοσης της ταξινόμησης καθώς τροφοδοτεί με μη χρήσιμα δεδομένα το μοντέλο.

Η επιλογή των γνωρισμάτων θεωρείται η διαδικασία στην οποία κάθε χαρακτηριστικό (λέξη) των κειμένων αποκτά μια βαθμολογία με βάση κάποιο κανόνα μέτρησης. Αφού η διαδικασία βαθμολόγησης όλων των λέξεων των κειμένων ολοκληρωθεί, τότε επιλέγεται το σύνολο των κατάλληλων γνωρισμάτων με βάση κάποιο κριτήριο αξιολόγησης της βαθμολογίας. Ένα τέτοιο κριτήριο είναι η συχνότητα εμφάνισης των λέξεων τα κείμενα.

Δεδομένου ότι τα κείμενα έχουν υποστεί προεπεξεργασία υποθέτουμε ότι έχουν πραγματοποιηθεί διαδικασίες όπως stemming, decapitalization, stopwords removal και ο όγκος των λέξεων έχει μειωθεί ήδη αρκετά. Κατόπιν, κατά την επιλογή των γνωρισμάτων πρέπει να ληφθούν υπόψη ορισμένα φίλτρα τα οποία μειώνουν περαιτέρω τον όγκο των λέξεων χωρίς να επηρεάζεται η ικανότητα ανάκτησης χρήσιμης πληροφορίας. Οι σπανίως εμφανιζόμενες λέξεις στο σύνολο δεδομένων επεξεργασίας δεν προσδίδουν χρήσιμη πληροφορία και συνεπώς δεν κρίνεται αναγκαία η χρήση τους στη διαδικασία της ταξινόμησης. Για το λόγο αυτό, αφού τα δεδομένα διαιρεθούν σε σύνολο εκπαίδευσης και σε σύνολο ελέγχου, οι λέξεις οι οποίες εμφανίζονται με πολύ μικρή συχνότητα στα έγγραφα περιορίζονται και τελικά προκύπτουν σημαντικά οφέλη.

Οι τεχνικές που υπάρχουν για επιλογή γνωρισμάτων από έγγραφα ταξινομούνται στις εξής κατηγορίες:

- Επιβλεπόμενες Τεχνικές: Αυτές οι τεχνικές εφαρμόζονται σε δεδομένα των οποίων είναι γνωστή η ετικέτα (labeled data) και χρησιμοποιούνται για να ταυτοποιηθούν τα γνωρίσματα ώστε να ενισχυθεί η αποδοτικότητα του μοντέλου.
- Μη επιβλεπόμενες Τεχνικές: Οι συγκεκριμένες τεχνικές χρησιμοποιούνται για δεδομένα των οποίων δεν είναι γνωστή η ετικέτα (unlabeled data).

Οι τεχνικές που βοηθούν στην καλύτερη εποπτεία των δεδομένων και επομένως χρησιμοποιούνται για την επιλογή γνωρισμάτων χωρίζονται σε τρεις κατηγορίες: τις

μεθόδους φίλτρου (filter methods), τις μεθόδους περιτυλίγματος (wrapped methods) και τις ενσωματωμένες μεθόδους (embedded methods).

Επιγραμματικά αναφέρονται οι μέθοδοι παρακάτω:

α) Μέθοδοι Φίλτρα

- ❏ Information Gain
- ❏ Chi-Square Test
- ❏ Fisher's Score
- ❏ Correlation Coefficient
- ❏ Variance threshold
- ❏ Mean Absolute Difference (MAD)
- ❏ Dispersion Ratio

β) Μέθοδοι περιτυλίγματος (wrapped methods)

- ❏ Forward Feature Selection
- ❏ Backward Feature Elimination
- ❏ Exhaustive Feature Selection
- ❏ Recursive Feature Elimination

γ) Μέθοδοι ενσωμάτωσης (embedded methods)

- ❏ LASSO Regularization (L_1)
- ❏ Random Forest Importance

Πηγή: ("Feature Selection Techniques in Machine Learning")

5.4 Εξαγωγή Χαρακτηριστικών (Feature Extraction) από τα δεδομένα

Στη μηχανική μάθηση με τον όρο *διαστατικότητα (dimensionality)* αναφερόμαστε στο πλήθος των χαρακτηριστικών στο σύνολο δεδομένων (dataset). Η λεγόμενη *κατάρα της διαστατικότητας (curse of dimensionality)* αναφέρεται στην ύπαρξη τεράστιου αριθμού χαρακτηριστικών (features) συγκριτικά με τον αριθμό των παρατηρήσεων των δεδομένων και η παρουσία της δυσκολεύει σε μεγάλο βαθμό το έργο των ταξινομητών. Η μείωση διαστατικότητας στα δεδομένα είναι πλέον πολύ σημαντικό μέρος των διαδικασιών της μηχανικής μάθησης και επιτυγχάνεται με την επιλογή και την εξαγωγή γνωρισμάτων (Feature Selection and Feature Extraction). Η βασική διαφορά της επιλογής και της εξαγωγής γνωρισμάτων είναι ότι η επιλογή γνωρισμάτων διατηρεί ένα υποσύνολο των αρχικών γνωρισμάτων ενώ η εξαγωγή δημιουργεί ένα καινούριο.

Η εξαγωγή χαρακτηριστικών είναι η διαδικασία εξαγωγής μια λίστας λέξεων από τα δεδομένα κειμένου και στη συνέχεια η μετατροπή τους σε ένα νέο σύνολο χαρακτηριστικών το οποίο μπορεί να χρησιμοποιηθεί από έναν ταξινομητή. Οι συνηθέστερες και απλούστερες μέθοδοι εξαγωγής χαρακτηριστικών παρατίθενται στη συνέχεια. Κάθε αλγόριθμος επιβλεπόμενης μάθησης απαιτεί ένα έγγραφο να αναπαρίσταται από ένα διάνυσμα χαρακτηριστικών ώστε να πραγματοποιηθεί η διαδικασία της εκμάθησης στο έγγραφο αυτό. Αυτό γίνεται μέσω της διαδικασίας Vector Space Modeling (VSM).

Η διαδικασία Vector Space Modelling μπορεί να εφαρμοστεί μέσω δύο τεχνικών οι οποίες χρησιμοποιούνται για την εξαγωγή χαρακτηριστικών από έγγραφα. Συγκεκριμένα:

- **Bag of Words**

Ο σάκος των λέξεων (Bag of Words) αποτελεί την πιο κοινή και απλούστερη μέθοδο εξαγωγής χαρακτηριστικών από κείμενο. Σχηματίζει ένα σύνολο χαρακτηριστικών από όλες τις λέξεις που υπάρχουν στο κείμενο. Καλείται σάκος (bag) καθώς δε λαμβάνει υπόψη τη σειρά εμφάνισης των λέξεων στο κείμενο αλλά ενδιαφέρεται μόνο για την ύπαρξή ή μη της συγκεκριμένης λέξης. Τα χαρακτηριστικά αυτά χρησιμοποιούνται για την κατασκευή μοντέλων καθώς αυτή η μέθοδος είναι εξαιρετικά απλή και ευέλικτη. Συνήθως χρησιμοποιείται για να εξαχθούν χαρακτηριστικά από έγγραφα με διάφορους τρόπους. Ο σάκος των λέξεων αποτελεί ουσιαστικά την αναπαράσταση των δεδομένων κειμένου. Διευκολύνει τον προσδιορισμό της συχνότητας εμφάνισης των λέξεων στο κείμενο καθώς περιλαμβάνει ένα λεξικό των γνωστών λέξεων που υπάρχουν και επιπλέον την πληροφορία εμφάνισης ή όχι καθεμίας από τις γνωστές αυτές λέξεις σε κάθε κείμενο.

- **TF-IDF**

Η προσέγγιση ενός προβλήματος με τη φιλοσοφία του Bag of Words είναι πως οι λέξεις με μεγάλη συχνότητα εμφάνισης στα έγγραφα παίζουν το σημαντικότερο ρόλο στην εξαγωγή πληροφορίας και επομένως στη διαδικασία της ταξινόμησης. Ωστόσο, σε πολλές περιπτώσεις οι λέξεις αυτές μπορεί να μην παρέχουν χρήσιμη πληροφορία στο μοντέλο και για το λόγο αυτό θα πρέπει να αφαιρεθούν ή να αγνοηθούν. Η επίλυση του προβλήματος αυτού έγκειται στον επαναπροσδιορισμό της συχνότητας εμφάνισης της κάθε λέξης σε όλα τα κείμενα αυτή τη φορά. Εξαιτίας αυτού, οι βαθμολογίες των συχνά εμφανιζόμενων όρων σε όλα τα κείμενα μειώνονται. Αυτός ο τρόπος βαθμολόγησης των συχνοτήτων αναφέρεται ως Term Frequency - Inverse Document Frequency (TF-IDF). Αναλυτικότερα,

- Term Frequency (TF) : Είναι η συχνότητα εμφάνισης μιας λέξης στο τρέχον έγγραφο

- ❏ Inverse Document Frequency (IDF) : Είναι η βαθμολογία των λέξεων μεταξύ όλων των εγγράφων

Με τις τιμές αυτές υπάρχει η δυνατότητα εύρεσης των όρων οι οποίοι είναι ικανοί να παρέχουν σημαντική πληροφορία για το κείμενο. Επιπλέον, μια υψηλή τιμή του IDF δείχνει ότι μια λέξη δεν είναι τόσο συχνή, ενώ μια χαμηλή τιμή του IDF δηλώνει ότι η συγκεκριμένη λέξη έχει μεγάλη συχνότητα εμφάνισης. Με τον τρόπο αυτό εξαιρούνται συχνά εμφανιζόμενες λέξεις οι οποίες δεν προσδίδουν πληροφορία και αυξάνουν ταυτόχρονα τον όγκο των χαρακτηριστικών.

Οι τιμές TF, IDF υπολογίζονται σε δύο πίνακες για κάθε όρο και κάθε κείμενο με βάση τις εξής σχέσεις:

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

$$IDF(t) = log(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ term\ t\ in\ it})$$

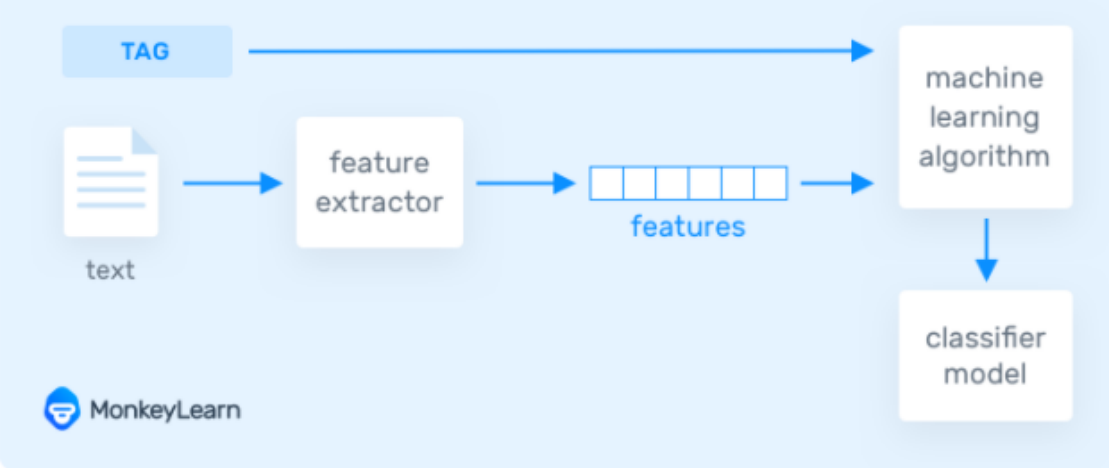
Η κανονικοποιημένη τιμή για τη βαθμολόγηση των λέξεων προκύπτει από το γινόμενο των δύο ανωτέρω σχέσεων και τελικά καθορίζει τη σημαντικότητα κάθε λέξης η οποία περιέχει το νόημα του εγγράφου.

Πηγές: (“Document feature extraction and classification”) (Resham N. Waykole and Anuradha D. Thakare #)

5.5 Αλγόριθμοι ταξινόμησης

Η ταξινόμηση κειμένου, αντί να πραγματοποιείται και να βασίζεται σε χειρόγραφους κανόνες όπως παλαιότερα, μαθαίνει να ταξινομεί σύμφωνα με προηγούμενες παρατηρήσεις. Χρησιμοποιώντας κείμενα, τα οποία έχουν ήδη γνωστή ετικέτα, ως δεδομένα εκπαίδευσης, ο ταξινομητής εντοπίζει συσχετίσεις που υπάρχουν σε τμήματα του κειμένου και μαθαίνει ποια στοιχεία των κειμένων (text) αναμένεται να αφορούν κάθε κατηγορία (tag). Με τον τρόπο αυτό το κάθε κείμενο αποκτά μια ετικέτα η οποία ουσιαστικά αποτελεί μια προκαθορισμένη κατηγορία στην οποία οποιοδήποτε κείμενο μπορεί να ενταχθεί. Κατόπιν της προεπεξεργασίας των κειμένων με τις διάφορες τεχνικές οι οποίες αναλύθηκαν παραπάνω, τα καθαρά πλέον δεδομένα παρέχονται ως είσοδοι σε αλγορίθμους ταξινόμησης ώστε να συντεθούν τα κατάλληλα προβλεπτικά μοντέλα.

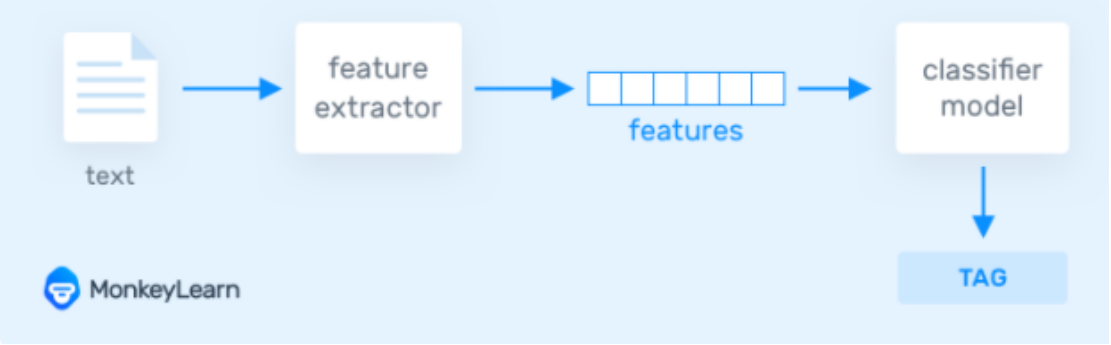
Η ροή και τα βήματα της διαδικασίας της εκπαίδευσης του μοντέλου φαίνεται στο Σχ.5.1.



Σχ.5.1 Διαδικασία εκμάθησης μοντέλου

Πηγή: ("Text Classification")

Εφόσον το μοντέλο εκπαιδευτεί με αρκετά δεδομένα εκπαίδευσης είναι ικανό να αρχίσει να πραγματοποιεί ακριβείς προβλέψεις. Η διαδικασία της πρόβλεψης φαίνεται στο Σχ.5.2.



Σχ.5.2 Διαδικασία πρόβλεψης

Πηγή: ("Text Classification")

Η ταξινόμηση κειμένου με τη χρήση μηχανικής μάθησης είναι συνήθως πολύ περισσότερο ακριβής συγκριτικά με ανθρώπινα σχεδιασμένα συστήματα κανόνων ειδικά σε πολύπλοκες περιπτώσεις ταξινόμησης επεξεργασίας φυσικής γλώσσας. Επιπλέον, οι ταξινομητές που βασίζονται σε μηχανική μάθηση έχουν την ικανότητα να διατηρούνται ώστε ο αναλυτής να μπορεί να εντάξει ανά πάσα στιγμή νέες εργασίες και νέα δεδομένα.

Οι αλγόριθμοι ταξινόμησης κειμένου επιλέγονται από τον αναλυτή από ένα ευρύ σύνολο αλγορίθμων σύμφωνα με τις ανάγκες του προβλήματος και του είδους των

δεδομένων. Αξίζει να σημειωθεί ότι δεν υπάρχει αποδοτικότερος ή μη αποδοτικότερος αλγόριθμος στο σύνολό του. Η αποτελεσματικότητα και η αποδοτικότητα των αλγορίθμων μεταβάλλονται ανάλογα με τα δεδομένα που δέχονται ως είσοδο και μετρώνται με ορισμένες μετρικές οι οποίες θα αναλυθούν στη συνέχεια. Ορισμένοι από τους δημοφιλέστερους αλγόριθμους ταξινόμησης κειμένου είναι ο Naive Bayes, ο Support Vector Machine (SVM), ο Random Forest και άλλοι που θα αναλυθούν λεπτομερώς στη συνέχεια.

5.5.1 Naive Bayes Classifier

Ο αλγόριθμος Naive Bayes είναι μια οικογένεια πιθανολογικών μοντέλων μηχανικής μάθησης τα οποία χρησιμοποιούνται για προβλήματα ταξινόμησης και η κεντρική τους ιδέα βασίζεται στο θεώρημα του Bayes. Με βάση το Θεώρημα Bayes, ο αλγόριθμος προβλέπει την κατηγορία ή την ετικέτα στην οποία ανήκει κάθε κείμενο (πχ ένα άρθρο μιας εφημερίδας, μια κριτική πελατών κλπ). Είναι πιθανολογικός με την έννοια ότι υπολογίζει την πιθανότητα της κάθε ετικέτας για κάθε κείμενο και τελικά στην έξοδό του καταλήγει στην ετικέτα με τη μεγαλύτερη πιθανότητα. Η πιθανότητα αυτή υπολογίζεται με βάση το Θεώρημα του Bayes το οποίο αποδίδει την πιθανότητα ενός χαρακτηριστικού (feature) δεδομένης της πρότερης γνώσης των συνθηκών που σχετίζονται με το χαρακτηριστικό αυτό. Ο συγκεκριμένος αλγόριθμος απλοποιεί σημαντικά τη διαδικασία της μάθησης καθώς θέτει την υπόθεση ότι τα χαρακτηριστικά είναι ανεξάρτητα μεταξύ τους. Παρόλο που η υπόθεση αυτή σπανιότατα έως ποτέ ανταποκρίνεται στην πραγματικότητα, ο αλγόριθμος ανταγωνίζεται σε σημαντικό βαθμό άλλους περισσότερο εξελιγμένους αλγόριθμους.

Θεώρημα του Bayes

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Με τη βοήθεια του θεωρήματος Bayes μπορεί να υπολογιστεί η πιθανότητα ενός ενδεχομένου A δεδομένου ενός συμβάντος B. Στην περίπτωση της ταξινόμησης, το συμβάν B είναι η απόδειξη και το A είναι η υπόθεση. Η παραδοχή όπως αναφέρθηκε προηγουμένως είναι πως η ύπαρξη ενός χαρακτηριστικού δεν επηρεάζει τα υπόλοιπα. Για το λόγο αυτό ο αλγόριθμος καλείται αφελής (naive).

Ο αλγόριθμος εκχωρεί μια πιθανή τάξη σε ένα δεδομένο παράδειγμα το οποίο αντιπροσωπεύεται από το διάνυσμα των χαρακτηριστικών του $X = \{X_1,... X_n\}$και παρά την παραδοχή του, καταλήγει σε αρκετά αξιόπιστα αποτελέσματα. Έχει αποδειχθεί αρκετά αποτελεσματικός σε προβλήματα ταξινόμησης κειμένου, ιατρικών διαγνώσεων και διαχείρισης της επίδοσης συστημάτων. Ο συγκεκριμένος αλγόριθμος αποδίδει εξίσου καλά και σε προβλήματα με μεγάλη εξάρτηση των χαρακτηριστικών, παρά την παραδοχή, καθώς η απόδοσή του δε σχετίζεται με το βαθμό ανεξαρτησίας

των χαρακτηριστικών αλλά με την ικανότητά του να ταιριάζει το πραγματικό με την υπόθεση στις περισσότερες περιπτώσεις.

Αναλυτικότερα, ο Naive Bayes λειτουργεί ως εξής:

Δεδομένου ενός προβλήματος που πρέπει να ταξινομηθεί το οποίο αναπαρίσταται από ένα διάνυσμα n ανεξάρτητων χαρακτηριστικών $\mathrm{X} = \{\mathrm{X}_1,\ldots \mathrm{X}_n\}$, ο αλγόριθμος εκχωρεί σε κάθε ετικέτα C_k μια πιθανότητα εμφάνισης $P(C_k|\, \mathrm{X}_1,\ldots \mathrm{X}_n)$ για καθεμία από τις τάξεις C_k.

Το μειονέκτημα αυτής της προσέγγισης είναι ότι σε περίπτωση που υπάρχει μεγάλος αριθμός χαρακτηριστικών n ή κάθε χαρακτηριστικό παίρνει μεγάλο πλήθος τιμών τότε η επίλυση με πίνακες πιθανότητας καθίσταται πρακτικά ανέφικτη. Επομένως, αναπροσαρμόζοντας τη διατύπωση του μοντέλου, κάνουμε χρήση της δεσμευμένης πιθανότητας η οποία υπολογίζεται κατά Bayes ως εξής:

$$P(C_k|\, X) \; = \frac{P(X|C_k)P(C_k)}{P(X)}$$

Σε επίπεδο γλώσσας η παραπάνω σχέση μεταφράζεται ως:

$$\text{Μεταγενέστερο} \; = \frac{\text{Προγενέστερο} \times \text{πιθανότητα}}{\text{Απόδειξη}}$$

Στην πράξη, ενδιαφέρον παρουσιάζεται ουσιαστικά στον αριθμητή του κλάσματος της παραπάνω σχέσης καθώς ο παρονομαστής είναι ανεξάρτητος της ετικέτας C_k και, δεδομένου ότι οι τιμές των χαρακτηριστικών είναι συγκεκριμένες, παραμένει σταθερός.

Ο αριθμητής είναι ισοδύναμος με την από κοινού πιθανότητας (joint probability):

$P(C_k, \mathrm{X}_1,\ldots \mathrm{X}_n)$,

η οποία μπορεί να γραφεί ως εξής αν επαναλαμβανόμενα εφαρμόσουμε τον κανόνα της αλυσίδας για τη δεσμευμένη (κατά συνθήκη) πιθανότητα:

$$P(C_k, \mathrm{X}_1,\ldots \mathrm{X}_n) = P(\mathrm{X}_1,\ldots, \mathrm{X}_n, C_k) \; = \; P(X_1|\, X_2,\ldots, X_{n,} C_k)P(\, X_2,\ldots, X_{n,} C_k) \; =$$

$$P(X_1|\, X_2,\ldots, X_{n,} C_k)P(\, X_2|\mathrm{X}_3,\ldots, X_{n,} C_k)P(\mathrm{X}_3,\ldots, X_{n,} C_k) \; =\ldots$$

$$=P(X_1|\, X_2,\ldots, X_{n,} C_k)P(\, X_2|\mathrm{X}_3,\ldots, X_{n,} C_k)\ldots P(\, X_{n-1}|\, X_n, C_k)P(\, X_n|C_k)P(C_k)$$

Αν στους υπολογισμούς εντάξουμε και την ανεξαρτησία των χαρακτηριστικών $\mathrm{X}_1,\ldots \mathrm{X}_n$ δεδομένης μιας κατηγορίας C_k τότε έχουμε το εξής:

$$P(\mathrm{X}_i|\, \mathrm{X}_{i+1},\ldots, \mathrm{X}_n, C_k) \; = \; P(\mathrm{X}_i|\, C_k)$$

Επομένως, η από κοινού πιθανότητας (joint probability) γράφεται ως εξής:

$P(C_k|\,X) \propto P(C_k, \mathrm{X}_1, \ldots \mathrm{X}_n)$

$\propto P(C_k)P(\mathrm{X}_1|C_k)P(X_2, C_k)\ldots$

$\propto P(C_k)\prod_{i=1}^{n} P(X_i, C_k)$, όπου $\propto$ συμβολισμός αναλογίας

Συνεπώς, σύμφωνα με τις υποθέσεις περί ανεξαρτησίας των χαρακτηριστικών, η δεσμευμένη πιθανότητα της κλάσης C προσδιορίζεται ως εξής:

$P(C_k|\,X) = \frac{P(C_k)\prod_{i=1}^{n} P(X_i C_k)}{\mathrm{Z}}$, όπου Z=P(X) ένας σταθερός παράγοντας ο οποίος εξαρτάται μόνο από τις τιμές των χαρακτηριστικών και είναι γνωστός αν τα χαρακτηριστικά είναι επίσης γνωστά.

Στους μέχρι στιγμής υπολογισμούς μας προσδιορίσαμε τη σχέση που δίνει την πιθανότητα κάποιας κλάσης δεδομένων ορισμένων χαρακτηριστικών. Ωστόσο, ο ταξινομητής δεν περιορίζεται μόνο στους υπολογισμούς αυτούς. Εφαρμόζει κατόπιν έναν κανόνα απόφασης σύμφωνα με τον οποίο καθορίζεται η έξοδος του μοντέλου. Ο πιο συνήθης κανόνας για την απόφαση της ετικέτας κλάσης είναι η επιλογή της ετικέτας με τη μέγιστη τιμή όλων των πιθανοτήτων.

Για να εμβαθύνουμε στη φιλοσοφία του αλγορίθμου ας θεωρήσουμε το παρακάτω παράδειγμα το οποίο περιλαμβάνει 5 προτάσεις ως δεδομένα εκπαίδευσης:

Text	Tag
"A great game"	Sports
"The election was over"	Not sports
"Very clean match"	Sports
"A clean but forgettable game"	Sports
"It was a close election"	Not sports

Το ζητούμενο είναι η κλάση της πρότασης: *A very close game*

Σύμφωνα με την πιθανολογική φιλοσοφία του αλγορίθμου, θα υπολογιστεί η πιθανότητα η πρόταση να ανήκει στην κλάση *Sports,* έπειτα η πιθανότητα να ανήκει στην κλάση *Not Sports* και τελικά θα επιλεγεί η κλάση με τη μεγαλύτερη πιθανότητα. Αγνοώντας στην παρούσα κατάσταση την προεπεξεργασία δεδομένων και την επιλογή χαρακτηριστικών για τις ανάγκες του παραδείγματος, έχουμε τους εξής μαθηματικούς υπολογισμούς:

$$P(Sports \mid a\ very\ close\ game) = \frac{P(a\ very\ close\ game \mid Sports) \times P(Sports)}{P(a\ very\ close\ game)}$$

Στη δεσμευμένη πιθανότητα $P(a\ very\ close\ game \mid Sports)$ καλούμαστε να αναζητήσουμε πόσες φορές εμφανίζεται η συγκεκριμένη πρόταση στα δεδομένα εκπαίδευσης και τελικά φαίνεται ότι αυτούσια δεν υπάρχει, όπως συμβαίνει και στα περισσότερα προβλήματα ταξινόμησης. Στο σημείο αυτό εκμεταλλευόμαστε την <<αφέλεια>> του αλγορίθμου ώστε να υποθέσουμε ότι κάθε λέξη μέσα στην πρόταση είναι ανεξάρτητη από τις υπόλοιπες. Αυτό σημαίνει ότι πλέον δεν αναζητούμε ολόκληρη πρόταση στα δεδομένα παρά μεμονωμένες λέξεις.

Επομένως έχουμε την πιθανότητα των χαρακτηριστικών:

$P(a\ very\ close\ game) = P(a) \times P(very) \times P(close) \times P(game)$ και τελικά:

$$P(a\ very\ close\ game \mid Sports) = P(a|Sports) \times P(very|Sports) \times P(close|Sports) \times P(game|Sports) = 0.0000276$$

$$P(a\ very\ close\ game \mid Not\ Sports) = P(a|Not\ Sports) \times P(very|Not\ Sports) \times P(close|Not\ Sports) \times P(game|Not\ Sports) =$$

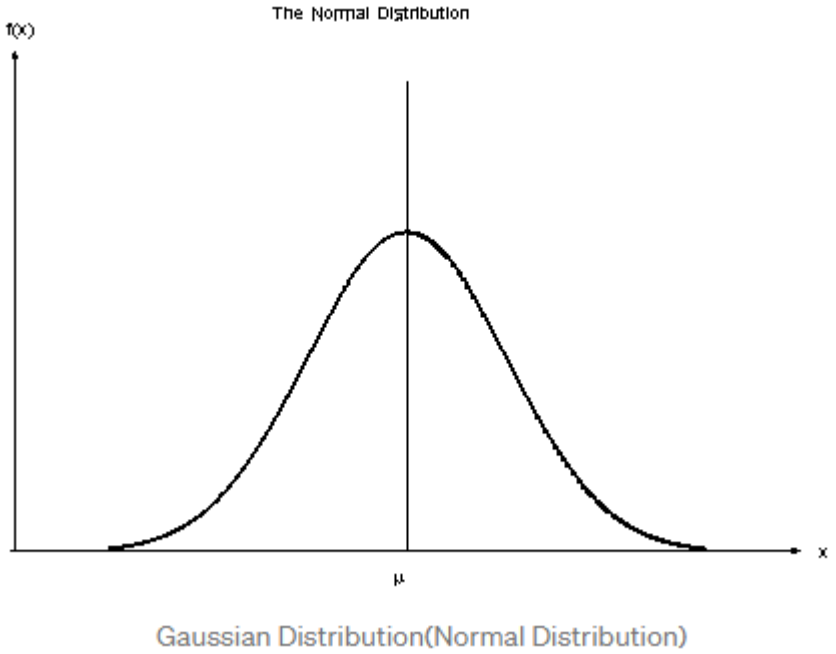
0.00000572

Τελικά, σύμφωνα με τον κανόνα απόφασης της μέγιστης πιθανότητας ετικέτας, η πρόταση *A very close game* αποκτά την ετικέτα *Sports.*

Υπάρχουν τρεις τύποι μοντέλων Naive Bayes στη βιβλιοθήκη Scikit-learn:

- Multinomial Naive Bayes : Κυρίως χρησιμοποιείται για ταξινόμηση εγγράφων, για παράδειγμα αν ένα έγγραφο ανήκει στην κατηγορία αθλητικών, πολιτικών, τεχνολογικών κλπ.
- Bernoulli Naive Bayes: Είναι παρόμοια με τα μοντέλα Multinomial Naive Bayes με τη διαφορά ότι οι τιμές πρόβλεψης είναι υποχρεωτικά δυαδικές (boolean). Οι παράμετροι που χρησιμοποιούνται για να προβλέψουν την τιμή μιας μεταβλητής λαμβάνουν μόνο τις τιμές ναι ή όχι, για παράδειγμα αν μια λέξη υπάρχει ή όχι μέσα στο κείμενο.
- Gaussian Naive Bayes: Στα συγκεκριμένα μοντέλα οι προβλέψεις λαμβάνουν συνεχείς και όχι διακριτές τιμές. Στην περίπτωση αυτή θεωρούμε ότι οι τιμές

που σχετίζονται με κάθε κλάση ακολουθούν την κατανομή του Gauss όπως φαίνεται στο σχήμα Σχ.3



Σχ.5.5.1 Κανονική Κατανομή (Κατανομή Gauss)

Πηγή: ("Naive Bayes Classifier")

Αν στα δεδομένα εκπαίδευσης υπάρχει το χαρακτηριστικό x σε συνεχή μορφή, τότε αρχικά τμηματοποιούνται τα δεδομένα σε κάθε κλάση και έπειτα υπολογίζεται η μέση τιμή μ_κ και η διακύμανση (Bessel Corrected Variance) σ^2_κτων τιμών του x που αφορούν την κλάση C_k. Έστω, ότι οι παρατηρήσεις μας έχουν μια τιμή v και επιθυμούμε να αποφανθούμε αν ανήκει ή οχι στην κλάση C_k. Τότε η δεσμευμένη πιθανότητα υπολογίζεται από τη σχέση της συνάρτησης πυκνότητας πιθανότητας της κανονικής κατανομής (Gaussian) σύμφωνα με τον τύπο:

$$P(v|C_k) = \frac{1}{\sqrt{2\pi\sigma^2_\kappa}} e^{-\frac{(v-\mu_\kappa)^2}{2\sigma^2_\kappa}}$$

Πηγές: ("Naive Bayes Classifier"), ("Naive Bayes Classifier"), ("A practical explanation of a Naive Bayes classifier")

Πλεονεκτήματα αλγορίθμου:

- Μεγάλη ταχύτητα λόγω της παραδοχής περί ανεξαρτησίας των χαρακτηριστικών. Αυτό αποτελεί πλεονέκτημα σε προβλήματα όπου είναι πολυτιμότερη η ταχύτητα παρά η απόλυτη ακρίβεια.
- Καταλληλότητα για επίλυση προβλημάτων πολλαπλών κατηγοριών (multi-class).
- Σε περίπτωση που ικανοποιείται η παραδοχή για την ανεξαρτησία των χαρακτηριστικών, παράγει προβλέψεις περισσότερο αξιόπιστες συγκριτικά με άλλα μοντέλα και ταυτόχρονα απαιτεί λιγότερα δεδομένα εκπαίδευσης.
- Είναι περισσότερο κατάλληλος για προβλήματα με κατηγορικές μεταβλητές εισόδου παρά με αριθμητικές μεταβλητές.

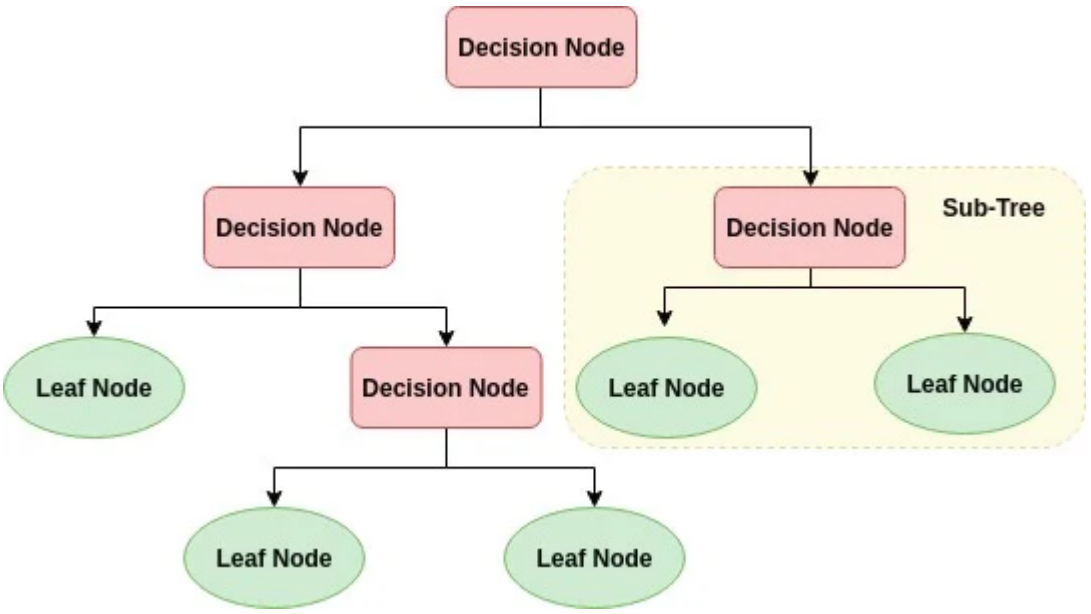
Μειονεκτήματα αλγορίθμου:

- Η παραδοχή του αλγορίθμου περί ανεξαρτησίας των χαρακτηριστικών σπανιότατα ανταποκρίνεται στην πραγματικότητα, γεγονός το οποίο περιορίζει την ακρίβειά του σε προβλήματα.
- Θέτει μηδενική συχνότητα σε μεταβλητές των οποίων η κατηγορία δεν υπήρχε στο σύνολο των δεδομένων εκπαίδευσης.

5.5.2 Decision Trees Classifier

Τα δένδρα απόφασης (Decision Tree) είναι ένας από τους απλούστερους και δημοφιλέστερους αλγόριθμους ταξινόμησης τόσο στην κατανόηση όσο και στην ερμηνεία του. Μπορεί να χρησιμοποιηθεί είτε σε προβλήματα ταξινόμησης (classification) είτε παλινδρόμησης (regression).

Ένα δένδρο απόφασης είναι ουσιαστικά ένα διάγραμμα ροής με τη μορφή δέντρου στο οποίο κάθε εσωτερικός κόμβος αποτελεί ένα χαρακτηριστικό (feature), κάθε κλάδος αποτελεί έναν κανόνα απόφασης και κάθε φύλλο αποτελεί ένα αποτέλεσμα. Ο αρχικός κόμβος ο οποίος βρίσκεται στην κορυφή του δένδρου απόφασης είναι γνωστός ως ρίζα (root node). Η φιλοσοφία του αλγορίθμου είναι πως η διαμέριση στους κόμβους πραγματοποιείται με αναδρομικό τρόπο και με βάση την τιμή κάποιου χαρακτηριστικού. Στο παρακάτω διάγραμμα φαίνεται η δομή ενός δένδρου απόφασης. Αξίζει να σημειωθεί ότι η λογική του συγκεκριμένου αλγορίθμου μιμείται σε μεγάλο βαθμό τον τρόπο με τον οποίο το ανθρώπινο μυαλό επεξεργάζεται τις πληροφορίες και λαμβάνει αποφάσεις. Για το λόγο αυτό ο αλγόριθμος είναι κατανοητός σε μεγάλο βαθμό από τους χρήστες.



Σχ. 5.5.2.1 *Γενική μορφή ενός δέντρου απόφασης*

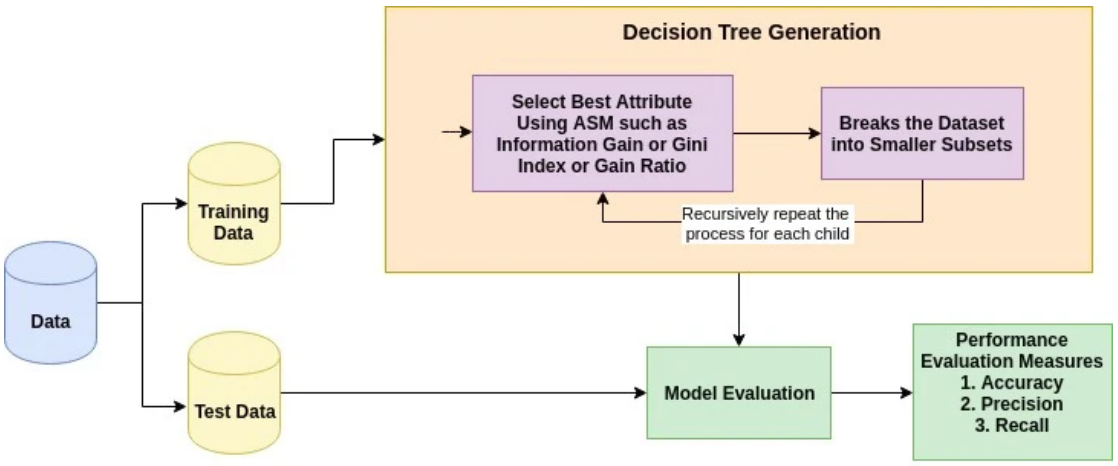
Πηγή: ("Decision Tree Classification in Python")

Η χρονική πολυπλοκότητα του αλγορίθμου είναι συνάρτηση του αριθμού των εγγράφων και του πλήθους των χαρακτηριστικών στο σύνολο των δεδομένων. Ο συγκεκριμένος αλγόριθμος θεωρείται μη παραμετροποιημένος και επιπλέον δεν εξαρτάται από κατανομές πιθανοτήτων συγκριτικά με άλλους αλγόριθμους ταξινόμησης. Τα δένδρα απόφασης έχουν τη δυνατότητα να διαχειρίζονται πολυδιάστατα δεδομένα με μεγάλη ακρίβεια.

Η κεντρική ιδέα του αλγορίθμου συνοψίζεται στα παρακάτω βήματα:

- Επιλογή των κατάλληλων χαρακτηριστικών με βάση τις μεθόδους επιλογής ώστε να χωριστούν τα δεδομένα
- Μετατροπή των χαρακτηριστικών σε κόμβους και διαίρεση του συνόλου δεδομένων σε μικρότερα υποσύνολα
- Σταδιακή δημιουργία του δένδρου απόφασης με επανάληψη της αναδρομικής διαδικασίας έως ότου να ικανοποιηθεί ένα από τα παρακάτω ενδεχόμενα:
  - ❏ Όλες οι πλειάδες να ανήκουν στην ίδια τιμή ενός χαρακτηριστικού.
  - ❏ Να μην υπάρχουν πλέον άλλα χαρακτηριστικά.
  - ❏ Να μην υπάρχουν άλλα κείμενα προς ταξινόμηση.

Στο παρακάτω σχήμα βλέπουμε τη διαδικασία που εκτελεί ο ταξινομητής έως ότου καταλήξει σε αποτελέσματα.



Σχ.5.5.2.2 *Διαδικασία του ταξινομητή Decision Tree*

Πηγή: ("Decision Tree Classification in Python")

Ο αλγόριθμος Decision Trees βρίσκεται στη διάθεση των αναλυτών μέσω της βιβλιοθήκης Scikit-Learn η οποία δίνει τη δυνατότητα βελτίωσής του μέσω ορισμένων παραμέτρων όπως:

- Criterion: κατ' επιλογήν (default = "gini") ή διαφορετική μέθοδος επιλογής χαρακτηριστικών: Η συγκεκριμένη παράμετρος δίνει τη δυνατότητα επιλογής της μετρικής σύμφωνα με την οποία επιλέγονται τα χαρακτηριστικά. Τα κριτήρια που υποστηρίζονται είναι το "gini" για τη μέθοδο Gino index και το "entropy" για το information gain.
- Splitter: string, κατ επιλογήν (default = "best") ή τακτική διαμέρισης: Η συγκεκριμένη παράμετρος μας επιτρέπει να επιλέξουμε την τακτική διαίρεσης σε κόμβους. Οι διαθέσιμες τακτικές είναι η "best" και η "random".
- max_depth: Int ή None, κατ' επιλογήν (default = "None). Η παράμετρος αυτή δίνει τη δυνατότητα επιλογής του μέγιστου βάθους του δέντρου.

Πλεονεκτήματα Αλγορίθμου:

- Ευκολία στην κατανόηση και την ερμηνεία
- Απαιτεί λιγότερη προεπεξεργασία των δεδομένων συγκριτικά με άλλους αλγόριθμους
- Μπορεί να χρησιμοποιηθεί για πρόβλεψη χαμένων ή άγνωστων τιμών, επιλογή μεταβλητών κλπ.
- Δεν περιλαμβάνει παραδοχές για κατανομές πιθανότητας στα δεδομένα.

Μειονεκτήματα αλγορίθμου:

- Ευαισθησία σε δεδομένα με θόρυβο

- Μικρές διακυμάνσεις στα δεδομένα οδηγούν σε τελείως διαφορετικά δένδρα απόφασης. Το πρόβλημα αυτό επιλύεται με κατάλληλους αλγορίθμους bagging and boosting.
- Υπάρχεις προκατάληψη (bias) σε ό,τι αφορά μη ισορροπημένα σύνολα δεδομένων (imbalanced datasets). Για το λόγο αυτό ο αναλυτής πρέπει να έχει μεριμνήσει προηγουμένως να την εξισορρόπηση των δεδομένων.

Στο παρακάτω παράδειγμα φαίνεται η εφαρμογή του αλγορίθμου στη διαδικασία λήψης της απόφασης στο ερώτημα εάν θα παίξουμε γκολφ ανάλογα με την πρόβλεψη του καιρού η οποία βασίζεται σε ένα σύνολο χαρακτηριστικών όπως η θερμοκρασία, η υγρασία και ο άνεμος.

Σχ. 5.5.2.3 Παράδειγμα αλγορίθμου Decision Tree

Πηγή: ("Decision Tree - Classification")

Στο παράδειγμα φαίνεται ότι το προβλεπτικό μοντέλο θέτει ως κόμβο-ρίζα την πρόβλεψη του καιρού και διαμερίζοντας με βάση τις τιμές των χαρακτηριστικών (temperature, humidity, wind) καταλήγει στα φύλλα τα οποία αποτελούν την τελική απόφαση, αν δηλαδή θα παίξουμε γκολφ ή όχι.

5.5.3 Random Forest Classifier

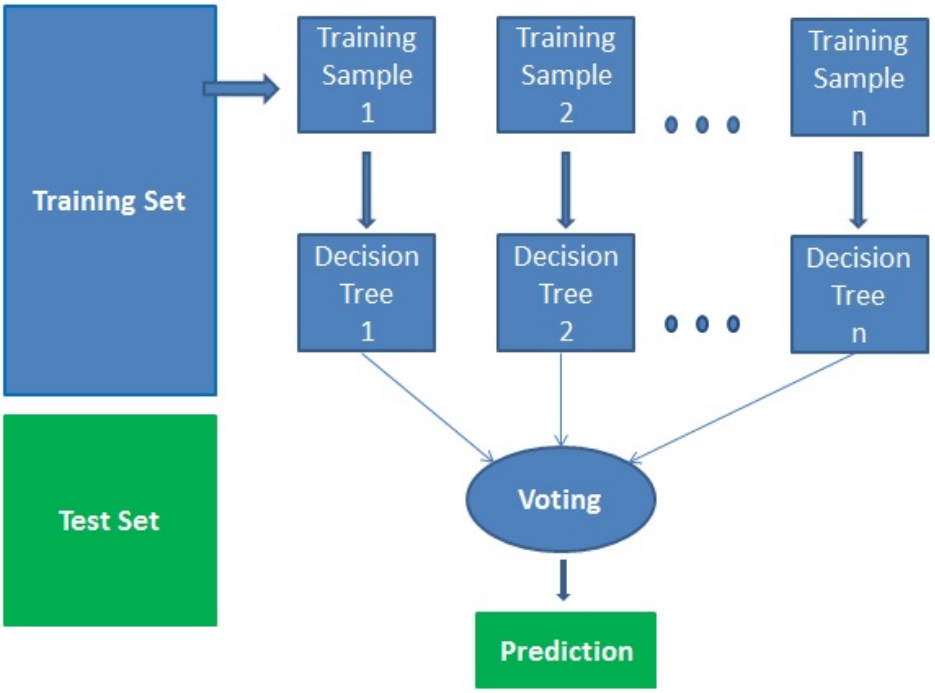
Ο ταξινομητής Random Forest είναι ένας αλγόριθμος ο οποίος διεξάγει προβλέψεις βασιζόμενος στο συνδυασμό διαφορετικών δέντρων αποφάσεων. Πρακτικά, εφαρμόζει πολλούς ταξινομητές δέντρων αποφάσεων σε υποσύνολα του συνόλου δεδομένων σχηματίζοντας το λεγόμενο <<δάσος>> (Forest). Επιπλέον, κάθε δένδρο στο δάσος δημιουργείται βασιζόμενο σε ένα τυχαίο σύνολο χαρακτηριστικών. Τελικά, η διαδικασία δημιουργίας και εφαρμογής όλων των διαφόρων δέντρων αποφάσεων δίνει τη δυνατότητα εύρεσης του καλύτερου συνόλου χαρακτηριστικών απ' όλα τα τυχαία σύνολα που χρησιμοποιήθηκαν. Ο αλγόριθμος Random Forest είναι αυτή τη

στιγμή ένας από τους καλύτερους και αποδοτικότερους αλγορίθμους σε προβλήματα ταξινόμησης.

Ο αλγόριθμος Random Forest είναι ένας αλγόριθμος επιβλεπόμενης μάθησης ο οποίος χρησιμοποιείται τόσο για προβλήματα ταξινόμησης όσο και για προβλήματα παλινδρόμησης. Βασίζεται στη λογική πως όσο περισσότερα δέντρα έχει το δάσος τόσο πιο ισχυρό είναι από πλευράς ικανότητας πρόβλεψης. Αφού δημιουργήσει ένα δέντρο απόφασης για κάθε τυχαίο υποσύνολο δεδομένων, εντοπίζει τη βέλτιστη λύση για κάθε ένα από τα δέντρα που προέκυψαν και τελικά επιλέγει τη βέλτιστη λύση από όλες τις παραπάνω λύσεις ως λύση του προβλήματος με τη μέθοδο της ψηφοφορίας. Σε προβλήματα ταξινόμησης, κάθε δέντρο ψηφίζει και η περισσότερο δημοφιλής κλάση επιλέγεται ως λύση του προβλήματος. Αντίθετα, σε προβλήματα παλινδρόμησης, η τελική λύση θεωρείται ως ο μέσος όρος όλων των λύσεων των επιμέρους δέντρων αποφάσεων.Ο αλγόριθμος παρέχει, επίσης, μια πολύ σαφή εικόνα της σημαντικότητας των διάφορων χαρακτηριστικών που υπάρχουν στα δεδομένα.

Ο συγκεκριμένος αλγόριθμος βρίσκει πληθώρα εφαρμογών όπως σε συστήματα προτάσεων (recommendation systems), ταξινόμηση εικόνων (image classification) και επιλογή χαρακτηριστικών (feature selection). Μπορεί, για παράδειγμα, να χρησιμοποιηθεί για την ταξινόμηση των τυπικών στις πληρωμές δανειοληπτών, τον εντοπισμό παράνομων δραστηριοτήτων και στην πρόβλεψη ασθενειών. Βασίζεται, κατά κύριο λόγο, στον αλγόριθμο Boruta ο οποίος εντοπίζει τα σημαντικά χαρακτηριστικά σε ένα σύνολο δεδομένων.

Σχηματικά, η διαδικασία του ταξινομητή Random Forest φαίνεται παρακάτω:



Σχ. 5.5.3.1 Διαδικασία ταξινομητή Random Forest

Πηγή: ("Understanding Random Forests Classifiers in Python")

Πιο συγκεκριμένα, η διαδικασία που εκτελεί ο ταξινομητής Random Forest είναι η εξής αποτελείται από τα εξής βήματα:

- Επιλογή τυχαίων δειγμάτων από το σύνολο δεδομένων
- Δημιουργία δέντρων απόφασης για κάθε ένα από τα δείγματα ξεχωριστά και πρόβλεψη του αποτελέσματος για κάθε δέντρο
- Διεξαγωγή ψηφοφορίας για κάθε αποτέλεσμα
- Επιλογή της τελικής λύσης με βάση τις περισσότερες ψήφους

Ο αλγόριθμος Random Forest είναι ακόμα ένα όπλο στη φαρέτρα της βιβλιοθήκης Scikit-Learn της Python και χρησιμοποιείται πολλές φορές επίσης και για επιλογή χαρακτηριστικών (Feature Selection). Η βιβλιοθήκη Scikit-Learn μαζί με τον αλγόριθμο παρέχει και μια επιπλέον μεταβλητή η οποία δίνει πληροφορίες για τη σημαντικότητα και τη συνεισφορά κάθε χαρακτηριστικού στην τελική πρόβλεψη. Με τον τρόπο αυτό βαθμολογείται η σχετικότητα των χαρακτηριστικών, ενώ οι τιμές της βαθμολόγησης τελικά έχουν άθροισμα 1 και βοηθούν τον αναλυτή να επιλέξει ποια χαρακτηριστικά θα διατηρήσει και ποια θα απορρίψει ως μη σημαντικά.

Πλεονεκτήματα αλγορίθμου:

- Θεωρείται ένας από τους πιο ισχυρούς και ακριβείς αλγορίθμους λόγω της ικανότητας να χρησιμοποιεί μεγάλο αριθμό δέντρων αποφάσεων.
- Δεν αντιμετωπίζει πρόβλημα σε περιπτώσεις overfitting καθώς λαμβάνει το μέσο όρο των προβλέψεων, πράγμα το οποίο καταργεί τις όποιες προκαταλήψεις (biases).
- Χρησιμοποιείται σε προβλήματα ταξινόμησης και παλινδρόμησης.
- Διαχειρίζεται τιμές που λείπουν με δύο τρόπους: ο ένας τρόπος είναι να χρησιμοποιήσει τις μέσες τιμές για την αντικατάσταση συνεχών μεταβλητών και ο άλλος τρόπος είναι η χρήση του σταθμισμένου μέσου όρου για την αντικατάσταση των τιμών που λείπουν.
- Δίνει πληροφορίες για τη σημαντικότητα των χαρακτηριστικών οι οποίες βοηθούν στην επιλογή χαρακτηριστικών που συμβάλλουν με τον καλύτερο τρόπο στον ταξινομητή.

Μειονεκτήματα αλγορίθμου:

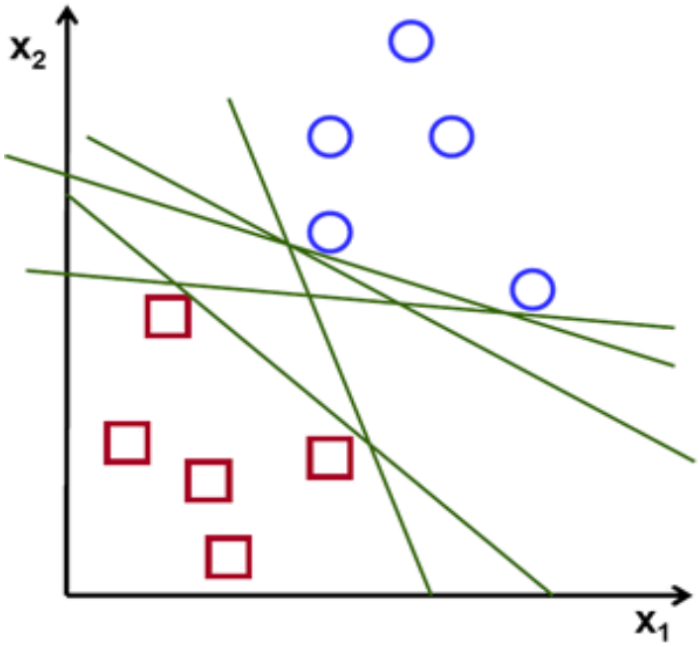
- Ο αλγόριθμος είναι αρκετά αργός στις προβλέψεις καθώς απαιτείται η δημιουργία πολλών δένδρων αποφάσεων. Έπειτα, κάθε δέντρο οφείλει να κάνει μια πρόβλεψη και μετά την πρόβλεψη να λάβει χώρα η ψηφοφορία για την τελική λύση. Οι διαδικασίες αυτές απαιτούν χρόνο.
- Ο αλγόριθμος είναι περίπλοκος στην κατανόηση και την ερμηνεία συγκριτικά με τα δένδρα αποφάσεων στα οποία η λήψη της απόφασης είναι ορατή στο διάγραμμα ακολουθώντας τις διαδρομές κλάδων και κόμβων.

5.5.4 Support Vector Machine (SVM) Classifier

Ο αλγόριθμος Support Vector Machine (SVM) προτιμάται έντονα από τους επιστήμονες δεδομένων καθώς χαρακτηρίζεται από μεγάλη ακρίβεια και χρησιμοποιείται τόσο για προβλήματα ταξινόμησης όσο και για προβλήματα παλινδρόμησης. Ωστόσο, περισσότερο χρησιμοποιείται στην ταξινόμηση.

Στόχος του αλγορίθμου είναι η καθιέρωση ενός τρόπου με τον οποίο τα N χαρακτηριστικά ταξινομούνται με ακρίβεια σε ένα χώρο N διαστάσεων. Όπως είναι φανερό, τα χαρακτηριστικά αναπαρίστανται με σημεία στο διάγραμμα των N διαστάσεων.

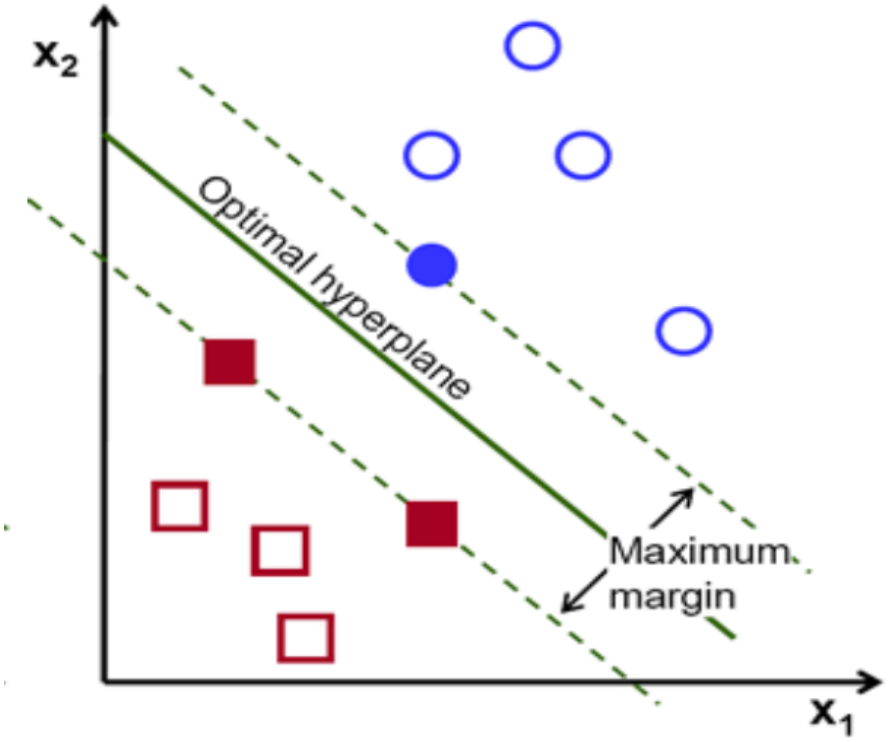
Στο παρακάτω σχήμα βλέπουμε την προσπάθεια ταξινόμησης των μπλε κύκλων και των κόκκινων τετραγώνων σε δύο διαφορετικές κλάσεις με την επιλογή του κατάλληλου διαχωριστικού ορίου.



Σχ. 5.5.4.1 Πιθανά όρια σε ταξινόμηση 2 κλάσεων

Πηγή: ("Support Vector Machine — Introduction to Machine Learning Algorithms")

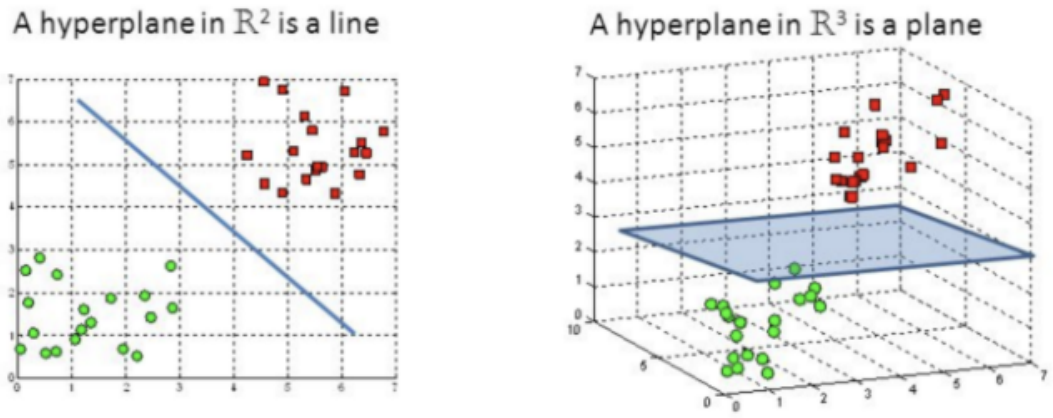
Όπως φαίνεται στο παραπάνω σχήμα, υπάρχουν πολλοί πιθανοί τρόποι να επιλεγούν τα όρια για το διαχωρισμό των κλάσεων. Εντούτοις, στόχος του ταξινομητή είναι η επιλογή του ορίου με τέτοιο τρόπο ώστε να μεγιστοποιείται το περιθώριο μεταξύ των κλάσεων, δηλαδή να μεγιστοποιείται η απόσταση της διαχωριστικής γραμμής από τα πλησιέστερα σε αυτήν σημεία των κλάσεων. Με τον τρόπο αυτό εξασφαλίζεται όλο και μεγαλύτερη πιθανότητα τα μελλοντικά δεδομένα που θα δοθούν στον ταξινομητή να τοποθετηθούν στη σωστή τους κλάση. Η βέλτιστη επιλογή διαχωριστικής φαίνεται στο παρακάτω σχήμα.



Σχ. 5.5.4.2 Επιλογή ορίου με το μεγαλύτερο περιθώριο

Πηγή: ("Support Vector Machine — Introduction to Machine Learning Algorithms")

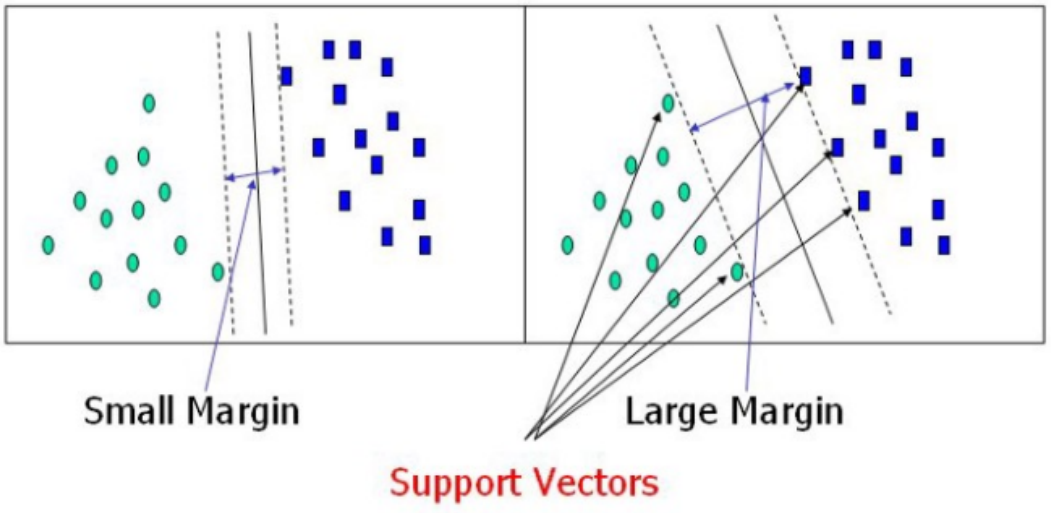
Τα διαχωριστικά σχήματα των κλάσεων αποτελούν ουσιαστικά τα όρια απόφασης τα οποία βοηθούν στην ταξινόμηση των σημείων των δεδομένων. Αν ορισμένα σημεία των δεδομένων βρεθούν σε διαφορετικές πλευρές του διαχωριστικού σχήματος, τότε θα ταξινομηθούν σε διαφορετικές κλάσεις. Επιπλέον, οι διαστάσεις του διαχωριστικού σχήματος εξαρτώνται από το πλήθος των χαρακτηριστικών. Για παράδειγμα, αν το πλήθος των χαρακτηριστικών είναι 2 τότε το διαχωριστικό σχήμα είναι απλά μια μονοδιάστατη γραμμή. Αν το πλήθος των χαρακτηριστικών είναι 3, τότε το διαχωριστικό σχήμα είναι δισδιάστατο. Αναλυτικότερα, τα διαχωριστικά σχήματα ανάλογα με το πλήθος των χαρακτηριστικών φαίνονται στο παρακάτω σχήμα.



Σχ. 5.5.4.3 Διαχωριστικά κλάσεων (hyperplanes) για 2 και 3 χαρακτηριστικά

Πηγή: ("Support Vector Machine — Introduction to Machine Learning Algorithms")

Τα σημαντικότερα σημεία των δεδομένων τα οποία καθορίζουν τη θέση του διαχωριστικού σχήματος και τελικά συνθέτουν το μοντέλο SVM είναι αυτά που είναι πλησιέστερα στα όρια. Η θέση τους καθορίζει την επιλογή του σχήματος ώστε να επιτευχθεί το μέγιστο δυνατό περιθώριο (margin) για την καλύτερη απόδοση του αλγορίθμου και σε μελλοντικά δεδομένα. Τα σημεία αυτά που βρίσκονται πλησιέστερα στα όρια καλούνται διανύσματα υποστήριξης (Support Vectors). Η ύπαρξη και η θέση τους φαίνονται στο ακόλουθο σχήμα.



Σχ. 5.5.4.4 Support Vectors

Πηγή: ("Support Vector Machine — Introduction to Machine Learning Algorithms")

Η προσαρμογή του διαχωριστικού σχήματος ώστε να επιτευχθεί το μέγιστο δυνατό περιθώριο πραγματοποιείται με τη χρήση μιας συνάρτησης κόστους και συνεχείς ενημερώσεις για την κλίση του σχήματος αυτού. Συγκεκριμένα η συνάρτηση κόστους είναι:

$$c(x, y, f(x)) = \left\{ \begin{array}{l} 0, \text{αν } y^* f(x) \geq 1 \\ 1 - y^* f(x), \text{αλλιώς} \end{array} \right\}$$

Η συνάρτηση κόστους παίρνει την τιμή 0 αν η προβλεπόμενη τιμή είναι ίδια με την πραγματική. Διαφορετικά υπολογίζουμε μια τιμή απώλειας στην οποία προσθέτουμε μια παράμετρο κανονικοποίησης ώστε να υπάρχει ισορροπία μεταξύ απώλειας και μεγιστοποίησης περιθωρίου. Συνεπώς, η νέα συνάρτηση κόστους φαίνεται παρακάτω.

$$min_w \lambda \left|\left|w^2\right|\right| + \sum_{i=1}^{n} (1 - y_i < x_i, w >), \text{ όπου ο όρος } w \text{ καθορίζει την κλίση}$$

Για την εύρεση της κλίσης που απαιτείται για την ελαχιστοποίηση της απώλειας χρησιμοποιούμε τις πλευρικές παραγώγους της συνάρτησης ως εξής:

$$\frac{\delta}{\delta w}(\lambda \left|\left|w^2\right|\right|) = 2\lambda w \qquad \text{και} \qquad \frac{\delta}{\delta w}(1 - y_i < x_i, w >) = \left\{ \begin{array}{l} 0 \text{ , αν } y_i < x_i, w > \geq 1 \\ -y_i x_i \text{ , αλλιώς} \end{array} \right\}$$

- Αν το μοντέλο προβλέπει σωστά στην κλάση του σημείου τότε ενημερώνεται η τιμή της κλίσης του σχήματος ως εξής:

  $$w = w - a * (2\lambda w)$$

- Αν το μοντέλο προβλέπει λάθος την κλάση του σημείου, τότε στην ενημέρωση της κλίσης συμπεριλαμβάνουμε την απώλεια στην παράμετρο κανονικοποίησης:

  $$w = w - a * (y_i x_i - 2\lambda w)$$

Πλεονεκτήματα αλγορίθμου:

- Λειτουργεί αρκετά καλά με ένα σαφές περιθώριο διαχωρισμού.
- Είναι αρκετά αποδοτικός σε χώρους πολλών διαστάσεων.
- Είναι αποδοτικός σε περιπτώσεις όπου ο αριθμός των διαστάσεων είναι μεγαλύτερος από τον αριθμό των δειγμάτων
- Χρησιμοποιεί ένα υποσύνολο του συνόλου των σημείων εκπαίδευσης (support vectors) στη συνάρτηση απόφασης και συνεπώς είναι αρκετά αποδοτικός σε ζητήματα μνήμης.

Μειονεκτήματα αλγορίθμου:

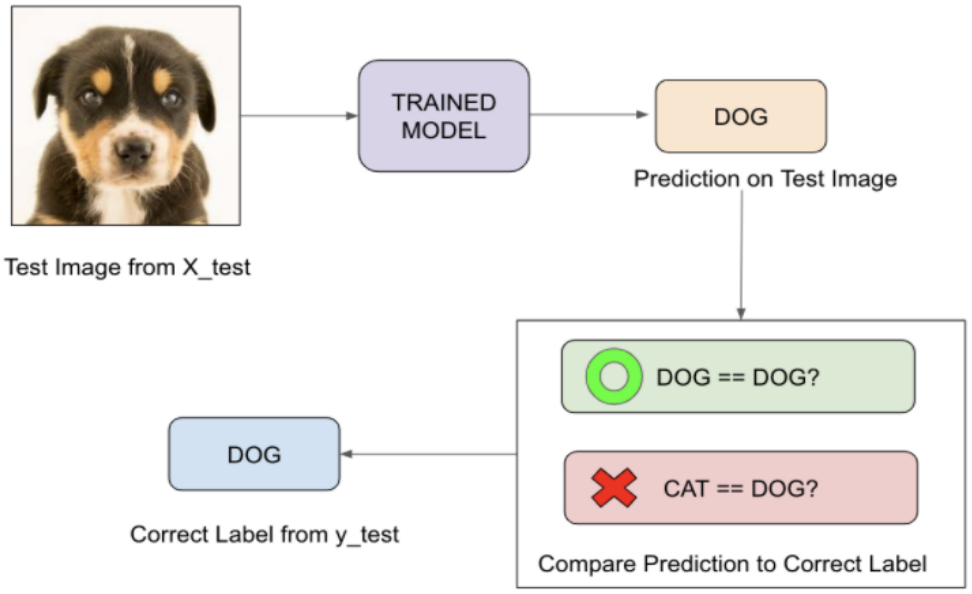
- Δε λειτουργεί αποδοτικά σε περίπτωση μεγάλου συνόλου δεδομένων καθώς ο χρόνος εκπαίδευσης είναι εκτεταμένος.
- Δε λειτουργεί αποδοτικά όταν υπάρχει αρκετός θόρυβος στο σύνολο δεδομένων, παραδείγματος χάρη, οι κλάσεις να αλληλοεπικαλύπτονται.
- Η μέθοδος SVM δεν υπολογίζει απευθείας τις πιθανότητες των εκτιμήσεων αλλά απαιτείται η χρήση ενός εργαλείου cross validation το οποίο παρέχεται από τη βιβλιοθήκη Scikit-Learn.

5.6 Αξιολόγηση των μοντέλων ταξινόμησης

Κατά τη δημιουργία και τη βελτιστοποίηση ενός μοντέλου ταξινόμησης, είναι σημαντική η ύπαρξη ενός δείκτη που καθορίζει την ακρίβεια της πρόβλεψης σε σχέση με το αναμενόμενο αποτέλεσμα. Η αξιολόγηση, λοιπόν, ενός μοντέλου ταξινόμησης είναι ένα πολύ κρίσιμο μέρος της διαδικασίας κατασκευής ενός αποδοτικού μοντέλου. Για την αξιολόγηση των μοντέλων υπάρχουν ορισμένα μέτρα με συχνότερο το μέτρο της *ακρίβειας (Accuracy)*.

Στην περίπτωση της δυαδικής ταξινόμησης όπου υπάρχουν μόνο δύο κλάσεις η πρόβλεψη του μοντέλου μπορεί να λάβει μόνο δύο τιμές. Επομένως η πρόβλεψη είναι είτε σωστή, δηλαδή η τιμή της πρόβλεψης είναι η πραγματική ετικέτα των δεδομένων ελέγχου (test data), είτε είναι λανθασμένη. Η διαδικασία αξιολόγησης του μοντέλου αποτελείται από ορισμένα βήματα που πραγματοποιούνται σειριακά. Σε προβλήματα ταξινόμησης επιβλεπόμενης μάθησης, όπου είναι γνωστή η κλάση των δεδομένων ελέγχου, αρχικά εκπαιδεύεται το μοντέλο στα δεδομένα εκπαίδευσης (train data) και μετά εφαρμόζονται στο μοντέλο τα δεδομένα ελέγχου. Αφού λάβουμε από το μοντέλο τις προβλέψεις για τα δεδομένα ελέγχου, συγκρίνουμε την πρόβλεψη με την ήδη γνωστή ετικέτα των δεδομένων αυτών.

Ας υποθέσουμε ότι εκπαιδεύουμε ένα μοντέλο να εκτελεί image classification και έπειτα του δίνουμε ως είσοδο την παρακάτω εικόνα σαν δεδομένο ελέγχου. Η διαδικασία της πρόβλεψης και της αξιολόγησης φαίνεται στο ακόλουθο σχήμα:



Σχ. 5.6.1 Σύγκριση πρόβλεψης και πραγματικής ετικέτας

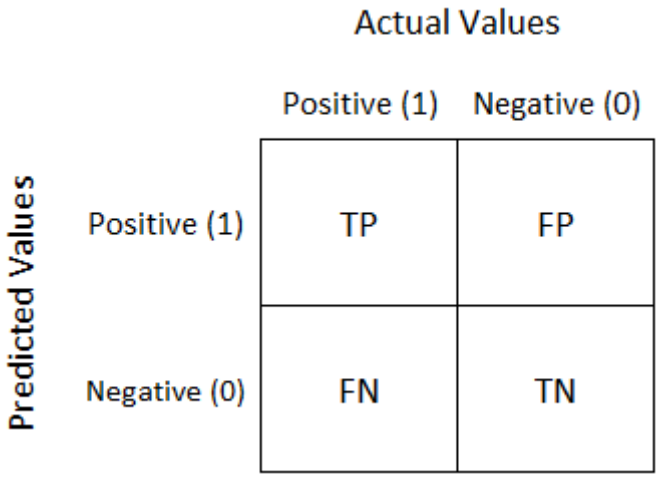
Πηγή: ("More Performance Evaluation Metrics for Classification Problems You Should Know")

Το μοντέλο προβλέπει ότι η εικόνα απεικονίζει ένα σκύλο και στη συνέχεια συγκρίνει την πρόβλεψη με την πραγματική ετικέτα. Αν η πρόβλεψη συμφωνεί με την ετικέτα "Dog" είναι σωστή, ενώ αν η πρόβλεψη είναι "Cat" τότε είναι λανθασμένη. Αν επαναλάβουμε τη διαδικασία σύγκρισης των προβλέψεων για κάθε δεδομένο εισόδου θα προκύψει ένας αριθμός σωστών και ένας αριθμός λανθασμένων προβλέψεων. Η ακρίβεια (Accuracy) ορίζεται ως:

$$Accuracy = \frac{Correct\ Predictions}{Total\ Predictions}$$

Η ακρίβεια έχει το πλεονέκτημα ότι είναι εξαιρετικά απλή στον υπολογισμό και την κατανόηση, όμως, είναι αξιόπιστη μόνο σε ισορροπημένα δεδομένα. Δεν παράγει αξιόπιστο αποτέλεσμα, δηλαδή, σε περίπτωση που το 95% των δεδομένων ανήκουν στη μία κλάση και το 5% των δεδομένων ανήκουν στην άλλη. Για το λόγο αυτό, δημιουργήθηκε η ανάγκη ένταξης περαιτέρω μετρήσεων στη διαδικασία της αξιολόγησης των μοντέλων, όπως η Recall και η Precision.

Προτού ορίσουμε τα μέτρα αξιολόγησης των μοντέλων ταξινόμησης κρίνεται απαραίτητο να ορίσουμε όλα τα πιθανά αποτελέσματα μιας διαδικασίας ταξινόμησης από ένα μοντέλο, τα οποία φαίνονται στον παρακάτω πίνακα (Confusion Matrix).



Σχ. 5.6.3 *Confusion Matrix*

Πηγή: ("Understanding Confusion Matrix")

Οι συμβολισμοί TP, FP, FN και TN αναπαριστούν την πιθανή έκβαση του ταξινομητή. Αναλυτικότερα:

TP: Στην πραγματικότητα είναι positive και στην πρόβλεψη επίσης positive.

πχ Το μοντέλο προβλέπει ότι ένας σκύλος είναι μαύρος ενώ όντως είναι.

FP: Στην πραγματικότητα είναι negative και στην πρόβλεψη είναι positive.

πχ Το μοντέλο προβλέπει ότι ένας σκύλος είναι λευκός ενώ δεν είναι.

FN: Στην πραγματικότητα είναι positive και στην πρόβλεψη είναι negative.

πχ Το μοντέλο προβλέπει ότι ένας σκύλος δεν είναι μαύρος ενώ είναι.

TN: Στην πραγματικότητα είναι negative και στην πρόβλεψη επίσης negative.

πχ Το μοντέλο προβλέπει ότι ένας σκύλος δεν είναι λευκός ενώ όντως δεν είναι.

Τα τέσσερα δημοφιλέστερα μέτρα αξιολόγησης της απόδοσης ενός ταξινομητή ορίζονται ως ακολούθως:

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

$$Recall = \frac{TP}{TP+FN}$$

$$Precision = \frac{TP}{TP+FP}$$

$$F_1 = \frac{2}{\frac{1}{RECALL}+\frac{1}{PRECISION}} (F_1\text{Score})$$

6. ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

6.1 Περιγραφή του προβλήματος

Η παρούσα εργασία βασίζεται στην έρευνα που έχει γίνει σχετικά με την εφαρμογή μερικών από τους δημοφιλέστερους αλγόριθμους ταξινόμησης σε ένα σύνολο κειμένων τα οποία αποτελούν βιογραφικά σημειώματα υποψηφίων που ανήκουν σε διαφορετικά επαγγέλματα και ειδικότητες. Οι ταξινομητές επιχειρούν να προβλέψουν την κατηγορία στην οποία βρίσκεται κάθε βιογραφικό σημείωμα με τη μέγιστη δυνατή ακρίβεια. Οι ταξινομήσεις οι οποίες διεξήχθησαν είναι κατά πλειοψηφία δυαδικές ενώ οι κατηγορίες για το αρχικό πρόβλημα αφορούν αρχικά δύο επαγγέλματα (Engineering, Information Technology). Το δεύτερο πρόβλημα το οποίο εξετάστηκε είναι η κατηγοριοποίηση των βιογραφικών σημειωμάτων ανάλογα με την καταλληλότητα του υποψηφίου να εργαστεί σε μία θέση εργασίας συναφή με την ειδικότητά του. Από τη φύση των προβλημάτων και των δεδομένων είναι φανερό ότι τα μοντέλα που δημιουργήθηκαν χρησιμοποιούν αλγόριθμους επιβλεπόμενης μάθησης. Επιπλέον, έγινε έρευνα στην χρήση τεχνικής μη επιβλεπόμενης μάθησης στο σύνολο των δεδομένων με την παραδοχή ότι δε χρησιμοποιήθηκε η γνώση των ήδη υπαρχουσών ετικετών στα δεδομένα αυτά.

Αναλυτικότερα, στο πλαίσιο του πειράματος της παρούσας εργασίας διενεργήθηκαν τα εξής:

- Εφαρμογή τεχνικών ανάλυσης και εξόρυξης κειμένου για την προετοιμασία των δεδομένων, δηλαδή την απομάκρυνση του θορύβου από τα δεδομένα και την προεπεξεργασία τους.
- Δημιουργία μοντέλων πρόβλεψης με βάση αλγορίθμους επιβλεπόμενης και μη επιβλεπόμενης μάθησης κατά περίπτωση και ξεχωριστά.
- Αξιολόγηση των μοντέλων πρόβλεψης με βάση ορισμένα μέτρα αξιολόγησης.

Για την αξιολόγηση των μοντέλων χρησιμοποιήθηκε το σύνολο δεδομένων **resume_dataset.csv** το οποίο αντλήθηκε από τον ιστότοπο **kaggle.com** και περιλαμβάνει 1219 γραμμές και τρεις στήλες εκ των οποίων η 1η αποτελεί το ID του αντίστοιχου βιογραφικού, η 2η την κατηγορία (επάγγελμα) και η 3η το κείμενο του βιογραφικού. Οι 25 κατηγορίες των βιογραφικών που περιέχονται στο dataset και το πλήθος των βιογραφικών που υπάρχουν σε κάθε κατηγορία φαίνονται στον ακόλουθο πίνακα:

Ειδικότητα	**Αριθμός εγγράφων**

Engineering	121
Information Technology	104
Education	102
Health & Fitness	77
Management	74
Accountant	67
Finance	66
Sales	61
Advocate	61
Digital Media	54
Designing	51
Banking	48
Business Development	44
Arts	43
HR	41
Building & Construction	29
Automobile	27
Consultant	26
BPO	25
Agricultural	24
Food & Beverages	22
Apparel	14
Aviation	13
Public Relations	13
Architects	12

Τα πειράματα έλαβαν χώρα στο προγραμματιστικό περιβάλλον του Jupyter Notebook και με τη βοήθεια της Python 3.

6.2 Προεπεξεργασία των δεδομένων

Αρχικά, αφού εισήχθησαν οι απαραίτητες βιβλιοθήκες (Nltk, pandas, matplotlib, numpy, scipy, scikit-learn) για την επεξεργασία και τις οπτικοποιήσεις των δεδομένων, αποθηκεύτηκε το dataset με τη μορφή dataframe. Στο σημείο αυτό και πριν ξεκινήσει η διαδικασία κατασκευής των μοντέλων απαιτείται ο καθαρισμός των δεδομένων από το θόρυβο και η προεπεξεργασία τους. Ο καθαρισμός των δεδομένων στο παρόν πείραμα περιλαμβάνει την απαλοιφή των σημείων στίξης και των χαρακτήρων ASCII, την αφαίρεση των stopwords (a, an, the, and, of, in, up, while κλπ), η μετατροπή των γραμμάτων από κεφαλαία σε πεζά και γενικότερα όλη η διαδικασία προεπεξεργασίας των δεδομένων που αναφέρθηκε στο Κεφ.4. Συγκεκριμένα, ο κώδικας για την προεπεξεργασία των δεδομένων φαίνεται παρακάτω.

In [4]:
```
def clean_text(doc: str):                                   #Χωρισμός λέξεων και παραγράφων
    doc = doc.replace('\\n', '\n')
    return doc

def to_ascii(doc: str):                                     #Αντικατάσταση χαρακτήρων ASCII με κενό
    for i in range(0,256):
        doc = doc.replace('\\{}'.format(hex(i)[-3:]), ' ')
        doc = doc.replace('\\x0c', ' ')
        doc = doc.replace('/', ' ')
    return doc

def remove_punctuation(x):                                  #Αφαίρεση σημείων στίξης
    table = str.maketrans({key: ' ' for key in string.punctuation})
    return x.translate(table)
```

In [5]:
```
data=pd.read_csv(r"C:\Users\User\Untitled Folder 2\resume_dataset.csv.zip")

data['Resume'] = [clean_text(entry) for entry in data['Resume'].values]
data['Resume'] = [to_ascii(entry) for entry in data['Resume'].values]
data['Resume'] = [remove_punctuation(entry) for entry in data['Resume'].values]
data['Resume'] = [entry.lower() for entry in data['Resume'].values]
```

In [8]:
```
from nltk import word_tokenize
def tokenizing(x):
    tmp = nltk.word_tokenize(x)
    return tmp
data['Resume'] = [tokenizing(entry) for entry in data['Resume'].values]

nltk.download('stopwords') # κατεβάζουμε ένα αρχείο που έχει stopwords στα αγγλικά
from nltk.corpus import stopwords

def remove_stopwords(text_tokens: list):
    tokens_without_sw = [word for word in text_tokens if not word in stopwords.words('english')]
    return tokens_without_sw
for i in range(0,1219):
    data.Resume[i] = remove_stopwords(data.Resume[i])
```

In [9]:
```
nltk.download('wordnet') # απαραίτητα download για τους stemmer/lemmatizer
nltk.download('rslp')

from nltk.stem import WordNetLemmatizer
wordnet_lemmatizer = WordNetLemmatizer()

from nltk.stem.porter import PorterStemmer
porter_stemmer = PorterStemmer()

def stemming(doc: list):
    stem_words = [porter_stemmer.stem(word) for word in doc]
    return stem_words

def lemmatization(doc: list):
    lem_words = [wordnet_lemmatizer.lemmatize(word) for word in doc]
    return lem_words

for i in range(0,1219):
    data.Resume[i] = stemming(data.Resume[i])
    data.Resume[i] = lemmatization(data.Resume[i])
```

In [10]:
```
def remove_numbers(doc: list):
    no_integers = [x for x in doc if not (x.isdigit()
                                          or x[0] == '-' and x[1:].isdigit())]
    return no_integers
data['Resume'] = [remove_numbers(entry) for entry in data['Resume'].values]
```

Οι συναρτήσεις που δημιουργήθηκαν εκτελούν τις διαδικασίες:

- Χωρισμός παραγράφων
- Αφαίρεση χαρακτήρων ASCII
- Αφαίρεση σημείων στίξης
- Μετατροπή γραμμάτων από κεφαλαία σε πεζά
- Tokenization
- Αφαίρεση Stopwords
- Lemmatization
- Stemming

Ενδεικτικά στην παρακάτω εικόνα φαίνεται ένα βιογραφικό από τα δεδομένα πριν την προεπεξεργασία και μετά από αυτή.

```
b'Name Surname\nAddress\nMobile No/Email\nPERSONAL PROFILE\nI am a self motivated individual who has a confident approach to pe
ople. I communicate\nwell with all levels of personnel and feel that I have a good listening ability which allows\nme to resolv
e problems quickly.\nI am enthusiastic about my role and enjoy working in HR, I like the fast paced\nenvironment which is alway
s changing and I like to adapt to these changes quickly\nallowing others to also adapt quickly.\nI am organized by nature and l
ike to ensure that I am up to date with my work. I enjoy\nnew challenges and I am always keen to learn new skills.\nEMPLOYMENT
HISTORY\nDate to Date or To Date \xe2\x80\x93 HR Consultant \xe2\x80\x93 Where?\nIn my role as HR Consultant, I visit clients a
nd provide HR advice and help resolve\nissues. My responsibilities include:\n\xef\x82\xb7Provide employment law advice\n\xef\x8
2\xb7Help with writing and issuing contracts of employment and employee handbooks\n\xef\x82\xb7Advise on maternity/paternity ri
ghts\n\xef\x82\xb7Help with payroll and holidays, sickness etc\n\xef\x82\xb7Building up relationships with new and existing cli
ents\n\xef\x82\xb7Training new staff when coming into a business and Managers to deal with their staff\nwith regards to personn
el\n\xef\x82\xb7Conduct disciplinary hearings or appeals as an intermediary person\n\xef\x82\xb7Devise staff benefits and incen
tives\n\xef\x82\xb7Learning and Development opportunities for companies and their staff.\nQUALIFICATIONS\nUniversity, College,
School \xe2\x80\x93 For all include titles/subjects and qualifications.\nSKILLS AND ABILITIES\nComputer skills \xe2\x80\x93 MS
Office, Excel??? Any specific HR databases or record keeping\nsoftware? CIPD qualification or working towards?\nHOBBIES & INTER
ESTS\nWhat do you like to do outside of work?\nREFERENCES\nAvailable on request.'
```

Σχ.6.1 Βιογραφικό με ID=1 του dataset σε μη επεξεργασμένη μορφή

```
['b', 'name', 'surnam', 'address', 'mobil', 'email', 'person', 'profil', 'self', 'motiv', 'individu', 'confid', 'approach', 'pe
opl', 'commun', 'well', 'level', 'personnel', 'feel', 'good', 'listen', 'abil', 'allow', 'resolv', 'problem', 'quickli', 'enthu
siast', 'role', 'enjoy', 'work', 'hr', 'like', 'fast', 'pace', 'environ', 'alway', 'chang', 'like', 'adapt', 'chang', 'quickl
i', 'allow', 'other', 'also', 'adapt', 'quickli', 'organ', 'natur', 'like', 'ensur', 'date', 'work', 'enjoy', 'new', 'challen
g', 'alway', 'keen', 'learn', 'new', 'skill', 'employ', 'histori', 'date', 'date', 'date', 'hr', 'consult', 'role', 'hr', 'cons
ult', 'visit', 'client', 'provid', 'hr', 'advic', 'help', 'resolv', 'issu', 'respons', 'includ', 'provid', 'employ', 'law', 'ad
vic', 'help', 'write', 'issu', 'contract', 'employ', 'employe', 'handbook', 'advis', 'matern', 'patern', 'right', 'help', 'payr
ol', 'holiday', 'sick', 'etc', 'build', 'relationship', 'new', 'exist', 'client', 'train', 'new', 'staff', 'come', 'busi', 'man
ag', 'deal', 'staff', 'regard', 'personnel', 'conduct', 'disciplinari', 'hear', 'appeal', 'intermediari', 'person', 'devi', 'st
aff', 'benefit', 'incent', 'learn', 'develop', 'opportun', 'compani', 'staff', 'qualif', 'univers', 'colleg', 'school', 'inclu
d', 'titl', 'subject', 'qualif', 'skill', 'abil', 'comput', 'skill', 'm', 'offic', 'excel', 'specif', 'hr', 'databas', 'recor
d', 'keep', 'softwar', 'cipd', 'qualif', 'work', 'toward', 'hobbi', 'interest', 'like', 'outsid', 'work', 'refer', 'avail', 're
quest']
```

Σχ.6.2 Βιογραφικό με ID=1 του dataset σε επεξεργασμένη μορφή

Βλέπουμε ότι μετά την επεξεργασία, το κείμενο έχει απαλλαγεί από το θόρυβο και ο όγκος του είναι πολύ μικρότερος. Περιέχει ένα σύνολο λέξεων οι οποίες αποτελούν τα χαρακτηριστικά τα οποία θα χρησιμοποιηθούν μετέπειτα από τους ταξινομητές.

6.3 Εφαρμογή αλγορίθμων μάθησης στα δεδομένα

- K-means Clustering

Στην περίπτωση εφαρμογής του αλγορίθμου K-means, ο οποίος είναι αλγόριθμος μη επιβλεπόμενης μάθησης, από τα προεπεξεργασμένα δεδομένα εξήχθησαν τα χαρακτηριστικά με βάση τη μέθοδο tf-idf. Οι προβλέψεις του Clustering αποτιμώνται μέσω της μετρικής silhouette score για τις διάφορες τιμές του k. Το βέλτιστο πλήθος κατηγοριών k λαμβάνεται για τη μέγιστη τιμή της μέτρησης sihlouette score.

In [26]:
```
def remove_digits_from_string(string: str):
    res = ''.join([i for i in string if not i.isdigit()])
    return res
def remove_single_letters(string: str):
    import re
    res = re.sub(r"\b[a-zA-Z]\b", "", string)
    return res
import re
def stemming_tokenizer(str_input):
    words = re.sub(r"[^A-Za-z0-9\-]", " ", str_input).lower().split()
    words = [porter_stemmer.stem(word) for word in words]
    return words
```

In [35]:
```
data_to_cluster=pd.read_csv(r"C:\Users\User\Untitled Folder 2\resume_dataset.csv.zip")
def clean_text(doc: str):                                    #Χωρισμός λέξεων και παραγράφων
    doc = doc.replace('\\n', '\n')
    return doc

def to_ascii(doc: str):                                      #Αντικατάσταση χαρακτήρων ASCII με κενό εφόσον δε
    for i in range(0,256):
        doc = doc.replace('\\{}'.format(hex(i)[-3:]), ' ')
        doc = doc.replace('\\x0c', ' ')
        doc = doc.replace('/', ' ')
    return doc
data_to_cluster['Resume'] = [clean_text(entry) for entry in data_to_cluster['Resume'].values]
data_to_cluster['Resume'] = [to_ascii(entry) for entry in data_to_cluster['Resume'].values]
data_to_cluster['Resume'] = [remove_punctuation(entry) for entry in data_to_cluster['Resume'].values]
data_to_cluster['Resume'] = [entry.lower() for entry in data_to_cluster['Resume'].values]
data_to_cluster['Resume'] = [remove_digits_from_string(entry) for entry in data_to_cluster['Resume'].values]
data_to_cluster['Resume'] = [remove_single_letters(entry) for entry in data_to_cluster['Resume'].values]
data_clust=[]
for i in range(0,1219):
    data_clust.append(data_to_cluster.Resume[i])
```

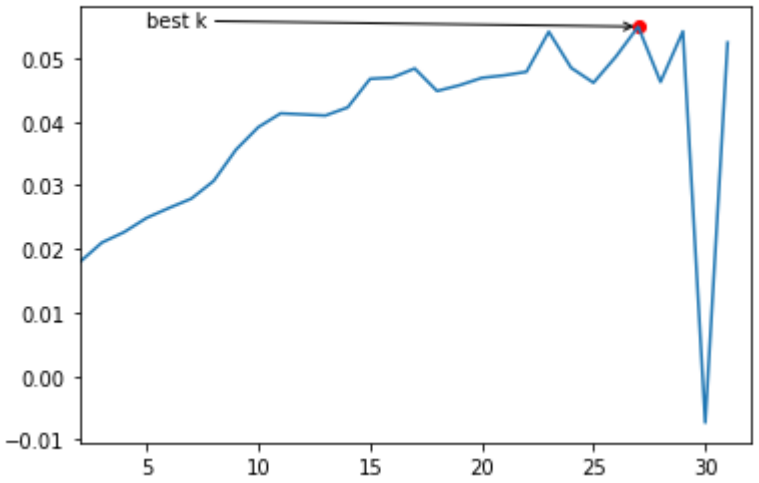
In [36]:
```
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(max_df=700, min_df=15, stop_words='english', tokenizer=stemming_tokenizer)

from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
tf_idf_array = vectorizer.fit_transform(data_clust).toarray()
```

In [33]:
```
silhouette_scores = []
vectorizer.get_feature_names()
for k in range(2, 32):
    print(k)
    km = KMeans(k)
    preds = km.fit_predict(tf_idf_array)
    silhouette_scores.append(silhouette_score(tf_idf_array, preds))
```

In [34]:
```
plt.plot(range(2, 32), silhouette_scores)
best_k = np.argmax(silhouette_scores) + 2 # +2 γιατί ξεκινάμε το range() από k=2 και όχι από 0 που ξεκινάει η αρίθμηση της λίστα
plt.scatter(best_k, silhouette_scores[best_k-2], color='r') # για τον ίδιο λόγο το καλύτερο k είναι αυτό 2 θέσεις παρακάτω από τ
plt.xlim([2,32])
plt.annotate("best k", xy=(best_k, silhouette_scores[best_k-2]), xytext=(5, silhouette_scores[best_k-2]),arrowprops=dict(arrowst
print('Maximum average silhouette score for k =', best_k)
```

Το αποτέλεσμα του αλγορίθμου K-means φαίνεται στο ακόλουθο διάγραμμα:

```
Maximum average silhouette score for k = 27
```



Επομένως βλέπουμε ότι ο k-means κατηγοριοποιεί τα δεδομένα σε 27 κλάσεις κατά βέλτιστο τρόπο, πράγμα το οποίο προσεγγίζει την πραγματικότητα αφού το dataset είναι διαιρεμένο σε 25 κλάσεις-επαγγέλματα.

Στην προσπάθεια ανάκτησης των κέντρων των κλάσεων τα οποία προβλέφθηκαν από τον K-means προέκυψαν τα εξής αποτελέσματα:

In [41]:
```
km = KMeans(best_k)
km.fit(tf_idf_array)
```

Out[41]:
```
KMeans(n_clusters=27)
```

In [42]:
```
terms = vectorizer.get_feature_names()
order_centroids = km.cluster_centers_.argsort()[:, ::-1]
for i in range(best_k):
    out = "Cluster %d:" % i
    for ind in order_centroids[i, :30]:
        out += ' %s' % terms[ind]
    print(out)
```

```
Cluster 0: xx nebraska student engin francisco school san ne middl relat teacher md tax organ program comput illinoi administr
edu design electr lincoln grade resum project research scienc activ fall taught
Cluster 1: school student resum theatr plant career nutrit respons train research anim farm colleg associ knowledg agricultur c
ompani program scienc use offic organ excel inform date job object project comput qualif
Cluster 2: custom sale retail sydney purchas excel ca suppli stock product abil store inventori career columbu client strong bu
si cash beach respons good high request depart merchandis order abl kitchen display
Cluster 3: law legal court bar attorney school draft case state lawyer associ research intern florida litig public contract inc
lud client crimin member student committe offic commerci cours employ busi summer firm
Cluster 4: financi financ oper busi plan account million capit execut market process profit corpor analysi debt compani report
divis strateg sale york improv cost client project fund perform bank senior tax
Cluster 5: patient medic nurs care clinic health procedur hospit treatment center diagnost room imag ray physician mental exam
provid perform ny healthcar offic prepar examin abil doctor certif emerg diseas nv
Cluster 6: construct project engin site safeti civil draw subcontractor design cad estim build cost contractor structur materi
inspect schedul supervisor concret plan supervis contract ny qualiti control equip plant includ bid
Cluster 7: busi test project data requir analyst softwar process applic user analysi client design function document sql use so
lut consult report technic creat implement support sap ms respons technolog server oracl
Cluster 8: project busi engin process implement client support product plan manufactur design technolog financ contract oper fi
nanci activ program schedul cost budget pmp includ prepar provid execut consult analysi softwar requir
Cluster 9: account financi report prepar tax financ reconcili invoic audit monthli statement payabl bank process payrol busi le
dger compani entri ca analysi client cash forecast budget corpor charter control data receiv
Cluster 10: hr employe human resourc recruit benefit compani train polici compens candid job process payrol program administr r
elat implement salari perform cost consult includ hire report function personnel interview depart busi
Cluster 11: design graphic fashion interior art product adob creativ magazin illustr user brand web client artist concept creat
ux photoshop project includ freelanc materi model print logo www ui student game
Cluster 12: photograph art galleri photographi exhibit museum artist curat photo magazin baltimor contemporari edit korea imag
editor maryland mi light print studio catalog ca textil shoot grant design md center austin
Cluster 13: web java php applic softwar css design mysql android javascript server html flash technolog databas use user site e
ngin websit sql project xml program comput jqueri jee test oracl code
Cluster 14: medicin medic yr doctor surgeri clinic hospit date month patient state diseas treatment citi physician facil colleg
good dr xxxx research resid attend practic health ct diagnost intern york school
Cluster 15: dayjob cv thi copyright person pleas ani use templat market coventri abl info birmingham avail busi account client
custom compani download permiss project howev welcom sale websit help abil pass
Cluster 16: automot paint vehicl engin automobil repair mechan car technician motor fuel ycmail damag detroit auto painter job
bodi design summari test histori qualif phone manufactur model custom surfac michigan failur
Cluster 17: student teacher school elementari teach classroom grade parent special princip program learn baltimor lesson taught
nebraska instruct high curriculum district class il administr plan provid implement state activ associ art


Cluster 18: resum student use list includ job letter employ organ posit thi inform research activ engin cover titl verb mn prog
ram page nj section refer career school relat bullet relev write
Cluster 19: sale market custom product busi respons account client compani repres revenu negoti train promot achiev price key m
illion strategi year gener store analyz increas latin territori growth region relationship includ
Cluster 20: health monash edu student hospit research nurs occup member current care monashecd committe program nov engin caree
r associ inform medic au pharmaci public studi school achiev copi feb rural distinct
Cluster 21: dental patient oral restor treatment dr clinic insur pediatr sponsor care surgeri medic indiana health diagnos prac
tic th india associ procedur xx maryland techniqu boston confer program american hospit md
Cluster 22: bank custom loan client financi branch teller account risk credit transact invest busi market cash product deposit
xxxx knowledg handl ca provid oper financ abil sale administr analyst procedur check
Cluster 23: market media social content digit brand custom campaign strategi busi sale facebook websit onlin increas product tw
itter web compani ny blog design googl revenu execut plan www project intern editor
Cluster 24: engin design mechan softwar comput project student use scienc research technolog program electr test chemic resum p
roduct linux societi gpa java member summer matlab kentucki embed intern civil address model
Cluster 25: flight attend pilot air aviat safeti crew airlin aircraft custom train abil beverag forc space command corpor emerg
food job procedur comfort armi land check medal load bilingu az bell
Cluster 26: custom food restaur chef train sale culinari store hotel ensur menu offic abc staff guest client cook kitchen abil
excel handl supervis prepar oper good cater center busi provid maintain
```

Ιδανικά θα έπρεπε το κέντρο κάθε κλάσης να αναφέρεται σε μία ειδικότητα από τις 25 διαφορετικές. Βλέπουμε ότι η περιγραφή των κέντρων των clusters προσεγγίζει την περιγραφή μιας ειδικότητας.

Ο αλγόριθμος K-means όπως παρατηρήθηκε από την εκτέλεσή του έχει τα εξής χαρακτηριστικά:

- Μεγάλη ταχύτητα εκτέλεσης παρά το μεγάλο πλήθος των δεδομένων
- Ευκολία στην εφαρμογή και στην ερμηνεία
- Ικανοποιητική ακρίβεια σε επίπεδο αποτελεσμάτων

Ωστόσο, μειονεκτεί στο γεγονός ότι πρέπει να αποφασιστεί εκ των προτέρων ο αριθμός k των κλάσεων και επίσης παράγει διαφορετικές προβλέψεις σε πολλαπλές εκτελέσεις.

- 1ο Πείραμα

Όπως αναφέρθηκε προηγουμένως, αρχικά από το σύνολο δεδομένων επιλέχθηκαν οι κατηγορίες των επαγγελμάτων *Engineering* και *Information Technology* οι οποίες περιλαμβάνουν 121 και 104 βιογραφικά αντιστοίχως. Τα δεδομένα αυτά αποθηκεύτηκαν σε ένα νέο dataframe με συνολικά 121+104=225 γραμμές. Στην κλάση Information Technology τοποθετήθηκε η ετικέτα "0", ενώ στην κλάση Engineering τοποθετήθηκε η ετικέτα "1". Οι ετικέτες εισήχθησαν ως τέταρτη στήλη στα δεδομένα.

Στη συνέχεια το νέο dataframe διαιρέθηκε σε σύνολο εκπαίδευσης (X_train) και σύνολο ελέγχου (X_test) τα οποία εφαρμόστηκαν στους ταξινομητές. Αναλυτικότερα, χρησιμοποιώντας τις default παραμέτρους της συνάρτησης train_split_test, ο διαχωρισμός πραγματοποιήθηκε σε ποσοστό 25% για το test set και το υπόλοιπο 75% για το train set. Επομένως από τα 225 βιογραφικά τα 56 (0,25*225) αποτελούν το σύνολο ελέγχου και τα 169 το σύνολο εκπαίδευσης. Συγκεκριμένα, διενεργήθηκαν τα εξής:

In [43]:
```
data=pd.read_csv(r"C:\Users\User\Untitled Folder 2\resume_dataset.csv.zip")

mask = (
        (data['Category'] == 'Engineering') |
        (data['Category'] == 'Information Technology')
)

data=data.loc[mask]
```

In [44]:
```
data['label']=data['Category'].apply(lambda x:0 if x=='Information Technology' else 1)
data
```

Out[44]:

	ID	Category	Resume	label
166	167	Information Technology	b'RESUME\nAJITHA SHENOY .K.B,\nPhD student (Co...	0
167	168	Information Technology	b'Mason\t\r \xc2\xa0Silber\t\r \xc2\xa0\n6595\...	0
168	169	Information Technology	b'Pramod XXXX\nMobile: +91-99********\n\nE-mai...	0
169	170	Information Technology	b"Harry M. Rohrer\n3748 Bee Street\nGrand Rapi...	0
170	171	Information Technology	b"Wilson Kunnan Jose\nSr. Consultant, QA\n\nSu...	0
...	...	...	...	...
989	990	Engineering	b'Sample Resume for Engineering Students (jr/s...	1
990	991	Engineering	b'CHEMICAL ENGINEERING R\xc3\x89SUM\xc3\x89S\n...	1
991	992	Engineering	b'Sample Resume - Chemical Engineer Resume\n\n...	1
992	993	Engineering	b'your resume and try to determine if they hav...	1
993	994	Engineering	b'Thomas J. Smith\n55 Northern Road | Sometown...	1

225 rows × 4 columns

Κατόπιν, στο νέο σύνολο δεδομένων πραγματοποιήθηκε προεπεξεργασία και εξαγωγή γνωρισμάτων.

In [45]:
```
from sklearn.model_selection import train_test_split

data['Resume'] = [clean_text(entry) for entry in data['Resume'].values]
data['Resume'] = [to_ascii(entry) for entry in data['Resume'].values]
data['Resume'] = [remove_punctuation(entry) for entry in data['Resume'].values]
data['Resume'] = [entry.lower() for entry in data['Resume'].values]
data['Resume'] = [remove_digits_from_string(entry) for entry in data['Resume'].values]
data['Resume'] = [remove_single_letters(entry) for entry in data['Resume'].values]

X_train, X_test, y_train, y_test = train_test_split(data['Resume'], data['label'], random_state=1)  #Χωρισμός των δεδομένων σε
```

In [169]:
```
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(strip_accents='ascii', token_pattern=u'(?ui)\\b\\w*[a-z]+\\w*\\b', lowercase=True, stop_words='english', to
X_train_cv = cv.fit_transform(X_train)
X_test_cv = cv.transform(X_test)

len(X_test_cv.toarray())
```

Naive Bayes Classifier

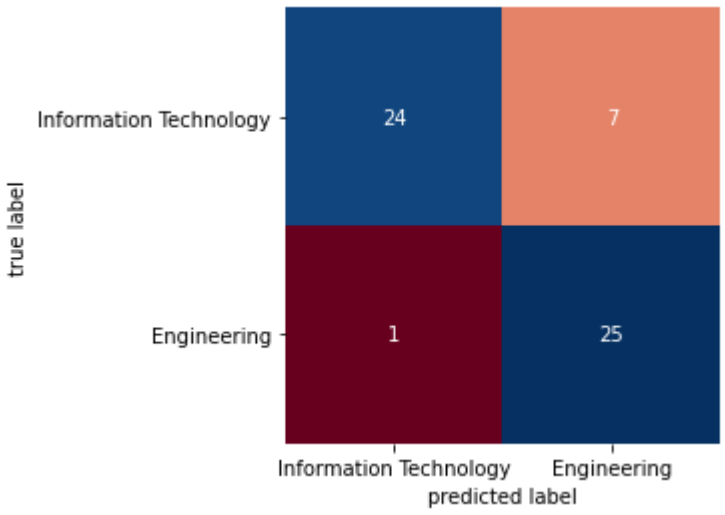
Η πρώτη μέθοδος που εφαρμόστηκε είναι ο ταξινομητής Naive Bayes. Κατόπιν των προβλέψεων που παρήγαγε, αξιολογήθηκαν τα αποτελέσματα με βάση των μέτρων accuracy, precision, recall. Οι προβλέψεις του ταξινομητή μελετήθηκαν τόσο οπτικά με βάση τον confusion matrix όσο και ποσοτικά με τα παραπάνω μέτρα αξιολόγησης.

```
In [30]: from sklearn.naive_bayes import MultinomialNB
         naive_bayes = MultinomialNB()
         naive_bayes.fit(X_train_cv, y_train)
         predictions = naive_bayes.predict(X_test_cv)
```

```
In [31]: from sklearn.metrics import accuracy_score, precision_score, recall_score
         print('Accuracy score:', accuracy_score(y_test, predictions))
         print('Precision score:', precision_score(y_test, predictions))
         print('Recall score:', recall_score(y_test, predictions))
```

```
Accuracy score: 0.8596491228070176
Precision score: 0.78125
Recall score: 0.9615384615384616
```

```
In [32]: from sklearn.metrics import confusion_matrix
         import matplotlib.pyplot as plt
         import seaborn as sns
         cm = confusion_matrix(y_test, predictions)
         print(cm)
         sns.heatmap(cm, square=True, annot=True, cmap='RdBu', cbar=False,
         xticklabels=['Information Technology', 'Engineering'], yticklabels=['Information Technology', 'Engineering'])
         plt.xlabel('predicted label')
         plt.ylabel('true label')
```



Στα αποτελέσματα βλέπουμε τις τιμές των μετρήσεων accuracy, precision, recall οι οποίες υπολογίστηκαν για το test set σύμφωνα με το πλήθος των TP, TN, FP, FN προβλέψεων που παρήγαγε ο ταξινομητής Naive Bayes. Είναι φανερό ότι επιτεύχθηκε ένα ικανοποιητικό ποσοστό ακρίβειας του μοντέλου στις προβλέψεις. Συγκεκριμένα, το μοντέλο έχει καλύτερη συμπεριφορά στην πρόβλεψη της κλάσης “Engineering” καθώς προβλέπει σωστά κατά τα 25 από τα 26 δείγματα. Αυτό οφείλεται στην επαρκή προεπεξεργασία των δεδομένων αλλά και στο ποσοστό

ανεξαρτησίας των χαρακτηριστικών, πράγμα το οποίο θέτει ως παραδοχή ο συγκεκριμένος ταξινομητής. Το accuracy αναφέρεται στο ποσοστό σωστών προβλέψεων στο σύνολο των παρατηρήσεων και είναι 86%. Αυτό συμβαίνει καθώς οι τιμές TP, TN είναι αρκετά μεγαλύτερες από τις λανθασμένες. Το μέτρο Precision δίνει τιμή 78%, καθώς στην πρόβλεψη της κλάσης Engineer δίνει 25 σωστές και 7 λανθασμένες προβλέψεις. Το ερώτημα στο οποίο απαντά το μέτρο precision είναι το εξής: Από όλες τα βιογραφικά που θεωρήθηκαν ως Engineer, πόσα από αυτά αφορούν όντως Engineer; Το ποσοστό 78% είναι αρκετά ικανοποιητικό. Το μέτρο Recall εξετάζει την ευαισθησία του μοντέλου με την έννοια ότι απαντά στην ερώτηση: Από όλα τα βιογραφικά που αφορούν όντως Information Technology, πόσα ορθώς προβλέφθηκαν ως Information Technology; Η απάντηση του Naive Bayes είναι 96% καθώς η αναλογία είναι 25/1 υπέρ της σωστής πρόβλεψης στο Engineering.

Decision Tree Classifier

Έπειτα, χρησιμοποιήθηκε ο ταξινομητής Decision Tree στα ίδια σύνολα εκπαίδευσης και ελέγχου. Ομοίως με πριν, μετά τις προβλέψεις οπτικοποιήθηκαν τα αποτελέσματά του με Confusion Matrix και μελετήθηκε η ορθότητά τους μέσω των μέτρων accuracy, precision, recall. Αναλυτικότερα, ο κώδικας του μοντέλου φαίνεται παρακάτω.

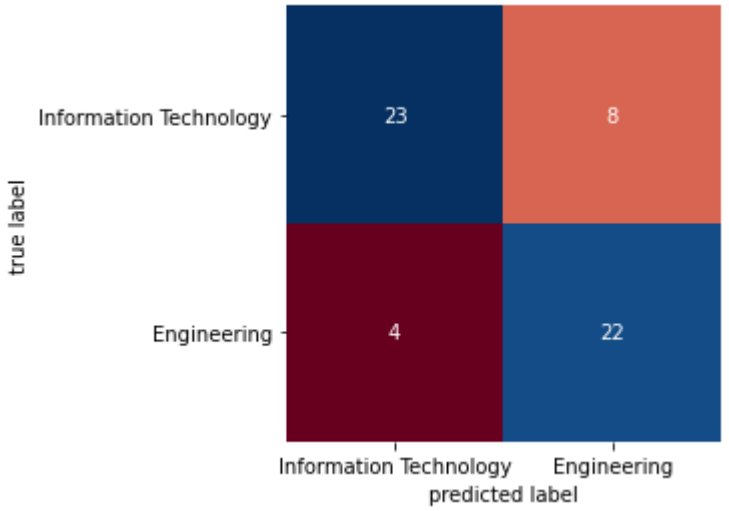
In [71]:
```
from sklearn.model_selection import train_test_split

data['Resume'] = [clean_text(entry) for entry in data['Resume'].values]
data['Resume'] = [to_ascii(entry) for entry in data['Resume'].values]
data['Resume'] = [remove_punctuation(entry) for entry in data['Resume'].values]
data['Resume'] = [entry.lower() for entry in data['Resume'].values]
data['Resume'] = [remove_digits_from_string(entry) for entry in data['Resume'].values]
data['Resume'] = [remove_single_letters(entry) for entry in data['Resume'].values]

X_train, X_test, y_train, y_test = train_test_split(data['Resume'], data['label'], random_state=1)  #Χωρισμός των δεδομένων σε tr
```

In [72]:
```
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import CountVectorizer

cv = CountVectorizer(strip_accents='ascii', token_pattern=u'(?ui)\\b\\w*[a-z]+\\w*\\b', lowercase=True, stop_words='english', tok
tfidf = TfidfTransformer()
classifier = tree.DecisionTreeClassifier(criterion='entropy', splitter='best')

X_train_cv = cv.fit_transform(X_train)
X_test_cv = cv.transform(X_test)

X_train_tfidf = tfidf.fit_transform(X_train_cv)
X_test_tfidf = tfidf.fit_transform(X_test_cv)

classifier.fit(X_train_tfidf, y_train)
predictions=classifier.predict(X_test_tfidf)
```

```
In [21]: from sklearn.metrics import accuracy_score, precision_score, recall_score
         print('Accuracy score:', accuracy_score(y_test, predictions))
         print('Precision score:', precision_score(y_test, predictions))
         print('Recall score:', recall_score(y_test, predictions))

         Accuracy score: 0.7894736842105263
         Precision score: 0.7333333333333333
         Recall score: 0.8461538461538461
```

```
In [23]: from sklearn.metrics import confusion_matrix
         import matplotlib.pyplot as plt
         import seaborn as sns
         cm = confusion_matrix(y_test, predictions)
         sns.heatmap(cm, square=True, annot=True, cmap='RdBu', cbar=False,
         xticklabels=['Information Technology', 'Engineering'], yticklabels=['Information Technology', 'Engineering'])
         plt.xlabel('predicted label')
         plt.ylabel('true label')
```

Οι προβλέψεις του ταξινομητή Decision Tree φαίνονται στον παρακάτω πίνακα:



Στα αποτελέσματα βλέπουμε τις τιμές των μετρήσεων accuracy, precision, recall τα οποία και πάλι υπολογίστηκαν με βάση την ορθότητα των προβλέψεων του ταξινομητή. Οι τιμές αυτές είναι σε ικανοποιητικά όρια. Το accuracy αναφέρεται στο ποσοστό σωστών προβλέψεων στο σύνολο των παρατηρήσεων και είναι 79%. Αυτό συμβαίνει καθώς οι τιμές TP, TN είναι αρκετά μεγαλύτερες από τις λανθασμένες. Το μέτρο Precision δίνει τιμή 73,3%, καθώς στην πρόβλεψη της κλάσης Engineer δίνει 22 σωστές και 8 λανθασμένες προβλέψεις. Το ερώτημα στο οποίο απαντά το μέτρο precision είναι το εξής: Από όλες τα βιογραφικά που θεωρήθηκαν ως Engineer, πόσα από αυτά αφορούν όντως Engineer; Το ποσοστό 73,3% είναι αρκετά ικανοποιητικό. Το μέτρο Recall εξετάζει την ευαισθησία του μοντέλου με την έννοια ότι απαντά στην ερώτηση: Από όλα τα βιογραφικά που αφορούν όντως Engineering, πόσα ορθώς προβλέφθηκαν ως Engineering; Η απάντηση του Decision Tree είναι 85%.

Random Forest Classifier

Στη συνέχεια εφαρμόστηκε ο ταξινομητής Random Forest ο οποίος αποτελείται από πολλούς Decision tree classifiers όπως προαναφέρθηκε. Τα αποτελέσματά του οπτικοποιούνται με confusion matrix όπου φαίνονται οι προβλέψεις αναλυτικά για την κάθε κλάση. Με βάση αυτές τις προβλέψεις υπολογίστηκαν τα μέτρα accuracy, precision, recall ομοίως με τα παραπάνω. Ο κώδικας για το συγκεκριμένο μοντέλο φαίνεται παρακάτω.

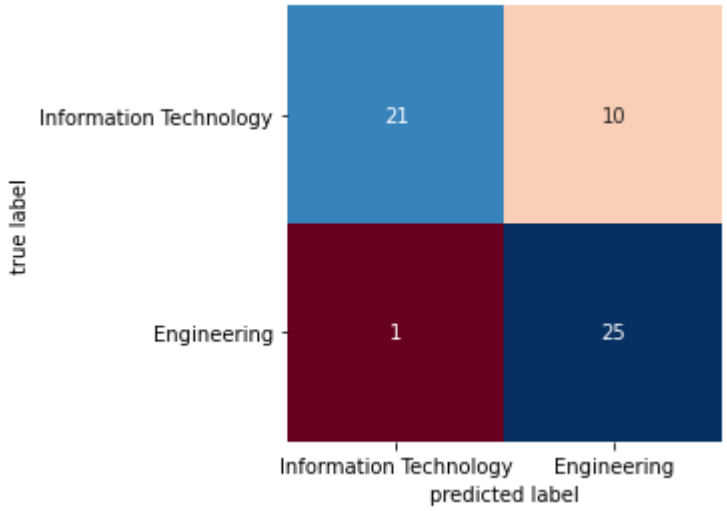
In [76]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import CountVectorizer

cv = CountVectorizer(strip_accents='ascii', token_pattern=u'(?ui)\\b\\w*[a-z]+\\w*\\b', lowercase=True, stop_words='english', tok
tfidf = TfidfTransformer()
classifier = RandomForestClassifier(n_estimators=15, random_state=0)

X_train_cv = cv.fit_transform(X_train)
X_test_cv = cv.transform(X_test)

X_train_tfidf = tfidf.fit_transform(X_train_cv)
X_test_tfidf = tfidf.fit_transform(X_test_cv)

classifier.fit(X_train_tfidf, y_train)
predictions = classifier.predict(X_test_tfidf)
```

In [39]:
```
from sklearn.metrics import accuracy_score, precision_score, recall_score
print('Accuracy score:', accuracy_score(y_test, predictions))
print('Precision score:', precision_score(y_test, predictions))
print('Recall score:', recall_score(y_test, predictions))
```

```
Accuracy score: 0.8070175438596491
Precision score: 0.7142857142857143
Recall score: 0.9615384615384616
```

In [40]:
```
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
cm = confusion_matrix(y_test, predictions)
sns.heatmap(cm, square=True, annot=True, cmap='RdBu', cbar=False,
xticklabels=['Information Technology', 'Engineering'], yticklabels=['Information Technology', 'Engineering'])
plt.xlabel('predicted label')
plt.ylabel('true label')
```

Τα αποτελέσματα του αλγορίθμου φαίνονται στον παρακάτω πίνακα:



Βλέπουμε ότι ο αλγόριθμος πετυχαίνει μεγάλη ακρίβεια στην πρόβλεψη της κατηγορίας Engineering. Η ταχύτητά του είναι μικρότερη σε σχέση με τον αλγόριθμο Decision Trees καθώς αναπτύσσει δάσος (πολλά Decision Trees) για την επίλυση του προβλήματος. Η ακρίβειά του είναι σαφώς μεγαλύτερη από του Decision Trees για τον ίδιο λόγο. Συγκεκριμένα στο σύνολο των προβλέψεων έχει accuracy 81%, ενώ ο Decision Forest έχει accuracy 79%. Το γεγονός αυτό οφείλεται στην ιδιότητα του Random Forest να διεξάγει ψηφοφορία για να επιλέξει τη βέλτιστη πρόβλεψη από όλα τα δένδρα του δάσους. Το μέτρο Precision βρίσκεται στο 71% καθώς προβλέπει 35 βιογραφικά ως Engineering, εκ των οποίων μόλις τα 25 είναι όντως Engineering. Το μέτρο Recall είναι 96% καθώς στην περίπτωση του Engineering επιτεύχθηκε μεγάλη απόδοση αφού 25 στις 26 προβλέψεις ήταν ορθές.

SVM Classifier

Τέλος, εφαρμόστηκε ο ταξινομητής Support Vector Model ο οποίος βασίζεται στο διαχωρισμό των κλάσεων από ένα διαχωριστικό σχήμα (hyperplane) ανάλογα με τις διαστάσεις των δεδομένων. Όπως και στα προηγούμενα μοντέλα, μετά την εκπαίδευση του μοντέλου και τη διεξαγωγή προβλέψεων για τα δεδομένα ελέγχου, εξετάστηκε η απόδοσή του με βάση τα μέτρα accuracy, precision, recall και τα αποτελέσματα οπτικοποιήθηκαν με confusion matrix. Αναλυτικότερα, φαίνεται παρακάτω η διαδικασία.

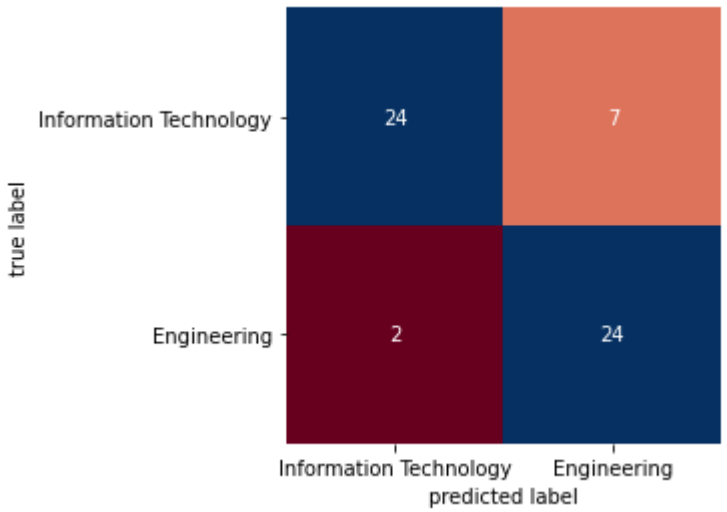
In [16]:
```
from sklearn.svm import LinearSVC
from sklearn import metrics
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer

cv = CountVectorizer(strip_accents='ascii', token_pattern=u'(?ui)\\b\\w*[a-z]+\\w*\\b', lowercase=True, stop_words='english', tok
tfidf = TfidfTransformer()
classifier = LinearSVC(random_state=0, tol=1e-05)

X_train_cv = cv.fit_transform(X_train)
X_test_cv = cv.transform(X_test)

X_train_tfidf = tfidf.fit_transform(X_train_cv)
X_test_tfidf = tfidf.fit_transform(X_test_cv)

classifier.fit(X_train_tfidf, y_train)
predictions = classifier.predict(X_test_tfidf)
```

In [17]:
```
from sklearn.metrics import accuracy_score, precision_score, recall_score
print('Accuracy score:', accuracy_score(y_test, predictions))
print('Precision score:', precision_score(y_test, predictions))
print('Recall score:', recall_score(y_test, predictions))
```

```
Accuracy score: 0.8421052631578947
Precision score: 0.7741935483870968
Recall score: 0.9230769230769231
```

In [18]:
```
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
cm = confusion_matrix(y_test, predictions)
sns.heatmap(cm, square=True, annot=True, cmap='RdBu', cbar=False,
xticklabels=['Information Technology', 'Engineering'], yticklabels=['Information Technology', 'Engineering'])
plt.xlabel('predicted label')
plt.ylabel('true label')
```

Οι προβλέψεις του ταξινομητή φαίνονται στον παρακάτω πίνακα:



Βλέπουμε ότι η ακρίβειά του είναι ικανοποιητική με κάποιες αστοχίες. Οι αστοχίες αυτές μπορεί να οφείλονται στο μεγάλο αριθμό των χαρακτηριστικών, στην ύπαρξη εναπομείναντος θορύβου στα δεδομένα μετά την προεπεξεργασία ή σε άλλους παράγοντες. Από τον Confusion Matrix είναι φανερό ότι το συγκεκριμένο μοντέλο

πέτυχε εξαιρετικό ποσοστό τόσο accuracy όσο και recall. Το accuracy βρίσκεται σε ποσοστό 84% ενώ το recall σε ποσοστό 92%. Αυτό σημαίνει ότι από τα 26 βιογραφικά που στην πραγματικότητα αφορούν Engineering τα 24 προβλέφθηκαν σωστά. Επίσης το μέτρο Precision βρίσκεται σε ποσοστό 77% που σημαίνει ότι από τα 31 βιογραφικά που προβλέφθηκαν συνολικά ως Engineering τα 24 ήταν στην πραγματικότητα Engineering. Αξίζει να σημειωθεί επιπλέον ότι ο συγκεκριμένος αλγόριθμος αποδίδει ικανοποιητικά σε περιπτώσεις όπου το πλήθος των διαστάσεων είναι μεγαλύτερο από το πλήθος των δειγμάτων.

- 2ο Πείραμα

Κατά το δεύτερο πείραμα, επιλέχθηκε από το dataset η ειδικότητα *Education* η οποία περιέχει 102 βιογραφικά. Μετά από διεξοδική μελέτη των βιογραφικών όλων των υποψηφίων τοποθετήθηκε σε κάθε βιογραφικό μια ετικέτα η οποία χαρακτηρίζει την καταλληλότητά του για μια θέση της αντίστοιχης ειδικότητας. Συγκεκριμένα, η ετικέτα “1” αναφέρεται σε κατάλληλο υποψήφιο (63 βιογραφικά) ενώ η ετικέτα “0” σε μη κατάλληλο (26 βιογραφικά). Τα δεδομένα υφίστανται την ίδια προεπεξεργασία με το 1ο πείραμα χρησιμοποιώντας τις συναρτήσεις που κατασκευάστηκαν στο 1ο πείραμα για την προεπεξεργασία των δεδομένων. Οι ταξινομητές αυτή τη φορά καλούνται να ταξινομήσουν τα βιογραφικά των υποψηφίων σύμφωνα με την καταλληλότητά τους για μια θέση.

Η λίστα *Status* που φαίνεται παρακάτω στον κώδικα, περιέχει όλες τις ετικέτες των επιλεγμένων βιογραφικών όπως αυτές τοποθετήθηκαν μετά από μελέτη του περιεχομένου τους. Στη συνέχεια, η λίστα αυτή τοποθετήθηκε ως τελευταία στήλη στο dataframe των δεδομένων.

Και πάλι το νέο σύνολο δεδομένων διαιρέθηκε σε δεδομένα εκπαίδευσης και σε δεδομένα ελέγχου. Συγκεκριμένα, το σύνολο δεδομένων εκπαίδευσης περιλαμβάνει 76 βιογραφικά, ενώ το σύνολο ελέγχου περιλαμβάνει 26 βιογραφικά.

Στο συγκεκριμένο πείραμα έχουμε υποδεκαπλάσιο αριθμό δεδομένων συγκριτικά με το πρώτο πείραμα. Επίσης, όλα τα βιογραφικά αναφέρονται σε υποψηφίους της κατηγορίας Education οπότε περιέχουν παρόμοιους όρους σε μεγάλο βαθμό. Για τους λόγους αυτούς αναμένουμε πιθανότητα μικρότερο ποσοστό ακρίβειας στα αποτελέσματα των ταξινομητών.

Παρακάτω φαίνεται ο κώδικας που αφορά την προεπεξεργασία των δεδομένων και την εισαγωγή της ετικέτας περί καταλληλότητας του υποψηφίου.

In [84]:
```
data=pd.read_csv(r"C:\Users\User\Untitled Folder 2\resume_dataset.csv.zip")
data['Resume'] = [clean_text(entry) for entry in data['Resume'].values]
data['Resume'] = [to_ascii(entry) for entry in data['Resume'].values]
data['Resume'] = [remove_punctuation(entry) for entry in data['Resume'].values]
data['Resume'] = [entry.lower() for entry in data['Resume'].values]
data['Resume'] = [remove_digits_from_string(entry) for entry in data['Resume'].values]
data['Resume'] = [remove_single_letters(entry) for entry in data['Resume'].values]
mask = (data['Category'] == 'Education')
data=data.loc[mask]
index=np.arange(0, len(data))
data.set_index(index)
```

Out[84]:

	ID	Category	Resume
0	271	Education	john smith \ninfo greatresumesfast com ...
1	272	Education	gracy signor\n\n maple avenue rapid city ...
2	273	Education	education sales representative resume\nfiled ...
3	274	Education	career highlights\nachieved recognition for g...
4	275	Education	teaching resume\nany street culver city...
...	...	...	...
97	368	Education	objective\nteaching position in elementary ed...
98	369	Education	job seeker\n pleasant street\nminneapolis mn...
99	370	Education	education resume example principal\n\nsusan...
100	371	Education	sample cover letter\njuly \n\nlaura longori...
101	372	Education	brian luikart\n\ntraining manager\n\n river s...

102 rows × 3 columns

In [243]:
```
status = [1, 1, 0, 0, 0, 1, 0, 1,
          0, 0, 1, 1, 1, 1, 1, 1,
          0, 1, 0, 1, 0, 1, 1, 1,
          0, 1, 0, 1, 1, 0, 0, 0,
          1, 0, 0, 0, 1, 0, 0, 1, 0,
          1, 1, 1, 0, 1, 1, 0, 1, 1,
          1, 0, 1, 1, 0, 1, 1, 0, 1,
          1, 1, 1, 0, 0, 1, 1, 1, 1, 1,
          1, 0, 0, 1, 1, 1, 0, 1, 0, 1,
          0, 1, 0, 1, 0, 1, 1, 0, 1,
          1, 1, 0, 1, 1, 0, 1, 0,
          1, 1, 0, 1, 1, 1]
```

In [244]:
```
data['Status'] = status
```

Naive Bayes Classifier

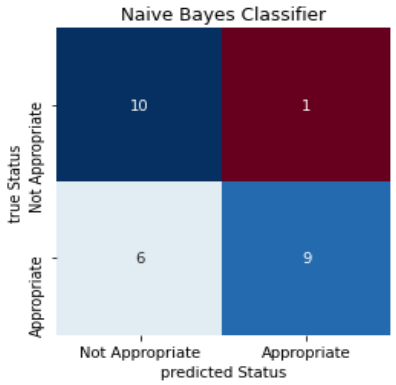
Αρχικά εφαρμόστηκε ο αλγόριθμος Naive Bayes για την ταξινόμηση των βιογραφικών Education ανάλογα με την καταλληλότητά τους για μια θέση εργασίας. Μετά την εφαρμογή του αλγορίθμου, τα αποτελέσματά του οπτικοποιήθηκαν μέσω ενός Confusion Matrix με τη βοήθεια του sklearn και τελικά παρουσιάστηκαν τα μέτρα απόδοσής του accuracy, precision, recall.

In [87]:
```
#We will classify resumes according to their status(Appropriate or Not)
#Naive Bayes Classifier

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data['Resume'], data['Status'], random_state=1)

from sklearn.feature_extraction.text import CountVectorizer

cv = CountVectorizer(strip_accents='ascii', token_pattern=u'(?ui)\\b\\w*[a-z]+\\w*\\b', lowercase=True, stop_words='english', tok
X_train_cv = cv.fit_transform(X_train)
X_test_cv = cv.transform(X_test)

from sklearn.naive_bayes import MultinomialNB  #fit the model and make predictions
naive_bayes = MultinomialNB()
naive_bayes.fit(X_train_cv, y_train)
predictions = naive_bayes.predict(X_test_cv)
```

In [35]:
```
from sklearn.metrics import accuracy_score, precision_score, recall_score
print('Accuracy score:', accuracy_score(y_test, predictions))
print('Precision score: ', precision_score(y_test, predictions))
print('Recall score: ', recall_score(y_test, predictions))

from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

cm = confusion_matrix(y_test, predictions)
sns.heatmap(cm, square=True, annot=True, cmap='RdBu', cbar=False,
xticklabels=['Not Appropriate', 'Appropriate'], yticklabels=['Not Appropriate', 'Appropriate'])
plt.title('Naive Bayes Classifier')
plt.xlabel('predicted Status')
plt.ylabel('true Status')
```

```
Accuracy score: 0.7307692307692307
Precision score:  0.9
Recall score:  0.6
```

Out[35]:
```
Text(91.68, 0.5, 'true Status')
```



Βλέπουμε ότι η ακρίβεια βρίσκεται στο 73% που σημαίνει ότι και για τις δύο κλάσεις (Appropriate, Not Appropriate) το μοντέλο πρόβλεψε σωστά 19 βιογραφικά από τα 26 στο σύνολο. Το μέτρο Precision είναι 90% που σημαίνει ότι από τις 10 προβλέψεις για κατάλληλο υποψήφιο, οι 9 αφορούσαν όντως κατάλληλο υποψήφιο και η 1 αφορούσε μη κατάλληλο υποψήφιο. Το μέτρο Recall είναι 60% που σημαίνει ότι από τα 15 βιογραφικά που αφορούν κατάλληλο υποψήφιο, τα 9 προβλέφθηκαν ως κατάλληλα.

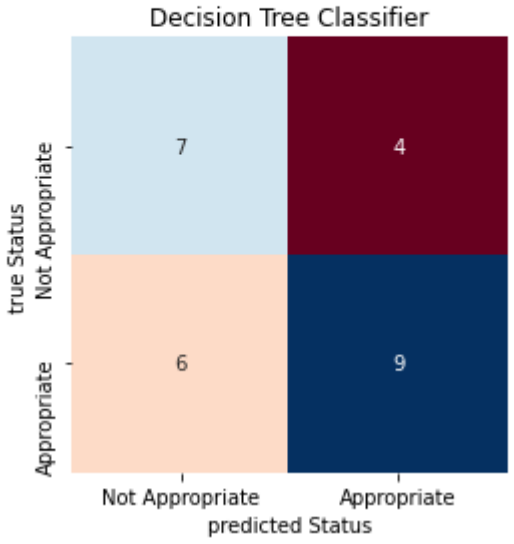
Decision Tree Classifier

Δευτερευόντως εφαρμόστηκε ο αλγόριθμος Decision Tree στα δεδομένα εκπαίδευσης και τα αποτελέσματά του μετρήθηκαν με βάση τις προβλέψεις στα δεδομένα ελέγχου. Τα αποτελέσματα αυτά οπτικοποιήθηκαν με Confusion Matrix ώστε να φανεί η απόδοση του μοντέλου και με βάση αυτά υπολογίστηκαν τα μέτρα accuracy, precision, recall.

In [40]:
```
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import CountVectorizer

cv = CountVectorizer(strip_accents='ascii', token_pattern=u'(?ui)\\b\\w*[a-z]+\\w*\\b', lowercase=True, stop_words='english', tok
tfidf = TfidfTransformer()
classifier = tree.DecisionTreeClassifier(criterion='entropy', splitter='best')

X_train_cv = cv.fit_transform(X_train)
X_test_cv = cv.transform(X_test)

X_train_tfidf = tfidf.fit_transform(X_train_cv)
X_test_tfidf = tfidf.fit_transform(X_test_cv)

classifier.fit(X_train_tfidf, y_train)
predictions=classifier.predict(X_test_tfidf)
print('Accuracy score:', accuracy_score(y_test, predictions))
print('Precision score: ', precision_score(y_test, predictions))
print('Recall score: ', recall_score(y_test, predictions))

cm = confusion_matrix(y_test, predictions)
sns.heatmap(cm, square=True, annot=True, cmap='RdBu', cbar=False,
xticklabels=['Not Appropriate', 'Appropriate'], yticklabels=['Not Appropriate', 'Appropriate'])
plt.title('Decision Tree Classifier')
plt.xlabel('predicted Status')
plt.ylabel('true Status')
```

```
Accuracy score: 0.6153846153846154
Precision score:  0.6923076923076923
Recall score:  0.6
```

Out[40]: Text(91.68, 0.5, 'true Status')



Βλέπουμε ότι η ακρίβεια του μοντέλου "πέφτει" στο 62%. Το γεγονός αυτό μπορεί να οφείλεται στο μικρό πλήθος δεδομένων εκπαίδευσης. Το μέτρο Precision του συγκεκριμένου ταξινομητή για τα δεδομένα βρίσκεται στο 69% που σημαίνει ότι από τα 13 βιογραφικά που προβλέφθηκαν ως κατάλληλα, τα 9 μόνο αφορούσαν κατάλληλο υποψήφιο. Επιπλέον, το μέτρο Recall βρίσκεται στο 60% καθώς από τα 15 βιογραφικά που ήταν όντως κατάλληλα με βάση το status, μόνο τα 9 προβλέφθηκαν ως κατάλληλα.

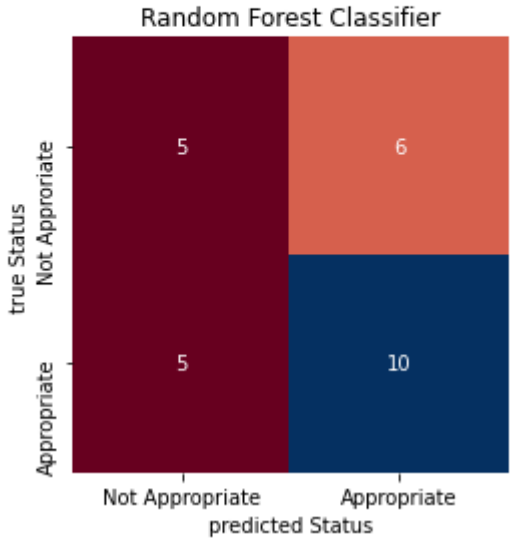
Random Forest Classifier

Στη συνέχεια, εφαρμόστηκε στα δεδομένα εκπαίδευσης ο ταξινομητής Random Forest ο οποίος αναμένουμε να δώσει καλύτερα αποτελέσματα από τον Decision Tree καθώς δημιουργεί πολλαπλά δένδρα απόφασης. Παρακάτω βλέπουμε τις προβλέψεις που έδωσε ο συγκεκριμένος ταξινομητής για το status των δεδομένων ελέγχου.

In [42]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import CountVectorizer

cv = CountVectorizer(strip_accents='ascii', token_pattern=u'(?ui)\\b\\w*[a-z]+\\w*\\b', lowercase=True, stop_words='english', tok
tfidf = TfidfTransformer()
classifier = RandomForestClassifier(n_estimators=15, random_state=0)

X_train_cv = cv.fit_transform(X_train)
X_test_cv = cv.transform(X_test)

X_train_tfidf = tfidf.fit_transform(X_train_cv)
X_test_tfidf = tfidf.fit_transform(X_test_cv)

classifier.fit(X_train_tfidf, y_train)
predictions = classifier.predict(X_test_tfidf)

from sklearn.metrics import accuracy_score, precision_score, recall_score
print('Accuracy score:', accuracy_score(y_test, predictions))
print('Precision score:', precision_score(y_test, predictions))
print('Recall score:', recall_score(y_test, predictions))

from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
cm = confusion_matrix(y_test, predictions)
sns.heatmap(cm, square=True, annot=True, cmap='RdBu', cbar=False,
xticklabels=['Not Appropriate', 'Appropriate'], yticklabels=['Not Approriate', 'Appropriate'])
plt.title('Random Forest Classifier')
plt.xlabel('predicted Status')
plt.ylabel('true Status')
```

```
Accuracy score: 0.5769230769230769
Precision score: 0.625
Recall score: 0.6666666666666666
```

Out[43]: Text(91.68, 0.5, 'true Status')



Βλέπουμε ότι η ακρίβεια του αλγορίθμου βρίσκεται στο 58% καθώς 15 από τα 26 βιογραφικά προβλέφθηκαν σωστά. Το μέτρο Precision είναι 63% που σημαίνει ότι από τις 16 προβλέψεις κατάλληλων βιογραφικών, τα 10 ήταν όντως κατάλληλα. Το μέτρο Recall βρίσκεται σε ποσοστό 67% που σημαίνει ότι από 15 κατάλληλα βιογραφικά, ως κατάλληλα προβλέφθηκαν μόλις τα 15. Παρατηρούμε ότι μόνο το μέτρο Recall του Random Forest υπερτερεί έναντι του Recall του Decision Tree. Τα μέτρα accuracy, precision είναι χαμηλότερα. Το γεγονός αυτό πιθανότατα οφείλεται στο μικρό αριθμό των δεδομένων ή σε ενδεχόμενο overfitting του αλγορίθμου στα δεδομένα.

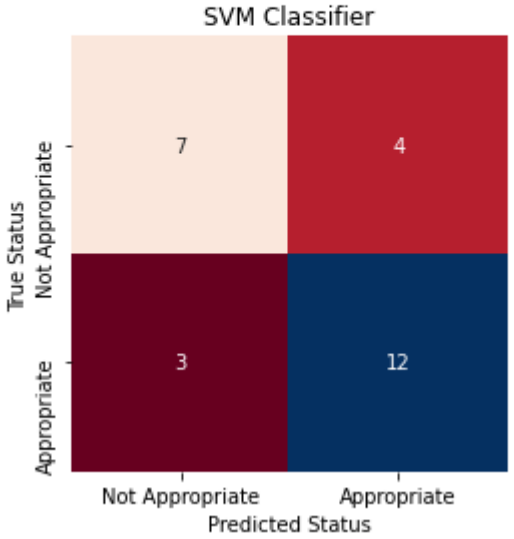
SVM Classifier

Τέλος εφαρμόστηκε ο αλγόριθμος ταξινόμησης Support Vector Model στα δεδομένα εκπαίδευσης. Ακολούθως ελέγχθηκε η αποδοτικότητά του με βάση τις προβλέψεις του status των δεδομένων ελέγχου. Η οπτικοποίηση των αποτελεσμάτων έγινε μέσω ενός Confusion Matrix και η αξιολόγηση της απόδοσης μέσω των μέτρων accuracy, precision, recall. Παρακάτω βλέπουμε τον κώδικα εφαρμογής του μοντέλου.

In [44]:
```
from sklearn.svm import LinearSVC
from sklearn import metrics
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer

cv = CountVectorizer(strip_accents='ascii', token_pattern=u'(?ui)\\b\\w*[a-z]+\\w*\\b', lowercase=True, stop_words='english', tok
tfidf = TfidfTransformer()
classifier = LinearSVC(random_state=0, tol=1e-05)

X_train_cv = cv.fit_transform(X_train)
X_test_cv = cv.transform(X_test)

X_train_tfidf = tfidf.fit_transform(X_train_cv)
X_test_tfidf = tfidf.fit_transform(X_test_cv)

classifier.fit(X_train_tfidf, y_train)
predictions = classifier.predict(X_test_tfidf)

from sklearn.metrics import accuracy_score, precision_score, recall_score
print('Accuracy score:', accuracy_score(y_test, predictions))
print('Precision score:', precision_score(y_test, predictions))
print('Recall score:', recall_score(y_test, predictions))

from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns
cm = confusion_matrix(y_test, predictions)
sns.heatmap(cm, square=True, annot=True, cmap='RdBu', cbar=False,
xticklabels=['Not Appropriate', 'Appropriate'], yticklabels=['Not Appropriate', 'Appropriate'])
plt.title('SVM Classifier')
plt.xlabel('Predicted Status')
plt.ylabel('True Status')
```

```
Accuracy score: 0.7307692307692307
Precision score: 0.75
Recall score: 0.8
```

Out[44]: Text(91.68, 0.5, 'True Status')



Με βάση τα αποτελέσματα των μετρήσεων φαίνεται ότι ο SVM ταξινομητής είναι περισσότερο αποδοτικός στο 2ο πρόβλημα συγκριτικά με τους Decision Trees/ Random Forest. Συγκεκριμένα, το accuracy βρίσκεται σε ποσοστό 73%, το precision 75% και το recall 80%.

7. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

7.1 Πλεονεκτήματα χρήσης τεχνικών μηχανικής μάθησης στην ταξινόμηση βιογραφικών σημειωμάτων

Στο πλαίσιο της παρούσας διπλωματικής εργασίας μελετήθηκε εκτενώς η εφαρμογή μοντέλων μηχανικής μάθησης για ταξινόμηση βιογραφικών σημειωμάτων και η αξιολόγηση της απόδοσής τους. Το ερώτημα είναι πόσο σημαντικό είναι το όφελος που προκύπτει από μια τέτοια μελέτη, τι πρακτική εφαρμογή έχει. Η απάντηση στο παραπάνω ερώτημα βρίσκεται στις εσωτερικές διεργασίες των επιχειρήσεων και των οργανισμών. Στη σύγχρονη εποχή του διαδικτύου, οι εταιρείες και οι οργανισμοί έχουν πρόσβαση σε πληθώρα πληροφοριών μεταξύ των οποίων είναι και τα βιογραφικά σημειώματα υποψηφίων οι οποίοι τα θέτουν στη διάθεση των εργοδοτών προς αξιολόγηση. Πλέον, κάθε υποψήφιος έχει τη δυνατότητα να εκθέσει το βιογραφικό του σε διαδικτυακές πλατφόρμες ή στα μέσα κοινωνικής δικτύωσης προκειμένου να γίνει μέλος μιας ψηφιακής κοινότητας ώστε να έχει άμεση πρόσβαση σε ευκαιρίες εργασίας που του αντιστοιχούν. Από την άλλη πλευρά, οι εταιρείες, εκμεταλλευόμενες τις διαδικτυακές πλατφόρμες, γνωστοποιούν την ανάγκη για συνεργασία με υπαλλήλους και, ως εκ τούτου, έχουν τη δυνατότητα να συλλέγουν δεδομένα για τεράστιο αριθμό υποψηφίων. Ο όγκος των δεδομένων αυτών αφενός καθιστά ιδιαίτερα δύσκολη και επίπονη τη διαχείρισή τους και αφετέρου δημιουργεί μια πρόκληση για τη δημιουργία εργαλείων με σκοπό την όλο και αποτελεσματικότερη αξιοποίησή τους. Θα ήταν πρακτικά αδύνατη η αξιοποίηση και η διαχείριση όλου αυτού του όγκου δεδομένων με παραδοσιακά μέσα που χρησιμοποιούνταν στο παρελθόν. Καθοριστικό ρόλο έπαιξε η χρήση της μηχανικής μάθησης και της τεχνητής νοημοσύνης που αντικατέστησαν τον ανθρώπινο παράγοντα στη διαδικασία διαχείρισης των βιογραφικών που συλλέγονται από τις εταιρείες και τους οργανισμούς. Η χρήση μεθόδων μηχανικής μάθησης στη διαχείριση των βιογραφικών δίνει τη δυνατότητα στους εργοδότες να έχουν σφαιρική εικόνα των αιτούντων και των ικανοτήτων τους αξιοποιώντας οπτικοποιήσεις και συνολικές μετρήσεις επί των δεδομένων. Επίσης, με τον τρόπο αυτό, μειώνεται δραματικά ο χρόνος αξιολόγησης των βιογραφικών καθώς η διαδικασία αυτή πραγματοποιείται από υπολογιστή και όχι από ανθρώπους. Επομένως επισπεύδονται όλες οι διαδικασίες μέχρι την τελική τοποθέτηση του εργαζομένου στην κατάλληλη θέση. Η αυτοματοποίηση της διαδικασίας ταξινόμησης των υποψηφίων σε κατηγορίες με βάση την ειδικότητά τους, βοηθάει επίσης τους οργανισμούς να παρατηρούν τη γενικότερη τάση των αιτούντων για μία θέση εργασίας και να διεξάγουν στατιστικές αναλύσεις για μελλοντική χρήση και εξαγωγή συμπερασμάτων. Σημαντικό πλεονέκτημα για τους οργανισμούς αποτελεί επίσης το γεγονός ότι τα μοντέλα που δημιουργούνται είναι διαθέσιμα για συνεχή χρήση και για δεδομένα εισόδου τα οποία ανα πάσα στιγμή βρίσκονται εκτεθειμένα στις βάσεις δεδομένων τους. Διαφορετικά δεδομένα εισόδου σε πλήθος ή σε μορφή μπορεί να

απαιτούν μικρές τροποποιήσεις στα μοντέλα για την κατάλληλη απόδοσή τους, ωστόσο αποτελούν ένα σταθερό όπλο στα χέρια των υπεύθυνων ανθρώπινου δυναμικού.

7.2 Συμπεράσματα από τη χρήση αλγορίθμων ταξινόμησης σε δεδομένα κειμένου

Με τη διεξαγωγή των τριών πειραμάτων στο πλαίσιο της παρούσας εργασίας με τη βοήθεια των βιογραφικών σημειωμάτων που αντλήθηκαν από το dataset του ιστοτόπου Kaggle, καταλήγουμε σε ορισμένα γενικότερα συμπεράσματα πέραν των απτών, αριθμητικών αποτελεσμάτων που λάβαμε. Αρχικά, όπως έχει παρατηρηθεί και ειπωθεί σε πολλές μελέτες ανάλυσης και ταξινόμησης κειμένου, η προεπεξεργασία των δεδομένων οφείλει να είναι ένα αναπόσπαστο μέρος της διαδικασίας ταξινόμησης κειμένων. Τα δεδομένα που κυκλοφορούν ελεύθερα στο διαδίκτυο περιέχουν μεγάλο ποσοστό θορύβου που συνήθως τα καθιστά μη διαχειρίσιμα. Ο ερευνητής, που προσπαθεί να πετύχει μεγάλα ποσοστά ακρίβειας, οφείλει να επενδύσει χρόνο και μνήμη στην βέλτιστη προεπεξεργασία των κειμένων. Μια ελλειπής ή κακή προεπεξεργασία αλλοιώνει σε μεγάλο βαθμό την απόδοση του μοντέλου. Αντιθέτως, η σωστή επεξεργασία οδηγεί στη μείωση του θορύβου, στη μείωση του όγκου των δεδομένων και τελικά στην καλύτερη απόδοση των ταξινομητών. Επίσης, έγινε φανερό από τα πειράματα ότι τα κείμενα με σαφείς διαφοροποιήσεις στο περιεχόμενό τους βοηθούσαν τον ταξινομητή να προβλέψει, με μεγαλύτερη ευκολία και συνεπώς λιγότερες αστοχίες, το γεγονός ότι ανήκουν σε διαφορετικές κλάσεις. Επιπλέον, είδαμε ότι οι παραδοχές που οδηγούν σε απλούστευση του προβλήματος αλλά κατά κύριο λόγο δεν ανταποκρίνονται στην πραγματικότητα, πολλές φορές οδηγούν σε αξιόπιστα αποτελέσματα. Χαρακτηριστικό παράδειγμα αποτελεί ο ταξινομητής Naive Bayes ο οποίος θέτει την παραδοχή περί ανεξαρτησίας των χαρακτηριστικών η οποία σπανίως ανταποκρίνεται στην πραγματικότητα. Όπως διαπιστώθηκε, ο ταξινομητής Naive Bayes πέτυχε εξαιρετικά ποσοστά ακρίβειας. Μια επιπλέον παρατήρηση με βάση τα πειράματα είναι ότι κάποιες φορές η ελαχιστοποίηση του χρόνου πρόβλεψης του μοντέλου υπερτερεί έναντι της απόλυτης ακρίβειας και το αντίστροφο. Εξαρτάται από τις ανάγκες του χρήστη τι επιθυμεί να θέσει ως προτεραιότητα. Τέλος, παρατηρήσαμε στην πλειοψηφία των μοντέλων ότι ο κάθε αλγόριθμος επιβεβαιώνει τα καταγεγραμμένα στη θεωρία πλεονεκτήματα και μειονεκτήματά του.

7.3 Προτάσεις για το μέλλον

Μετά από εκτεταμένη μελέτη είδαμε ότι η χρήση μεθόδων μηχανικής μάθησης για ταξινόμηση βιογραφικών σημειωμάτων βρίσκει πολλές εφαρμογές στο βιομηχανικό και επιχειρηματικό τομέα και αποτελεί μια καινοτομία στα χέρια των υπεύθυνων ανθρώπινου δυναμικού. Πλέον υπάρχει μεγάλος αριθμός ιστοσελίδων, όπως LinkedIn, Collegelink, kariera, Indeed κλπ, οι οποίες διευκολύνουν την επαφή μεταξύ των οργανισμών και των υποψηφίων. Οι εταιρείες έχουν τη δυνατότητα να

δημοσιεύουν αγγελίες εργασίας και οι υποψήφιοι να αναζητούν αγγελίες συναφείς με την ειδικότητά τους. Επίσης, στις ιστοσελίδες αυτές οι υποψήφιοι μπορούν να δημιουργούν προφίλ με τα στοιχεία τους, την εκπαίδευση και τις δεξιότητές τους και το σύστημα να τους προτείνει σχετικές αγγελίες. Στο σημείο αυτό θα ήταν σκόπιμο να μπορεί ο υποψήφιος να ανεβάζει το πλήρες βιογραφικό του σε μορφή κειμένου (doc, pdf), στο οποίο θα εφαρμόζονται ταξινομητές προκειμένου να λειτουργεί ένα recommendation system αγγελιών προς τον ίδιο. Από την πλευρά των εταιρειών οι αυτόματες ταξινομήσεις βιογραφικών μπορούν να παρέχουν recommendations από χρήστες της ιστοσελίδας που ταιριάζουν στις ανάγκες της εταιρείας. Εκτός από τις ιστοσελίδες ευρέσεως εργασίας, πολλές εταιρείες ενημερώνουν τις δικές τους ιστοσελίδες μέσω της στήλης Careers για ανοιχτές θέσεις εργασίας. Στο ίδιο περιβάλλον, οι ενδιαφερόμενοι αναρτούν τα βιογραφικά τους προς αξιολόγηση από τους υπεύθυνους. Σημαντική θα ήταν η εφαρμογή αυτόματης ταξινόμησης κειμένου στα αναρτημένα βιογραφικά εντός της ιστοσελίδας, ώστε να προβλεφθεί ποιός από τους υποψηφίους είναι κατάλληλος ή όχι για τη θέση (Appropriate, Not Appropriate Classification). Πέραν της επιλογής ανθρώπινου δυναμικού, η ταξινόμηση κειμένου με μηχανική μάθηση μπορεί να εφαρμοστεί και σε άλλες επιχειρησιακές λειτουργίες εσωτερικές ή εξωτερικές. Για παράδειγμα, μπορεί να εφαρμοστεί κατά την αξιολόγηση των ανθρώπινων πόρων με ταξινόμηση γραπτών αναφορών από τους προϊσταμένους των τμημάτων. Επίσης μπορεί να εφαρμοστεί στην περίπτωση διεξαγωγής έρευνας από το τμήμα ανθρώπινου δυναμικού σχετικά με την ικανοποίηση ή μη των υπαλλήλων από την εταιρεία. Με τη βοήθεια γραπτών κειμένων από τους ίδιους τους υπαλλήλους και με την ταξινόμηση αυτών, μπορούν να προκύψουν σημαντικά στοιχεία και συμπεράσματα για τους υπεύθυνους ώστε να αναπροσαρμόσουν τη στρατηγική τους. Τέλος, εκτός από τους ανθρώπινους πόρους, μπορεί να εντατικοποιηθεί η χρήση της ταξινόμησης κειμένου σε ό,τι αφορά τις κριτικές πελατών ή στα μέιλ πελατών ώστε να κατατάσσονται σε θετικά ή αρνητικά ή ανάλογα με το αίτημά τους σε διάφορες κατηγορίες.

Εκτός από τον επιχειρησιακό τομέα, μπορούμε να πούμε ότι η χρήση μηχανικής μάθησης για ταξινόμηση κειμένου ανοίγει το δρόμο για περαιτέρω εφαρμογές και σε άλλους τομείς, όπως για παράδειγμα τον πανεπιστημιακό. Πολλά οφέλη θα προσέφερε η αυτοματοποίηση της ταξινόμησης των βιογραφικών των φοιτητών οι οποίοι αιτούνται την παρακολούθηση ορισμένων προγραμμάτων σε πανεπιστήμια. Ειδικά στις περιπτώσεις όπου επιλέγεται ένας συγκεκριμένος αριθμός φοιτητών από το πανεπιστήμιο, μπορούν να εφαρμοστούν τεχνικές ταξινόμησης σε ό,τι αφορά την καταλληλότητα ή μη των υποψηφίων για την παρακολούθηση του εκάστοτε προγράμματος. Τέλος, η ταξινόμηση κειμένου στην επιλογή φοιτητών μπορεί να εφαρμοστεί, εκτός από τα βιογραφικά, και σε άλλου είδους έγγραφα όπως συστατικές επιστολές εργοδοτών ή καθηγητών, cover letters κλπ, ώστε να ληφθεί τελική απόφαση περί καταλληλότητας του υποψηφίου.

ΒΙΒΛΙΟΓΡΑΦΙΑ

"Επεξεργασία και Κατανόηση Φυσικής Γλώσσας." https://repository.kallipos.gr/bitstream/11419/3385/1/02_chapter_07.pdf. Accessed 5 1 2021.

"Επεξεργασία φυσικής γλώσσας." *Επεξεργασία φυσικής γλώσσας*, wikipedia, https://el.wikipedia.org/wiki/%CE%95%CF%80%CE%B5%CE%BE%CE%B5%CF%81%CE%B3%CE%B1%CF%83%CE%AF%CE%B1_%CF%86%CF%85%CF%83%CE%B9%CE%BA%CE%AE%CF%82_%CE%B3%CE%BB%CF%8E%CF%83%CF%83%CE%B1%CF%82. Accessed 7 1 2021.

Νταλιακούρας, Νικόλαος. *Αυτόματη Δημιουργία Ερωτήσεων/Ασκήσεων για Εκπαιδευτικό Σύστημα Διδασκαλίας Τεχνητής Νοημοσύνης*. 2016, https://nemertes.lis.upatras.gr/jspui/bitstream/10889/9424/4/Ntaliakouras%28com%29.pdf. Accessed 5 1 2021.

Γκάσης, Παύλος, translator. *Διοίκηση Επιχειρήσεων*. 11 ed., πανεπιστημιο του Sheffield, ΕΚΔΟΣΕΙΣ ΤΖΙΟΛΑ, 2017.

Τσιλιγιάννη, Ελένη. *Αλγόριθμοι Classification σε Big Data*. Αθήνα, 2015.

Κατάκης, Ιωάννης. *Machine learning methods for automated text classification*. 2009.

"Decision Tree - Classification." *Saedsayad*, https://www.saedsayad.com/decision_tree.htm.

"Decision Tree Classification in Python." *Datacamp*, https://www.datacamp.com/community/tutorials/decision-tree-classification-python.

"Document feature extraction and classification." *Towards Data Science*, https://towardsdatascience.com/document-feature-extraction-and-classification-53f0e813d2d3.

"8 best Python Natural Language Processing (NLP) libraries." *Sunscrapers*, https://sunscrapers.com/blog/8-best-python-natural-language-processing-nlp-libraries/.

"Feature Selection Techniques in Machine Learning." *Analytics Vidhya*, https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/.

"A Gentle Introduction to Scikit-Learn: A Python Machine Learning Library." *Machine Learning Mastery*, https://machinelearningmastery.com/a-gentle-introduction-to-scikit-learn-a-python-machine-learning-library/. Accessed 27 1 2021.

"More Performance Evaluation Metrics for Classification Problems You Should Know." *KdNuggets*, https://www.kdnuggets.com/2020/04/performance-evaluation-metrics-classification.html.

"Naive Bayes Classifier." *Towards Data Science*, https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c.

"Naive Bayes Classifier." *wikipedia*, https://en.wikipedia.org/wiki/Naive_Bayes_classifier.

"Κατανόηση φυσικής γλώσσας: Τι είναι και πώς διαφέρει από το NLP." https://zephyrnet.com/el/natural-language-understanding-what-is-it-and-how-is-it-different-from-nlp/.

"A practical explanation of a Naive Bayes classifier." *MonkeyLearn*, https://monkeylearn.com/blog/practical-explanation-naive-bayes-classifier/.

"Python for NLP: Introduction to the Pattern Library." *StackAbuse*, https://stackabuse.com/python-for-nlp-introduction-to-the-pattern-library/.

"The Random forest algorithm." *Packt*, https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781784396909/6/ch06lvl1sec42/the-random-forest-algorithm.

Resham N. Waykole, and Anuradha D. Thakare. "A REVIEW OF FEATURE EXTRACTION METHODS FOR TEXT CLASSIFICATION." http://ijaerd.com/papers/finished_papers/A_review_of_feature_extraction_methods_for_text_classification-IJAERDV05I0489982.pdf.

"SHRDLU." *SHRDLU*, https://en.wikipedia.org/wiki/SHRDLU.

"Spacy." *wikipedia*, https://en.wikipedia.org/wiki/SpaCy.

"Support Vector Machine — Introduction to Machine Learning Algorithms." *Towards Data Science*, https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

"Text Analytics for Beginners using NLTK." *Datacamp*, https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk.

"TextBlob: Simplified Text Processing." *ReadtheDocs*, https://textblob.readthedocs.io/en/dev/.

"Text Classification." *MonkeyLearn*, https://monkeylearn.com/text-classification/.

"Text mining." *Wikipedia*, https://en.wikipedia.org/wiki/Text_mining. Accessed 26 1 2021.

"Text Mining in Python: Steps and Examples." *pubtowardsai*, https://pub.towardsai.net/text-mining-in-python-steps-and-examples-78b3f8fd913b.

"Understanding Confusion Matrix." *Becoming Human*, https://becominghuman.ai/understanding-confusion-matrix-eb6f0f662c3a.

"Understanding Random Forests Classifiers in Python." *Datacamp*, https://www.datacamp.com/community/tutorials/random-forests-classifier-python.

"What is Text Mining, Text Analytics and Natural Language Processing?" *What is Text Mining, Text Analytics and Natural Language Processing?*, https://www.linguamatics.com/what-text-mining-text-analytics-and-natural-language-processing.