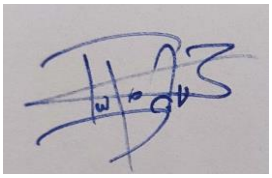
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.

# ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

## Machine Learning Classifiers ως ψηφιακό εργαλείο για την διάγνωση ιατρικών παθήσεων, από αρχεία κειμένου, ψηφιακών φακέλων Υγείας με τεχνικές NLP

**Ιωάννης Πολύζος,** **Α.Μ.: HN08166**

Επιβλέπων: Κωνσταντίνος Οικονόμου, Ακαδημαϊκός Υπότροφος



*(Υπογραφή)*

.....

**ΙΩΑΝΝΗΣ ΠΟΛΥΖΟΣ**

Ηλεκτρολόγος Μηχανικός Τ.Ε., ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

# ΠΕΡΙΛΗΨΗ

Η πολυπλοκότητα και η αύξηση των δεδομένων στον τομέα της υγειονομικής περίθαλψης καθιστά την εφαρμογή της τεχνητής νοημοσύνης όλο και περισσότερο απαραίτητη. Αρκετοί τύποι τεχνητής νοημοσύνης χρησιμοποιούνται ήδη από παρόχους περίθαλψης και εταιρείες βιοεπιστημών. Οι βασικές κατηγορίες εφαρμογών περιλαμβάνουν συστάσεις διάγνωσης και θεραπείας, εμπλοκή και συμμόρφωση ασθενών και διοικητικές δραστηριότητες. Αν και υπάρχουν πολλές περιπτώσεις στις οποίες η τεχνητή νοημοσύνη μπορεί να εκτελεί καθήκοντα υγειονομικής περίθαλψης εξίσου καλά ή καλύτερα από τους ανθρώπους, οι παράγοντες υλοποίησης θα αποτρέψουν την αυτοματοποίηση μεγάλης κλίμακας των θέσεων εργασίας των επαγγελματιών υγείας για μεγάλο χρονικό διάστημα. Επομένως η μηχανική μάθηση μπορεί να είναι το κλειδί για την μέγιστη αξιοποίηση των ιατρικών αρχείων. Η εργασία χωρίζεται σε τρία κεφάλαια στα οποία αναλύονται θεμελιώδη ζητήματα της χρήσης της μηχανικής μάθησης στον υγειονομικό τομέα.

**Λέξεις Κλειδιά:** Μηχανική Μάθηση, ταξινομητές, υποστήριξη κλινικών αποφάσεων, συστήματα ηλεκτρονικών αρχείων υγείας

# ABSTRACT

The complexity and growth of data in healthcare makes the application of artificial intelligence increasingly necessary. Several types of AI are already being used by healthcare providers and life sciences companies. Key application categories include diagnosis and treatment recommendations, patient engagement and compliance, and administrative activities. While there are many cases in which AI can perform healthcare tasks as well or better than humans, implementation factors will prevent large-scale automation of healthcare professionals' jobs for a long time. So machine learning can be the key to making the most of medical records. The paper is divided into three chapters in which fundamental issues of the use of machine learning in the health sector are analyzed.

**Keywords: M**achine learning, classifiers, clinical decision support, electronic health record systems

# ΕΥΧΑΡΙΣΤΙΕΣ

Αρχικά θα ήθελα να ευχαριστήσω τον επιβλέποντα της πτυχιακής μου εργασίας Κωνσταντίνο Οικονόμου. Τον ευχαριστώ για όλες τις συμβουλές, και την καθοδήγηση που μου έδωσε όλο αυτό το διάστημα. Επίσης θέλω να ευχαριστήσω τον επιστημονικό συνεργάτη μου από το Vrije Universiteit Amsterdam, Tjardo Maarseveen. Οι γνώσεις του πάνω στο αντικείμενο της εργασίας αλλά και η βοήθειά του, αποτέλεσαν σημαντικό στοιχείο για την ολοκλήρωση της εργασίας. Με την ολοκλήρωση της παρούσας εργασίας κλείνει και ο κύκλος των προπτυχιακών μου σπουδών στη σχολή Ηλεκτρολόγων Μηχανικών του ΠΔΜ. Σε όλη αυτή την διαδρομή απέκτησα γνώσεις και εμπειρίες οι οποίες θα με αντιπροσωπεύουν ως άτομο από εδώ και στο εξής. Για αυτές τις εμπειρίες θα ήθελα να ευχαριστήσω από καρδιάς όλους ανεξαιρέτως τους φίλους και συμφοιτητές μου. Τέλος θα ήθελα να ευχαριστήσω την οικογένειά μου για όλη την υποστήριξη που μου προσφέρουν όλα αυτά τα χρόνια, δίνοντας μου μεταξύ άλλων και την δυνατότητα να σπουδάσω αυτό που ήθελα.

# ΠΊΝΑΚΑΣ ΠΕΡΙΕΧΟΜΈΝΩΝ

## Περιεχόμενα

# ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

# ΕΙΣΑΓΩΓΗ

Η μηχανική μάθηση είναι μια εφαρμογή της Τεχνητής Νοημοσύνης που χρησιμοποιεί αλγόριθμους και στατιστικές για να βρει μοτίβα σε μεγάλες ποσότητες δεδομένων. Το λογισμικό μηχανικής μάθησης αναλύει αυτά τα δεδομένα και στη συνέχεια «μαθαίνει» από αυτά εφαρμόζοντας μοτίβα από τα οποία μπορεί να κάνει προβλέψεις. Ο αλγόριθμος Machine Learning αναζητά ένα σύνολο κανόνων που του επιτρέπουν να συμπεράνει τα γενικά χαρακτηριστικά των στοιχείων μέσα σε μια ομάδα με στόχο την εφαρμογή της μάθησης σε παρόμοια στοιχεία. Όταν δίνεται στον υπολογιστή μια εντελώς νέα εικόνα, θα μπορεί να προβλέψει τη σωστή ετικέτα με βάση την «προηγουμένως αποκτηθείσα εμπειρία».

Η μηχανική μάθηση έχει γίνει δημοφιλής στην υγειονομική περίθαλψη για την ικανότητά της να βοηθά στην έγκαιρη διάγνωση ασθενειών, σχετικά γρήγορα και με ακρίβεια.

Στο πλαίσιο της υγειονομικής περίθαλψης, η μηχανική μάθηση μπορεί να χρησιμοποιηθεί για να βοηθήσει στη διάγνωση της νόσου. Χρησιμοποιείται ευρέως για να βοηθήσει στην εξέταση πολλών και διαφόρων παθήσεων. Αυτή η κατάσταση προσεγγίζεται χρησιμοποιώντας την ταξινόμηση της εποπτευόμενης μάθησης, καθώς αυτό που θέλουμε να γνωρίζουμε είναι εάν υπάρχει η πάθηση ή όχι (μια διακριτή δυαδική ετικέτα). Επιπλέον ως προστιθέμενη αξία, είναι δυνατό για τον αλγόριθμο να δηλώσει γιατί ταξινόμησε μια περίπτωση με τον τρόπο που έκανε, δημιουργώντας πολύτιμη γνώση για τους ειδικούς της υγείας. Προφανώς μια τέτοια τεχνολογία θα έχει τεράστιο αντίκτυπο στην ανθρώπινη υγεία. Θα πρέπει να σημειωθεί, ωστόσο, ότι η μηχανική μάθηση μπορεί να βοηθήσει στον εντοπισμό της πάθησης, αλλά η φροντίδα και η θεραπεία που λαμβάνει ένα άτομο καθορίζονται από τον γιατρό.

Σκοπός της εργασίας είναι η συμβολή στη θεμελίωση της βάσης για την ολοένα και εντονότερη χρήση της μηχανικής μάθησης σε υγειονομικά αλλά και σε διάφορα άλλα ζητήματα . Η μηχανική μάθηση αποτελεί ένα χρήσιμο εργαλείο που μπορεί να προσφέρει μία διαφορετική και ίσως πολύ πιο ωφέλιμη λύση σχετικά με τους συνήθεις τρόπους αντιμετώπισης ενός προβλήματος. Έτσι μπορεί να κερδηθεί χρόνος και εφόδια στην καθημερινότητα του ανθρώπου και κατά συνέπεια στη βελτίωση της ζωή του.

# ΚΕΦΑΛΑΙΟ 1: ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

### *1.1.1 Τι είναι;*

Η μηχανική μάθηση (ML) είναι ένα πεδίο έρευνας αφιερωμένο στην κατανόηση και τη δημιουργία μεθόδων που «μαθαίνουν», δηλαδή μεθόδους που αξιοποιούν δεδομένα για τη βελτίωση της απόδοσης σε κάποιο σύνολο εργασιών. Θεωρείται ως μέρος της τεχνητής νοημοσύνης. Οι αλγόριθμοι μηχανικής μάθησης χτίζουν ένα μοντέλο που βασίζεται σε δείγματα δεδομένων, γνωστά ως δεδομένα εκπαίδευσης, προκειμένου να κάνουν προβλέψεις ή αποφάσεις χωρίς να είναι ρητά προγραμματισμένοι να το κάνουν. Οι αλγόριθμοι μηχανικής μάθησης χρησιμοποιούνται σε μια μεγάλη ποικιλία εφαρμογών, όπως στην ιατρική, το φιλτράρισμα email, την αναγνώριση ομιλίας και την όραση υπολογιστή, όπου είναι δύσκολο ή ανέφικτο να αναπτυχθούν συμβατικοί αλγόριθμοι για την εκτέλεση των απαραίτητων εργασιών.

Ένα υποσύνολο της μηχανικής μάθησης σχετίζεται στενά με τις υπολογιστικές στατιστικές, οι οποίες επικεντρώνονται στην πραγματοποίηση προβλέψεων χρησιμοποιώντας υπολογιστές, αλλά δεν είναι όλη η μηχανική μάθηση στατιστική μάθηση. Η μελέτη της μαθηματικής βελτιστοποίησης παρέχει μεθόδους, θεωρίες και τομείς εφαρμογής στο πεδίο της μηχανικής μάθησης. Η εξόρυξη δεδομένων είναι ένα σχετικό πεδίο μελέτης, που εστιάζει στην διερευνητική ανάλυση δεδομένων μέσω της μάθησης χωρίς επίβλεψη. Ορισμένες υλοποιήσεις μηχανικής μάθησης χρησιμοποιούν δεδομένα και νευρωνικά δίκτυα με τρόπο που μιμείται τη λειτουργία ενός βιολογικού εγκεφάλου. Στην εφαρμογή της σε επιχειρηματικά προβλήματα, η μηχανική μάθηση αναφέρεται επίσης ως προγνωστική ανάλυση.

### *1.1.2 Πότε χρησιμοποιείται;*

Η Μηχανική μάθηση εφαρμόζεται σε μια σειρά από υπολογιστικές εργασίες, όπου τόσο ο σχεδιασμός όσο και ο ρητός προγραμματισμός των αλγορίθμων είναι ανέφικτος. Παραδείγματα εφαρμογών αποτελούν τα φίλτρα spam (spam filtering), η οπτική αναγνώριση χαρακτήρων (OCR), οι μηχανές αναζήτησης και η υπολογιστική όραση. Η Μηχανική μάθηση μερικές φορές συγχέεται με την εξόρυξη δεδομένων, όπου η τελευταία επικεντρώνεται περισσότερο στην εξερευνητική ανάλυση των δεδομένων, γνωστή και ως μη επιτηρούμενη μάθηση.

## *1.2 Είδη Μηχανικής Μάθησης*

Έχουν αναπτυχθεί πολλές τεχνικές μηχανικής μάθησης που χρησιμοποιούνται ανάλογα με τη φύση του προβλήματος και εμπίπτουν σε ένα από τα παρακάτω δυο είδη:

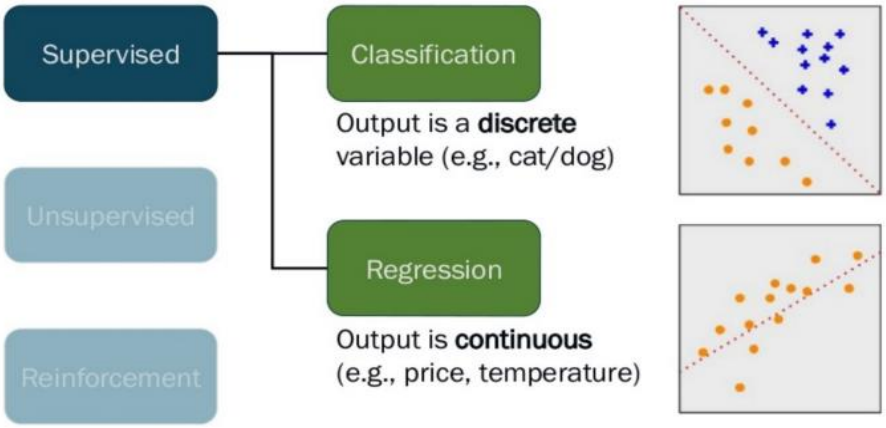
- μάθηση με επίβλεψη (supervised learning) ή μάθηση με παραδείγματα (learning from examples),

->Στη μάθηση με επίβλεψη το σύστημα καλείται να "μάθει" μια έννοια ή συνάρτηση από ένα σύνολο δεδομένων, η οποία αποτελεί περιγραφή ενός μοντέλου.

- μάθηση χωρίς επίβλεψη (unsupervised learning) ή μάθηση από παρατήρηση (learning from observation).

-> Στη μάθηση χωρίς επίβλεψη το σύστημα πρέπει μόνο του να ανακαλύψει συσχετίσεις ή ομάδες σε ένα σύνολο δεδομένων, δημιουργώντας πρότυπα, χωρίς να είναι γνωστό αν υπάρχουν, πόσα και ποια είναι.

### *1.2.1 Μάθηση με Επίβλεψη*




**Εικόνα 1: Μάθηση με επίβλεψη (Πηγή: https://www.congrelate.com/download-supervised-machine-learning-like-pics/ )**

Στη μάθηση με επίβλεψη το σύστημα πρέπει να "μάθει" επαγωγικά μια συνάρτηση που ονομάζεται συνάρτηση στόχος (target function) και αποτελεί έκφραση του μοντέλου που περιγράφει τα δεδομένα.

Η συνάρτηση στόχος χρησιμοποιείται για την πρόβλεψη της τιμής μιας μεταβλητής, που ονομάζεται εξαρτημένη μεταβλητή ή μεταβλητή εξόδου, βάσει των τιμών ενός συνόλου μεταβλητών, που ονομάζονται ανεξάρτητες μεταβλητές ή μεταβλητές εισόδου ή χαρακτηριστικά.

Η επαγωγική μάθηση στηρίζεται στην "υπόθεση επαγωγικής μάθησης" (inductive learning hypothesis), σύμφωνα με την οποία:

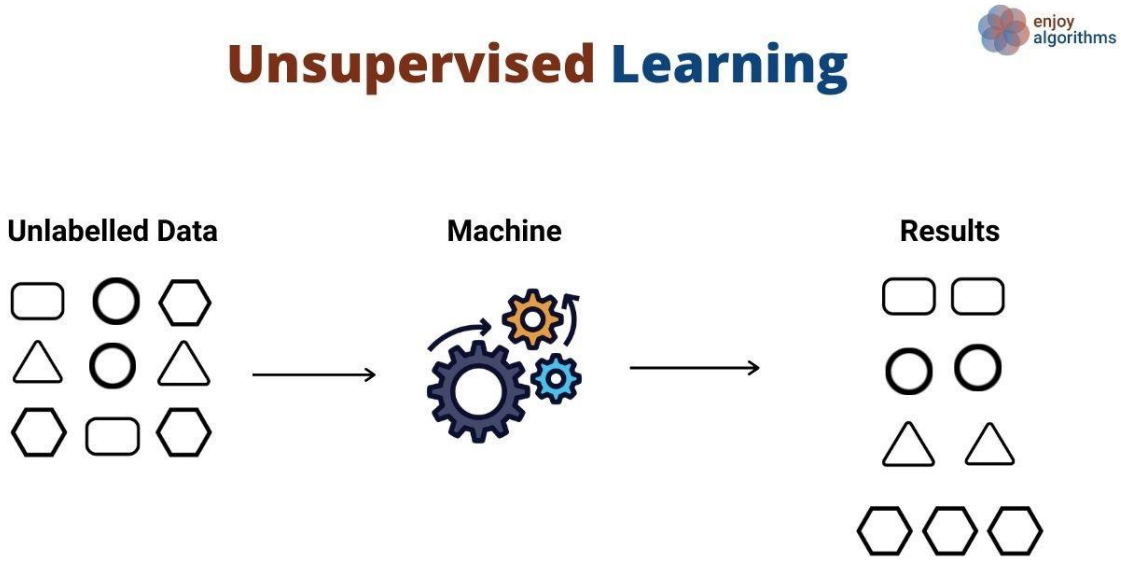
- Κάθε υπόθεση h που προσεγγίζει καλά τη συνάρτηση στόχο για ένα αρκετά μεγάλο σύνολο παραδειγμάτων, θα προσεγγίζει το ίδιο καλά τη συνάρτηση στόχο και για περιπτώσεις που δεν έχει εξετάσει.

Στην μάθηση με επίβλεψη διακρίνονται δυο είδη προβλημάτων (learning tasks), τα προβλήματα ταξινόμησης και τα προβλήματα παρεμβολής.

Η ταξινόμηση (classification) αφορά στη δημιουργία μοντέλων πρόβλεψης διακριτών τάξεων (κλάσεων/κατηγοριών) (π.χ. ομάδα αίματος).

Η παρεμβολή (regression) αφορά στη δημιουργία μοντέλων πρόβλεψης αριθμητικών τιμών (π.χ. πρόβλεψη ισοτιμίας νομισμάτων ή τιμής μετοχής).

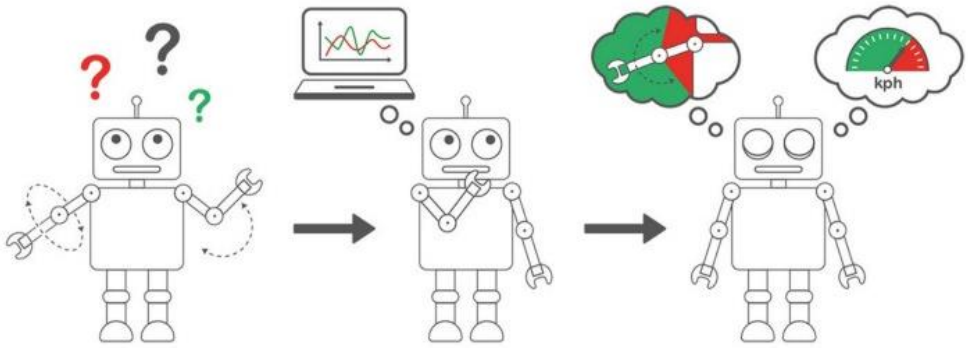
### *1.2.2 Μάθηση Χωρίς Επίβλεψη*




**Εικόνα 2: Μάθηση χωρίς επίβλεψη (Πηγή:https://www.congrelate.com/download-supervised-machine-learning-like-pics/ )**

Στη μάθηση χωρίς επίβλεψη το σύστημα έχει στόχο να ανακαλύψει συσχετίσεις και ομάδες από τα δεδομένα, βασιζόμενο μόνο στις ιδιότητές τους.
Σαν αποτέλεσμα προκύπτουν πρότυπα (περιγραφές), κάθε ένα από τα οποία περιγράφει ένα μέρος από τα δεδομένα.
Παραδείγματα προτύπων πληροφόρησης είναι
-οι κανόνες συσχέτισης (association rules) και
-οι ομάδες (clusters), οι οποίες προκύπτουν από τη διαδικασία της ομαδοποίησης (clustering).

### *1.3 Επιλογή κατάλληλου αλγόριθμου*




**Εικόνα 3: Επιλογή αλγορίθμου (Πηγή: https://towardsdatascience.com/a-journey-into-supervised-machine-learning-f26f238b0477)**

Για να επιλεγεί ο κατάλληλος αλγόριθμος μηχανικής μάθησης πρέπει να ληφθούν υπόψιν τα παρακάτω ζητήματα:

#### *1.3.1 Ταξινόμηση προβλήματος*

Σημαντικό βήμα είναι η κατηγοριοποίηση του προβλήματος.

- Κατηγοριοποίηση βάσει της εισαγωγής:

Εάν πρόκειται για δεδομένα με ετικέτα, πρόκειται για ένα εποπτευόμενο μαθησιακό πρόβλημα.
Εάν είναι δεδομένα χωρίς ετικέτα με σκοπό την εύρεση δομής, είναι ένα πρόβλημα μάθησης χωρίς επίβλεψη.
Εάν η λύση συνεπάγεται τη βελτιστοποίηση μιας αντικειμενικής συνάρτησης μέσω αλληλεπίδρασης με ένα περιβάλλον, είναι ένα πρόβλημα ενίσχυσης μάθησης.

- Κατηγοριοποίηση ανά έξοδο:

Εάν η έξοδος του μοντέλου είναι αριθμός, είναι πρόβλημα παλινδρόμησης.
Εάν η έξοδος του μοντέλου είναι μια κλάση, είναι πρόβλημα ταξινόμησης.
Εάν η έξοδος του μοντέλου είναι ένα σύνολο ομάδων εισόδου, είναι πρόβλημα ομαδοποίησης.

### *1.3.2 Κατανόηση των δεδομένων*

Τα δεδομένα από μόνα τους δεν είναι το τελικό παιχνίδι, αλλά μάλλον η πρώτη ύλη σε όλη τη διαδικασία ανάλυσης. Οι επιτυχημένες εταιρείες όχι μόνο καταγράφουν και έχουν πρόσβαση σε δεδομένα, αλλά είναι επίσης σε θέση να αντλούν γνώσεις που οδηγούν σε καλύτερες αποφάσεις, οι οποίες έχουν ως αποτέλεσμα καλύτερη εξυπηρέτηση πελατών, ανταγωνιστική διαφοροποίηση και υψηλότερη αύξηση εσόδων. Η διαδικασία κατανόησης των δεδομένων παίζει βασικό ρόλο στη διαδικασία επιλογής του σωστού αλγορίθμου για το σωστό πρόβλημα. Ορισμένοι αλγόριθμοι μπορούν να λειτουργήσουν με μικρότερα σύνολα δειγμάτων, ενώ άλλοι απαιτούν τόνους και τόνους δειγμάτων. Κάποιοι αλγόριθμοι λειτουργούν με κατηγορικά δεδομένα ενώ σε άλλους αρέσει να εργάζονται με αριθμητική εισαγωγή.

### *1.3.3 Ανάλυση των Δεδομένων*

Σε αυτό το βήμα, υπάρχουν δύο σημαντικές εργασίες που είναι η κατανόηση δεδομένων με περιγραφικές στατιστικές και η κατανόηση δεδομένων με οπτικοποίηση και γραφικές παραστάσεις.

- Επεξεργασία δεδομένων

Τα στοιχεία της επεξεργασίας δεδομένων περιλαμβάνουν την προεπεξεργασία, τη δημιουργία προφίλ, τον καθαρισμό, ενώ συχνά περιλαμβάνει επίσης τη συγκέντρωση δεδομένων από διαφορετικά εσωτερικά συστήματα και εξωτερικές πηγές.

- Μεταμόρφωση δεδομένων

Η παραδοσιακή ιδέα της μετατροπής δεδομένων από μια ακατέργαστη κατάσταση σε μια κατάσταση κατάλληλη για μοντελοποίηση είναι εκεί που ταιριάζει η μηχανική χαρακτηριστικών. Τα δεδομένα μετασχηματισμού και η μηχανική χαρακτηριστικών μπορεί, στην πραγματικότητα, να είναι συνώνυμα. Και εδώ είναι ένας ορισμός της τελευταίας έννοιας. Η μηχανική χαρακτηριστικών είναι η διαδικασία μετατροπής ακατέργαστων δεδομένων σε χαρακτηριστικά που αντιπροσωπεύουν καλύτερα το υποκείμενο πρόβλημα στα μοντέλα πρόβλεψης, με αποτέλεσμα τη βελτιωμένη ακρίβεια του μοντέλου σε αόρατα δεδομένα.

- Εύρεση των διαθέσιμων αλγόριθμων

Μετά την κατηγοριοποίηση του προβλήματος και την κατανόηση των δεδομένων, το επόμενο ορόσημο είναι ο εντοπισμός των αλγορίθμων που είναι εφαρμόσιμοι και πρακτικοί για εφαρμογή σε εύλογο χρονικό διάστημα.

Μερικά από τα στοιχεία που επηρεάζουν την επιλογή ενός μοντέλου είναι:

- Η ακρίβεια του μοντέλου.
- Η ερμηνευσιμότητα του μοντέλου.
- Η πολυπλοκότητα του μοντέλου.
- Η επεκτασιμότητα του μοντέλου.

- Πόσος χρόνος χρειάζεται για την κατασκευή, την εκπαίδευση και τη δοκιμή του μοντέλου;
- Πόσος χρόνος χρειάζεται για να γίνουν προβλέψεις χρησιμοποιώντας το μοντέλο
- Πληροί το μοντέλο τον επιχειρηματικό στόχο

### *1.4 Επεξεργασία Φυσικής Γλώσσας ή NLP (Natural Language Processing)*

Η επεξεργασία φυσικής γλώσσας (Natural Language Processing), δηλαδή η αποκρυπτογράφηση κειμένου και δεδομένων από μηχανές, έχει φέρει επανάσταση στην ανάλυση δεδομένων σε όλους τους κλάδους.

#### *1.4.1 Τι είναι η Επεξεργασία Φυσικής Γλώσσας;*

Η Επεξεργασία Φυσικής Γλώσσας είναι μια μορφή τεχνητής νοημοσύνης που δίνει στις μηχανές τη δυνατότητα όχι απλώς να διαβάζουν, αλλά να κατανοούν και να ερμηνεύουν την ανθρώπινη γλώσσα. Με το NLP, οι μηχανές μπορούν να βγάλουν νόημα από γραπτό ή προφορικό κείμενο και να εκτελέσουν εργασίες όπως η αναγνώριση ομιλίας, η ανάλυση συναισθημάτων και η αυτόματη σύνοψη κειμένου.

#### *1.4.2 Εφαρμογές Επεξεργασίας Φυσικής Γλώσσας*

Ορισμένες λύσεις που βασίζονται σε NLP περιλαμβάνουν μετάφραση, αναγνώριση ομιλίας, ανάλυση συναισθήματος, συστήματα ερωτήσεων/απαντήσεων, chatbots, αυτόματη σύνοψη κειμένου, ευφυΐα αγοράς, αυτόματη ταξινόμηση κειμένου και αυτόματο έλεγχο γραμματικής. Αυτές οι τεχνολογίες βοηθούν τους οργανισμούς να αναλύουν δεδομένα, να ανακαλύπτουν πληροφορίες, να αυτοματοποιούν χρονοβόρες διαδικασίες ή/και να αποκτούν ανταγωνιστικά πλεονεκτήματα.

**Μετάφραση**

Η μετάφραση γλωσσών είναι πιο περίπλοκη από μια απλή μέθοδο αντικατάστασης λέξης προς λέξη. Δεδομένου ότι κάθε γλώσσα έχει γραμματικούς κανόνες, η πρόκληση της μετάφρασης ενός κειμένου είναι να το κάνετε γίνει χωρίς να αλλάξει το νόημα και το στυλ του. Εφόσον οι υπολογιστές δεν καταλαβαίνουν τη γραμματική, χρειάζονται μια διαδικασία κατά την οποία μπορούν να αποδομήσουν μια πρόταση και στη συνέχεια να την ανακατασκευάσουν σε άλλη γλώσσα με τρόπο που να έχει νόημα.

Το Google Translate είναι ένα από τα πιο γνωστά διαδικτυακά εργαλεία μετάφρασης. Η Μετάφραση Google χρησιμοποιούσε κάποτε τη Μηχανική Μετάφραση βάσει Φράσεων (PBMT), η οποία αναζητά παρόμοιες φράσεις μεταξύ διαφορετικών γλωσσών. Προς το παρόν, η Google χρησιμοποιεί τη Μετάφραση Νευρωνικής Μηχανής Google (GNMT), η οποία χρησιμοποιεί ML με NLP για να αναζητήσει μοτίβα στις γλώσσες.

### Αναγνώρισης ομιλίας

Η αναγνώριση ομιλίας είναι η ικανότητα ενός μηχανήματος να αναγνωρίζει και να ερμηνεύει φράσεις και λέξεις από την προφορική γλώσσα και να τις μετατρέπει σε μορφή αναγνώσιμη από μηχανή. Χρησιμοποιεί το NLP για να επιτρέπει στους υπολογιστές να προσομοιώνουν την ανθρώπινη αλληλεπίδραση και το ML για να ανταποκρίνεται με τρόπο που μιμείται τις ανθρώπινες αποκρίσεις.

Το Google Now, η Alexa και το Siri είναι μερικά από τα πιο δημοφιλή παραδείγματα αναγνώρισης ομιλίας. Απλώς λέγοντας «καλέστε την Μαρία», μια κινητή συσκευή αναγνωρίζει τι σημαίνει αυτή η εντολή και θα πραγματοποιήσει τώρα μια κλήση στην επαφή που έχει αποθηκευτεί ως Μαρία.

### Ανάλυση Συναισθήματος

Η ανάλυση συναισθήματος χρησιμοποιεί NLP και ML για να ερμηνεύσει και να αναλύσει τα συναισθήματα σε υποκειμενικά δεδομένα, όπως άρθρα ειδήσεων και tweets. Μπορούν να εντοπιστούν θετικές, αρνητικές και ουδέτερες απόψεις για να προσδιοριστεί το συναίσθημα ενός πελάτη για μια επωνυμία, ένα προϊόν ή μια υπηρεσία. Η ανάλυση συναισθήματος χρησιμοποιείται για τη μέτρηση της κοινής γνώμης, την παρακολούθηση της φήμης της επωνυμίας και την καλύτερη κατανόηση των εμπειριών των πελατών.

Το χρηματιστήριο είναι ένα ευαίσθητο πεδίο που μπορεί να επηρεαστεί σε μεγάλο βαθμό από τα ανθρώπινα συναισθήματα. Το αρνητικό κλίμα μπορεί να οδηγήσει σε πτώση των τιμών των μετοχών, ενώ το θετικό κλίμα μπορεί να προκαλέσει την αγορά περισσότερων μετοχών της εταιρείας, προκαλώντας αύξηση των τιμών των μετοχών.

### Chatbots

Τα chatbots είναι προγράμματα που χρησιμοποιούνται για την παροχή αυτοματοποιημένων απαντήσεων σε κοινά ερωτήματα πελατών. Έχουν συστήματα αναγνώρισης προτύπων με ευρετικές αποκρίσεις, τα οποία χρησιμοποιούνται για τη διεξαγωγή συνομιλιών με ανθρώπους. Αρχικά, τα chatbots χρησιμοποιήθηκαν για να απαντήσουν σε βασικές ερωτήσεις για να ανακουφίσουν τα μεγάλα τηλεφωνικά κέντρα και να προσφέρουν γρήγορες υπηρεσίες υποστήριξης πελατών.

Ωστόσο, τα chatbot που λειτουργούν με τεχνητή νοημοσύνη έχουν σχεδιαστεί για να χειρίζονται πιο περίπλοκα αιτήματα, καθιστώντας τις εμπειρίες συνομιλίας όλο και πιο διαισθητικές. Τα chatbots στην υγειονομική περίθαλψη, για παράδειγμα, μπορούν να συλλέξουν δεδομένα πρόσληψης, να βοηθήσουν τους ασθενείς να αξιολογήσουν τα συμπτώματά τους και να καθορίσουν τα επόμενα βήματα. Αυτά τα chatbots μπορούν να κλείσουν ραντεβού με τον κατάλληλο γιατρό και ακόμη και να προτείνουν θεραπείες.

## Συστήματα Ερωτήσεων – Απαντήσεων

Τα συστήματα ερωτήσεων και απαντήσεων είναι έξυπνα συστήματα που χρησιμοποιούνται για την παροχή απαντήσεων σε ερωτήματα πελατών. Εκτός από τα chatbot, τα συστήματα ερωτήσεων-απαντήσεων έχουν μια τεράστια γκάμα γνώσεων και καλή κατανόηση της γλώσσας αντί για τυποποιημένες απαντήσεις. Μπορούν να απαντήσουν σε ερωτήσεις όπως «Πότε δολοφονήθηκε ο Αβραάμ Λίνκολν;» ή "Πώς θα πάω στο αεροδρόμιο;" και μπορεί να δημιουργηθεί για την αντιμετώπιση δεδομένων κειμένου, ήχου, εικόνων και βίντεο.

Τα συστήματα ερωτήσεων απαντήσεων μπορούν να βρεθούν σε συνομιλίες μέσων κοινωνικής δικτύωσης και εργαλεία όπως το Siri και το Watson της IBM. Το 2011, ο υπολογιστής Watson της IBM διαγωνίστηκε στο Jeopardy, μια παράσταση παιχνιδιού κατά την οποία δίνονται πρώτα οι απαντήσεις και οι διαγωνιζόμενοι δίνουν τις ερωτήσεις. Ο υπολογιστής ανταγωνίστηκε τους δύο μεγαλύτερους πρωταθλητές όλων των εποχών της σειράς και κατέπληξε τη βιομηχανία της τεχνολογίας όταν κέρδισε την πρώτη θέση.

## Αυτόματη σύνοψη κειμένου

Η αυτόματη σύνοψη κειμένου είναι το έργο της συμπύκνωσης ενός τμήματος κειμένου σε μια συντομότερη έκδοση, εξάγοντας τις κύριες ιδέες του και διατηρώντας το νόημα του περιεχομένου. Αυτή η εφαρμογή του NLP χρησιμοποιείται σε τίτλους ειδήσεων, αποσπάσματα αποτελεσμάτων στην αναζήτηση ιστού και ενημερωτικά δελτία αναφορών αγοράς.

## Ευφυΐα Αγοράς

Το Market Intelligence (Ευφυΐα Αγοράς) είναι η συλλογή πολύτιμων γνώσεων γύρω από τις τάσεις, τους καταναλωτές, τα προϊόντα και τους ανταγωνιστές για την εξαγωγή πληροφοριών που μπορούν να χρησιμοποιηθούν για τη λήψη στρατηγικών αποφάσεων. Το Market Intelligence μπορεί να αναλύσει θέματα, συναισθήματα, λέξεις-κλειδιά και πρόθεση σε μη δομημένα δεδομένα και είναι λιγότερο χρονοβόρα από την παραδοσιακή έρευνα γραφείου.

Χρησιμοποιώντας το Market Intelligence, οι οργανισμοί μπορούν να ενημερωθούν για ερωτήματα αναζήτησης και να προσθέσουν συνώνυμα σχετικά με τα συμφραζόμενα στα αποτελέσματα αναζήτησης. Μπορεί επίσης να βοηθήσει τους οργανισμούς να αποφασίσουν ποια προϊόντα ή υπηρεσίες θα διακόψουν ή ποιους πελάτες θα στοχεύσουν.

## Αυτόματη ταξινόμηση κειμένου

Η αυτόματη ταξινόμηση κειμένου είναι μια άλλη θεμελιώδης λύση του NLP. Είναι η διαδικασία αντιστοίχισης ετικετών στο κείμενο σύμφωνα με το περιεχόμενο και τη σημασιολογία του, η οποία επιτρέπει τη γρήγορη και εύκολη ανάκτηση πληροφοριών στη φάση αναζήτησης. Αυτή η εφαρμογή NLP μπορεί να διαφοροποιήσει τα ανεπιθύμητα από τα μη ανεπιθύμητα με βάση το περιεχόμενό του.

**Αυτόματος γραμματικός έλεγχος**

Ο αυτόματος γραμματικός έλεγχος, η εργασία ανίχνευσης και διόρθωσης γραμματικών λαθών και ορθογραφικών λαθών στο κείμενο ανάλογα με το περιβάλλον, είναι ένα άλλο σημαντικό μέρος του NLP. Ο Αυτόματος Έλεγχος Γραμματικής θα σας ειδοποιήσει για ένα πιθανό σφάλμα υπογραμμίζοντας τη λέξη με κόκκινο.

**Πλεονεκτήματα και μειονεκτήματα της Επεξεργασίας Φυσικής Γλώσσας**
Όπως πολλές άλλες μορφές Τεχνητής Νοημοσύνης, η χρήση της Επεξεργασίας Φυσικής Γλώσσας έχει πλεονεκτήματα αλλά και μειονεκτήματα.

**Τα πλεονεκτήματα του NLP περιλαμβάνουν:**

- Μόλις εφαρμοστεί, η χρήση του NLP είναι λιγότερο δαπανηρή και πιο χρονικά αποδοτική από την απασχόληση ενός ατόμου.
- Το NLP μπορεί επίσης να βοηθήσει τις επιχειρήσεις να προσφέρουν ταχύτερους χρόνους απόκρισης εξυπηρέτησης πελατών. Ανεξάρτητα από την ώρα της ημέρας ή την ημέρα της εβδομάδας, οι πελάτες λαμβάνουν άμεσες απαντήσεις στις ερωτήσεις τους.
- Τα προεκπαιδευμένα μοντέλα μηχανικής εκμάθησης είναι ευρέως διαθέσιμα για προγραμματιστές για τη διευκόλυνση διαφορετικών εφαρμογών του NLP, καθιστώντας τα εύκολα στην εφαρμογή τους.

Οι εξελίξεις στο NLP είναι πολλά υποσχόμενες, αλλά υπάρχουν και ορισμένα μειονεκτήματα στο NLP.

**Τα μειονεκτήματα του NLP περιλαμβάνουν:**

- Η εκπαίδευση μπορεί να είναι χρονοβόρα. Εάν ένα νέο μοντέλο πρέπει να αναπτυχθεί χωρίς τη χρήση προεκπαιδευμένου μοντέλου, μπορεί να χρειαστούν εβδομάδες μέχρι να επιτευχθεί υψηλό επίπεδο απόδοσης.
- Ένα άλλο μειονέκτημα του NLP είναι ότι η ML δεν είναι 100 τοις εκατό αξιόπιστη. Υπάρχει πάντα η πιθανότητα σφαλμάτων σε προβλέψεις και αποτελέσματα που πρέπει να ληφθούν υπόψη.

# ΚΕΦΑΛΑΙΟ 2: ΤΑΞΙΝΟΜΗΤΕΣ

## *2.1 Τι είναι η ταξινόμηση;*

Ταξινόμηση είναι η διαδικασία πρόβλεψης της κατηγορίας των δεδομένων σημείων. Οι κλάσεις μερικές φορές ονομάζονται στόχοι/ετικέτες ή κατηγορίες. Η προγνωστική μοντελοποίηση ταξινόμησης είναι το έργο της προσέγγισης μιας συνάρτησης αντιστοίχισης (f) από τις μεταβλητές εισόδου (X) στις διακριτές μεταβλητές εξόδου (y).

Για παράδειγμα, ο εντοπισμός ανεπιθύμητης αλληλογραφίας σε παρόχους υπηρεσιών email μπορεί να αναγνωριστεί ως πρόβλημα ταξινόμησης. Αυτή είναι η δυαδική ταξινόμηση αφού υπάρχουν μόνο 2 κατηγορίες ως ανεπιθύμητα και όχι ανεπιθύμητα. Ένας ταξινομητής χρησιμοποιεί ορισμένα δεδομένα εκπαίδευσης για να κατανοήσει πώς οι δεδομένες μεταβλητές εισόδου σχετίζονται με την κλάση. Σε αυτήν την περίπτωση, τα γνωστά ανεπιθύμητα και μη ανεπιθύμητα μηνύματα ηλεκτρονικού ταχυδρομείου πρέπει να χρησιμοποιούνται ως δεδομένα εκπαίδευσης. Όταν ο ταξινομητής εκπαιδεύεται με ακρίβεια, μπορεί να χρησιμοποιηθεί για τον εντοπισμό ενός άγνωστου email.

Η ταξινόμηση ανήκει στην κατηγορία της εποπτευόμενης μάθησης όπου οι στόχοι παρέχουν και τα δεδομένα εισόδου. Υπάρχουν πολλές εφαρμογές στην ταξινόμηση σε πολλούς τομείς όπως στην έγκριση πιστώσεων, την ιατρική διάγνωση, το μάρκετινγκ στόχων κ.λπ.

Υπάρχουν δύο τύποι μαθητών στην ταξινόμηση ως τεμπέληδες (lazy) και πρόθυμοι (eager) μαθητές (learners).

1. Lazy learners

Οι τεμπέληδες μαθητές (lazy learners) απλώς αποθηκεύουν τα δεδομένα εκπαίδευσης και περιμένουν μέχρι να εμφανιστούν τα δεδομένα δοκιμής. Όταν συμβεί αυτό, η ταξινόμηση πραγματοποιείται με βάση τα πιο σχετικά δεδομένα στα αποθηκευμένα δεδομένα εκπαίδευσης. Σε σύγκριση με τους πρόθυμους μαθητές, οι τεμπέληδες μαθητές έχουν λιγότερο χρόνο εκπαίδευσης αλλά περισσότερο χρόνο στην πρόβλεψη.

Πχ. k-πλησιέστερος γείτονας, Συλλογισμός που βασίζεται σε περίπτωση

2. Eager Learners

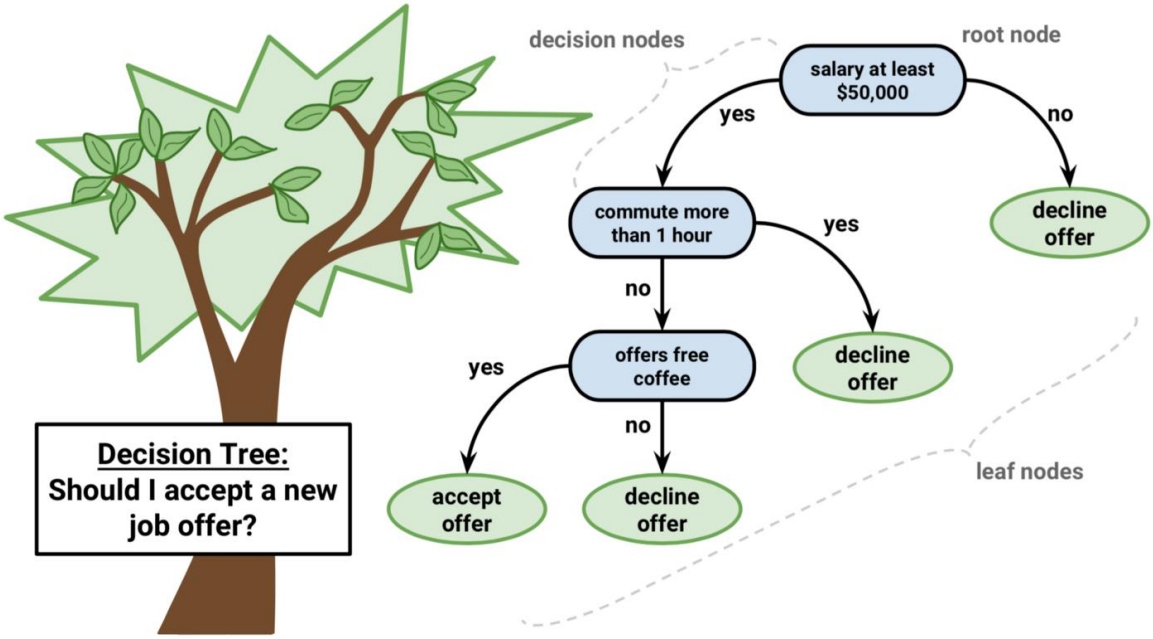
Οι πρόθυμοι μαθητές (eager learners) κατασκευάζουν ένα μοντέλο ταξινόμησης με βάση τα δεδομένα εκπαίδευσης πριν λάβουν δεδομένα για ταξινόμηση. Πρέπει να είναι σε θέση να δεσμευτεί σε μια ενιαία υπόθεση που καλύπτει ολόκληρο τον χώρο του στιγμιότυπου. Λόγω της κατασκευής του μοντέλου, οι πρόθυμοι μαθητές χρειάζονται πολύ χρόνο για να εκπαιδευτούν και λιγότερο χρόνο για να προβλέψουν.

Πχ. Decision Tree, Naive Bayes, Τεχνητά Νευρωνικά Δίκτυα

## *2.2 Αλγόριθμοι ταξινόμησης*

Υπάρχουν πολλοί αλγόριθμοι ταξινόμησης διαθέσιμοι τώρα, αλλά δεν είναι δυνατόν να συμπεράνουμε ποιος είναι ανώτερος από τον άλλο. Εξαρτάται από την εφαρμογή και τη φύση του διαθέσιμου συνόλου δεδομένων. Παρακάτω αναφέρονται οι βασικότεροι αλγόριθμοι ταξινόμησης.

### *2.2.1 Decision Tree (Δέντρο απόφασης)*




**Εικόνα 4: Δέντρο απόφασης (Πηγή: https://www.devops.ae/decision-tree-classification-algorithm/)**

Το δέντρο απόφασης δημιουργεί μοντέλα ταξινόμησης ή παλινδρόμησης με τη μορφή δομής δέντρου. Χρησιμοποιεί ένα σύνολο κανόνων αν-τότε που είναι αμοιβαία αποκλειστικό και εξαντλητικό για ταξινόμηση. Οι κανόνες μαθαίνονται διαδοχικά χρησιμοποιώντας τα δεδομένα εκπαίδευσης ένα κάθε φορά. Κάθε φορά που μαθαίνεται ένας κανόνας, αφαιρούνται οι πλειάδες που καλύπτονται από τους κανόνες. Αυτή η διαδικασία συνεχίζεται στο σετ εκπαίδευσης μέχρι να εκπληρωθεί ένας όρος τερματισμού.

Το δέντρο κατασκευάζεται με αναδρομικό από πάνω προς τα κάτω, διαίρει και βασίλευε. Όλα τα χαρακτηριστικά πρέπει να είναι κατηγορηματικά. Διαφορετικά, θα πρέπει να διακριτοποιηθούν εκ των προτέρων. Τα χαρακτηριστικά στην κορυφή του δέντρου έχουν μεγαλύτερη επίδραση στην ταξινόμηση και προσδιορίζονται χρησιμοποιώντας την έννοια του κέρδους πληροφοριών.

Ένα δέντρο απόφασης μπορεί εύκολα να υπερ-τοποθετηθεί δημιουργώντας πάρα πολλά κλαδιά και μπορεί να αντανακλά ανωμαλίες λόγω θορύβου ή ακραίων τιμών. Ένα υπερ-προσαρμοσμένο μοντέλο έχει πολύ κακή απόδοση στα αόρατα δεδομένα, παρόλο που δίνει εντυπωσιακή απόδοση στα δεδομένα προπόνησης. Αυτό μπορεί να αποφευχθεί με το προ-κλάδεμα που σταματά την κατασκευή δέντρων νωρίς ή μετά το κλάδεμα που αφαιρεί τα κλαδιά από το πλήρως αναπτυγμένο δέντρο.

**Γιατί Αλγόριθμος Δέντρου Αποφάσεων;**

Το Decision Tree θεωρείται ένας από τους πιο χρήσιμους αλγόριθμους Machine Learning, καθώς μπορεί να χρησιμοποιηθεί για την επίλυση ποικίλων προβλημάτων. Ακολουθούν μερικοί λόγοι για τους οποίους πρέπει να χρησιμοποιηθεί το Δέντρο αποφάσεων:

1. Θεωρείται ο πιο κατανοητός αλγόριθμος Machine Learning και μπορεί εύκολα να ερμηνευτεί.
2. Μπορεί να χρησιμοποιηθεί για προβλήματα ταξινόμησης και παλινδρόμησης.
3. Σε αντίθεση με τους περισσότερους αλγόριθμους Machine Learning, λειτουργεί αποτελεσματικά με μη γραμμικά δεδομένα.
4. Η κατασκευή ενός δέντρου αποφάσεων είναι μια πολύ γρήγορη διαδικασία, καθώς χρησιμοποιεί μόνο ένα χαρακτηριστικό ανά κόμβο για να χωρίσει τα δεδομένα.

Ένα δέντρο απόφασης έχει την ακόλουθη δομή:

- **Ριζικός κόμβος**: Ο ριζικός κόμβος είναι το σημείο εκκίνησης ενός δέντρου. Σε αυτό το σημείο εκτελείται η πρώτη διαίρεση.

- **Εσωτερικοί κόμβοι**: Κάθε εσωτερικός κόμβος αντιπροσωπεύει ένα σημείο απόφασης (μεταβλητή πρόβλεψης) που τελικά οδηγεί στην πρόβλεψη του αποτελέσματος.

- **Κόμβοι Φύλλων/Τερματικών**: Οι κόμβοι φύλλων αντιπροσωπεύουν την τελική κατηγορία του αποτελέσματος και επομένως ονομάζονται και τερματικοί κόμβοι.

- **Διακλαδώσεις**: Οι κλάδοι είναι συνδέσεις μεταξύ κόμβων, αντιπροσωπεύονται ως βέλη. Κάθε κλάδος αντιπροσωπεύει μια απάντηση όπως ναι ή όχι.

**Πώς λειτουργεί ο αλγόριθμος του δέντρου αποφάσεων;**
Ο αλγόριθμος ακολουθεί τα παρακάτω βήματα:

**Βήμα 1:** Επιλογή του χαρακτηριστικού (μεταβλητή πρόβλεψης) που ταξινομεί καλύτερα το σύνολο δεδομένων στις επιθυμητές κλάσεις και εκχώρηση αυτού του χαρακτηριστικού στον ριζικό κόμβο.

**Βήμα 2**: Πέρασμα προς τα κάτω από τον ριζικό κόμβο, ενώ λαμβάνονται σχετικές αποφάσεις σε κάθε εσωτερικό κόμβο, έτσι ώστε κάθε εσωτερικός κόμβος να ταξινομεί καλύτερα τα δεδομένα.

**Βήμα 3**: Ξανά πίσω στο βήμα 1 και επανάληψη μέχρι να αντιστοιχιστεί μια κλάση στα δεδομένα εισόδου.

Τα προαναφερθέντα βήματα αντιπροσωπεύουν τη γενική ροή εργασίας ενός Δέντρου Αποφάσεων που χρησιμοποιείται για σκοπούς ταξινόμησης.

### *2.2.2 Naive Bayes*

Ο Naive Bayes είναι ένας πιθανολογικός ταξινομητής εμπνευσμένος από το θεώρημα Bayes με μια απλή υπόθεση ότι τα χαρακτηριστικά είναι υπό όρους ανεξάρτητα.

$$P(\mathbf{X} \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i) = P(x_1 \mid C_i) \times P(x_2 \mid C_i) \times \ldots \times P(x_n \mid C_i)$$

**Εικόνα 5: Εξίσωση Naive Bayes (Πηγή: https://towardsdatascience.com/introduction-to-na%C3%AFve-bayes-classifier-fa59e3e24aaf)**

Η ταξινόμηση πραγματοποιείται με την εξαγωγή του μέγιστου οπισθίου που είναι το μέγιστο P(Ci|X) με την παραπάνω υπόθεση να ισχύει για το θεώρημα Bayes. Αυτή η υπόθεση μειώνει σημαντικά το υπολογιστικό κόστος μετρώντας μόνο την κατανομή κλάσης. Παρόλο που η υπόθεση δεν ισχύει στις περισσότερες περιπτώσεις, καθώς τα χαρακτηριστικά εξαρτώνται, παραδόξως ο Naive Bayes κατάφερε να αποδώσει εντυπωσιακά.

Ο Naive Bayes είναι ένας πολύ απλός αλγόριθμος στην εφαρμογή του και έχει επιτύχει καλά αποτελέσματα στις περισσότερες περιπτώσεις. Μπορεί εύκολα να κλιμακωθεί σε μεγαλύτερα σύνολα δεδομένων, καθώς απαιτεί γραμμικό χρόνο, αντί με ακριβή επαναληπτική προσέγγιση όπως χρησιμοποιείται για πολλούς άλλους τύπους ταξινομητών.

Ο Naive Bayes μπορεί να υποφέρει από ένα πρόβλημα που ονομάζεται πρόβλημα μηδενικής πιθανότητας. Όταν η υπό όρους πιθανότητα είναι μηδέν για ένα συγκεκριμένο χαρακτηριστικό, αποτυγχάνει να δώσει μια έγκυρη πρόβλεψη. Αυτό πρέπει να διορθωθεί ρητά χρησιμοποιώντας έναν εκτιμητή Laplace.

**Τι είναι ο Naive Bayes;**

Ο Naive Bayes είναι ένας από τους πιο απλούς και ισχυρούς αλγόριθμους ταξινόμησης με βάση το θεώρημα Bayes με μια υπόθεση ανεξαρτησίας μεταξύ των προγνωστικών. Το μοντέλο Naive Bayes είναι εύκολο στην κατασκευή και ιδιαίτερα χρήσιμο για πολύ μεγάλα σύνολα δεδομένων. Υπάρχουν δύο μέρη σε αυτόν τον αλγόριθμο:

- Naïve
- Bayes

Ο ταξινομητής Naive Bayes υποθέτει ότι η παρουσία ενός χαρακτηριστικού σε μια κλάση δεν σχετίζεται με οποιοδήποτε άλλο χαρακτηριστικό. Ακόμα κι αν αυτά τα χαρακτηριστικά εξαρτώνται το ένα από το άλλο ή από την ύπαρξη των άλλων χαρακτηριστικών, όλες αυτές οι ιδιότητες συμβάλλουν ανεξάρτητα στην πιθανότητα ότι ένα συγκεκριμένο φρούτο είναι ένα μήλο ή ένα πορτοκάλι ή μια μπανάνα και γι' αυτό είναι γνωστό ως "Naive(Αφελής)".

**Τι είναι το θεώρημα Bayes;**

Στη Στατιστική και τη θεωρία πιθανοτήτων, το θεώρημα του Bayes περιγράφει την πιθανότητα ενός γεγονότος, με βάση την προηγούμενη γνώση των συνθηκών που μπορεί να σχετίζονται με το γεγονός. Χρησιμεύει ως τρόπος να υπολογιστεί η υπό όρους πιθανότητα.

Με δεδομένη την υπόθεση H και την απόδειξη E, το θεώρημα του Bayes δηλώνει ότι η σχέση μεταξύ της πιθανότητας της Υπόθεσης πριν από τη λήψη της απόδειξης P(H) και της πιθανότητας της υπόθεσης μετά τη λήψη της απόδειξης P(H|E) είναι:

$$P(H|E) = \frac{P(E|H).P(H)}{P(E)}$$

**Εικόνα 6: P(H|E) (Πηγή: https://towardsdatascience.com/introduction-to-na%C3%AFve-bayes-classifier-fa59e3e24aaf)**

Αυτό συσχετίζει την πιθανότητα της υπόθεσης πριν από τη λήψη της απόδειξης P(H), με την πιθανότητα της υπόθεσης μετά τη λήψη της απόδειξης, P(H|E). Για το λόγο αυτό, ονομάζεται προηγούμενη πιθανότητα, ενώ η P(H|E) ονομάζεται οπίσθια πιθανότητα. Ο παράγοντας που συσχετίζει τα δύο, P(H|E) / P(E), ονομάζεται λόγος πιθανότητας. Χρησιμοποιώντας αυτούς τους όρους, το θεώρημα του Bayes μπορεί να αναδιατυπωθεί ως εξής:

"Η μεταγενέστερη πιθανότητα ισούται με την προηγούμενη πιθανότητα επί την αναλογία πιθανότητας."

**Πλεονεκτήματα και μειονεκτήματα**

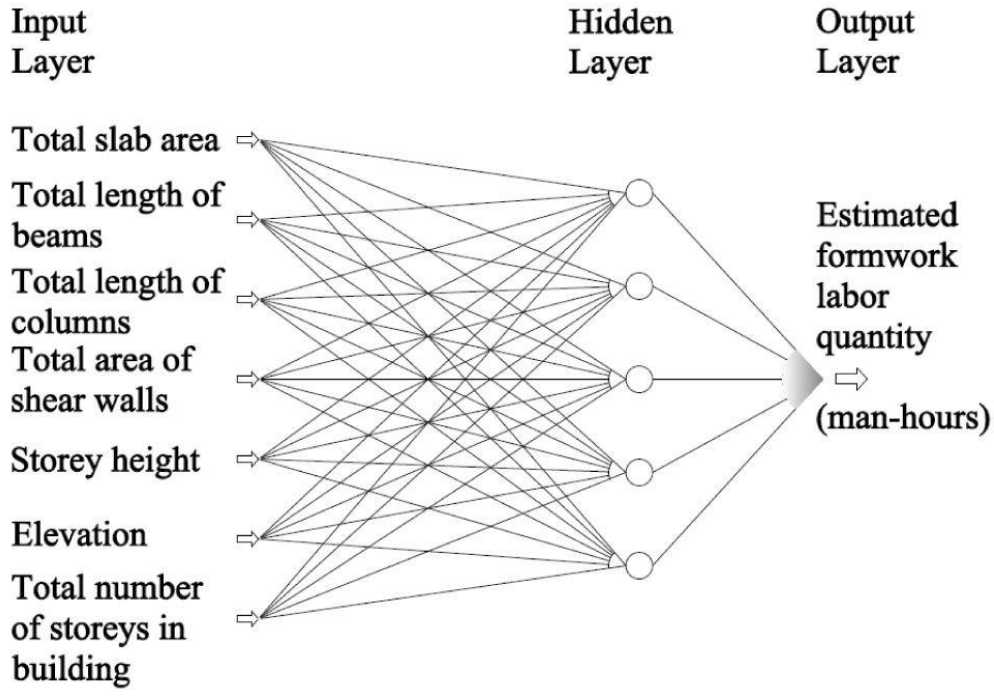
Ο ταξινομητής Naive Bayes απαιτεί μια μικρή ποσότητα δεδομένων εκπαίδευσης για να εκτιμήσει τις απαραίτητες παραμέτρους για να ληφθούν τα αποτελέσματα. Είναι εξαιρετικά γρήγοροι στη φύση τους σε σύγκριση με άλλους ταξινομητές.

Το μόνο μειονέκτημα είναι ότι είναι γνωστό ότι είναι κακός εκτιμητής.

**Περιπτώσεις χρήσης**

- Προβλέψεις ασθενειών
- Ταξινόμηση Εγγράφων
- Φίλτρα ανεπιθύμητης αλληλογραφίας
- Ανάλυση Συναισθήματος

### *2.2.3 Artificial Neural Networks(Τεχνητά Νευρωνικά Δίκτυα)*




**Εικόνα 7: Νευρωνικό δίκτυο (Πηγή: https://today.duke.edu/2020/12/accurate-neural-network-computer-vision-without-black-box)**

Το Τεχνητό Νευρωνικό Δίκτυο είναι ένα σύνολο συνδεδεμένων μονάδων εισόδου/εξόδου, όπου κάθε σύνδεση έχει ένα βάρος που σχετίζεται με αυτήν που ξεκίνησε από ψυχολόγους και νευροβιολόγους για την ανάπτυξη και τη δοκιμή υπολογιστικών αναλόγων νευρώνων. Κατά τη φάση εκμάθησης, το δίκτυο μαθαίνει προσαρμόζοντας τα βάρη έτσι ώστε να μπορεί να προβλέψει τη σωστή ετικέτα κλάσης των πλειάδων εισόδου.

Υπάρχουν πολλές αρχιτεκτονικές δικτύου διαθέσιμες τώρα, όπως Feed-forward, Convolutional, Recurrent κ.λπ. Η κατάλληλη αρχιτεκτονική εξαρτάται από την εφαρμογή του μοντέλου. Για τις περισσότερες περιπτώσεις, τα μοντέλα feed-forward δίνουν εύλογα ακριβή αποτελέσματα και ειδικά για εφαρμογές επεξεργασίας εικόνας, τα συνελικτικά δίκτυα αποδίδουν καλύτερα.

Μπορεί να υπάρχουν πολλά κρυφά επίπεδα στο μοντέλο ανάλογα με την πολυπλοκότητα της συνάρτησης που πρόκειται να αντιστοιχιστεί από το μοντέλο. Η ύπαρξη περισσότερων κρυφών επιπέδων θα επιτρέψει τη μοντελοποίηση πολύπλοκων σχέσεων, όπως τα βαθιά νευρωνικά δίκτυα.

Ωστόσο, όταν υπάρχουν πολλά κρυφά στρώματα, χρειάζεται πολύς χρόνος για την εκπαίδευση και την προσαρμογή των βαρών. Το άλλο μειονέκτημα είναι η κακή ερμηνεία του μοντέλου σε σύγκριση με άλλα μοντέλα όπως το Decision Trees λόγω της άγνωστης συμβολικής σημασίας πίσω από τα μαθημένα βάρη.

Όμως, τα τεχνητά νευρωνικά δίκτυα έχουν αποδώσει εντυπωσιακά στις περισσότερες εφαρμογές του πραγματικού κόσμου. Έχει υψηλή ανοχή σε θορυβώδη δεδομένα και μπορεί να ταξινομήσει μη εκπαιδευμένα μοτίβα. Συνήθως, τα τεχνητά νευρωνικά δίκτυα αποδίδουν καλύτερα με εισόδους και εξόδους συνεχούς αξίας.

Ένα νευρωνικό δίκτυο αποτελείται από τρία σημαντικά επίπεδα:

- **Input Layer(Επίπεδο εισόδου)**: Όπως υποδηλώνει το όνομα, αυτό το επίπεδο δέχεται όλες τις εισόδους που παρέχονται από τον προγραμματιστή.
- **Hidden Layer(Κρυμμένο Επίπεδο)**: Ανάμεσα στο επίπεδο εισόδου και εξόδου υπάρχει ένα σύνολο επιπέδων γνωστών ως Hidden layers. Σε αυτό το επίπεδο, εκτελούνται υπολογισμοί που καταλήγουν στην έξοδο.
- **Output Layer(Επίπεδο εξόδου)**: Οι είσοδοι περνούν από μια σειρά μετασχηματισμών μέσω του κρυφού στρώματος που τελικά καταλήγει στην έξοδο που παραδίδεται μέσω αυτού του επιπέδου.

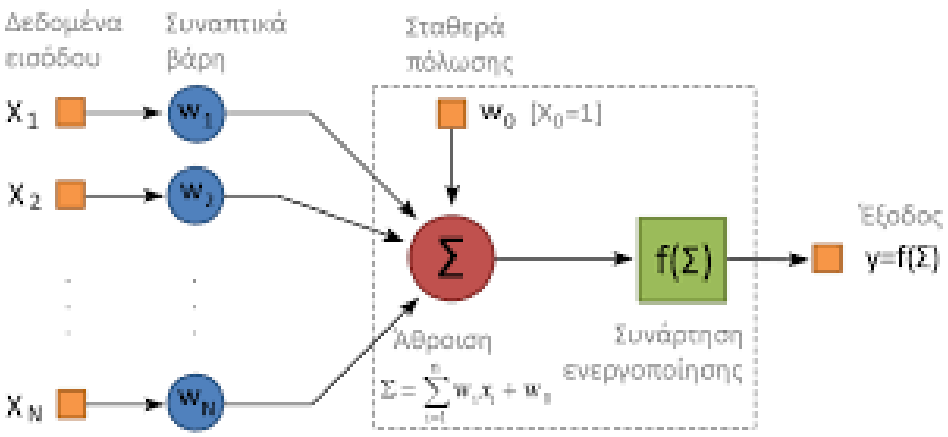
**Πώς λειτουργεί ένα νευρωνικό δίκτυο;**

Για να κατανοηθούν τα νευρωνικά δίκτυα, πρέπει να τα αναλυθεί και να κατανοηθεί η πιο βασική μονάδα ενός νευρωνικού δικτύου, δηλαδή το Perceptron.

**Τι είναι το Perceptron;**

Το Perceptron είναι ένα νευρωνικό δίκτυο ενός επιπέδου που χρησιμοποιείται για την ταξινόμηση γραμμικών δεδομένων. Έχει 4 σημαντικά συστατικά:

1. Εισροές
2. Βάρη και Πόλωση
3. Συνάρτηση άθροισης
4. Συνάρτηση ενεργοποίησης ή μετασχηματισμού




**Εικόνα 8: Perceptron (Πηγή: https://www.educba.com/single-layer-perceptron/)**

Η βασική λογική πίσω από ένα Perceptron είναι η εξής:
Οι είσοδοι (x) που λαμβάνονται από το επίπεδο εισόδου πολλαπλασιάζονται με τα βάρη w. Στη συνέχεια, οι πολλαπλασιασμένες τιμές προστίθενται για να σχηματιστεί το σταθμισμένο άθροισμα. Το σταθμισμένο άθροισμα των εισροών και τα αντίστοιχα βάρη τους εφαρμόζονται στη συνέχεια σε μια σχετική συνάρτηση ενεργοποίησης. Η λειτουργία ενεργοποίησης αντιστοιχίζει την είσοδο στην αντίστοιχη έξοδο.

**Βάρη και πόλωση στη βαθιά μάθηση**

Τα βάρη πρέπει να αντιστοιχηθούν σε κάθε είσοδο.
Μόλις μια μεταβλητή εισόδου τροφοδοτηθεί στο δίκτυο, μια τυχαία επιλεγμένη τιμή εκχωρείται ως το βάρος αυτής της εισόδου. Το βάρος κάθε σημείου δεδομένων εισόδου υποδεικνύει πόσο σημαντική είναι αυτή η είσοδος για την πρόβλεψη του αποτελέσματος.

Η παράμετρος προκατάληψης, από την άλλη πλευρά, επιτρέπει να προσαρμοστεί η καμπύλη της συνάρτησης ενεργοποίησης με τέτοιο τρόπο ώστε να επιτυγχάνεται ακριβής έξοδος.

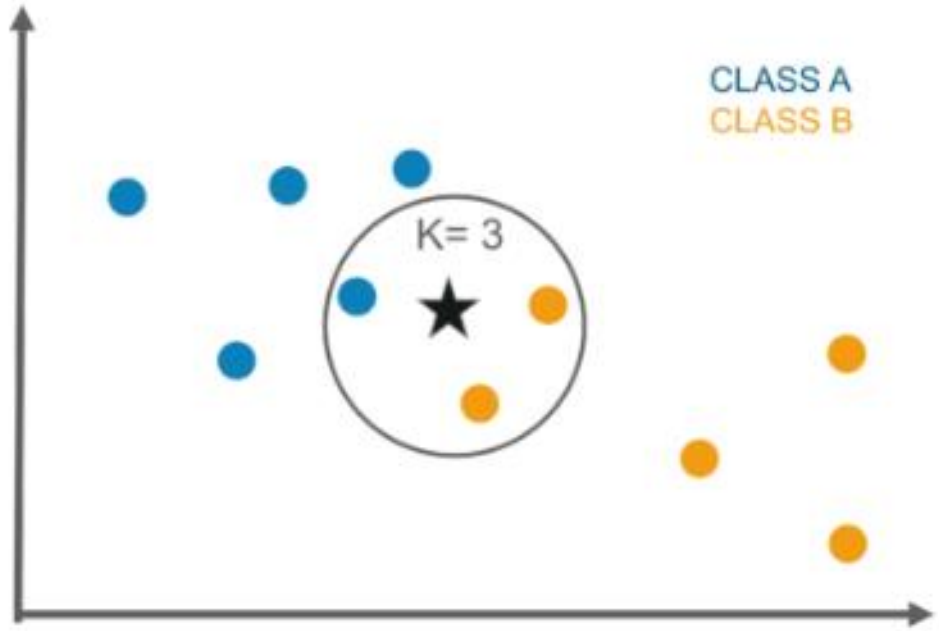
**Συνάρτηση άθροισης**

Μόλις αντιστοιχιστεί στις εισόδους κάποιο βάρος, λαμβάνεται το γινόμενο της αντίστοιχης εισόδου και βάρους. Η προσθήκη όλων αυτών των προϊόντων μας δίνει το σταθμισμένο άθροισμα. Αυτό γίνεται από τη συνάρτηση άθροισης.

**Λειτουργία ενεργοποίησης**

Ο κύριος στόχος των συναρτήσεων ενεργοποίησης είναι να αντιστοιχίσουν το σταθμισμένο άθροισμα στην έξοδο. Οι συναρτήσεις ενεργοποίησης όπως tanh, ReLU, sigmoid και ούτω καθεξής είναι παραδείγματα συναρτήσεων μετασχηματισμού.

### *2.2.4 k-Nearest Neighbor (KNN)*




**Εικόνα 9: KNN (Πηγή: https://www.datacamp.com/tutorial/k-nearest-neighbor-classification-scikit-learn)**

Το k-Nearest Neighbor είναι ένας οκνηρός αλγόριθμος εκμάθησης που αποθηκεύει όλες τις περιπτώσεις που αντιστοιχούν σε σημεία δεδομένων εκπαίδευσης σε n-διάστατο χώρο. Όταν λαμβάνεται ένα άγνωστο διακριτό στοιχείο, αναλύει τον πλησιέστερο k αριθμό αποθηκευμένων περιπτώσεων (πλησιέστεροι γείτονες) και επιστρέφει την πιο κοινή κλάση ως πρόβλεψη και για δεδομένα πραγματικών τιμών επιστρέφει τον μέσο όρο των k πλησιέστερων γειτόνων.

**Τι αντιπροσωπεύει το «k» στον αλγόριθμο kNN;**
Το k στον αλγόριθμο kNN αντιπροσωπεύει τον αριθμό των πλησιέστερων γειτονικών σημείων που ψηφίζουν για την κατηγορία των νέων δεδομένων δοκιμής.
Αν k=1, τότε στα παραδείγματα δοκιμής δίνεται η ίδια ετικέτα με το πλησιέστερο παράδειγμα στο σετ εκπαίδευσης.

Αν k=3, ελέγχονται οι ετικέτες των τριών πλησιέστερων κλάσεων και εκχωρείται η πιο κοινή (δηλαδή, εμφανίζεται τουλάχιστον δύο φορές) ετικέτα και ούτω καθεξής για μεγαλύτερα ks.

Στον αλγόριθμο του πλησιέστερου γείτονα με στάθμιση απόστασης, σταθμίζει τη συμβολή καθενός από τους k γείτονες ανάλογα με την απόστασή τους χρησιμοποιώντας το ακόλουθο ερώτημα δίνοντας μεγαλύτερη βαρύτητα στους πλησιέστερους γείτονες.

$$w \equiv \frac{1}{d(x_q, x_i)^2}$$

**Εικόνα 10: Υπολογισμός απόστασης (Πηγή: https://www.datacamp.com/tutorial/k-nearest-neighbor-classification-scikit-learn)**

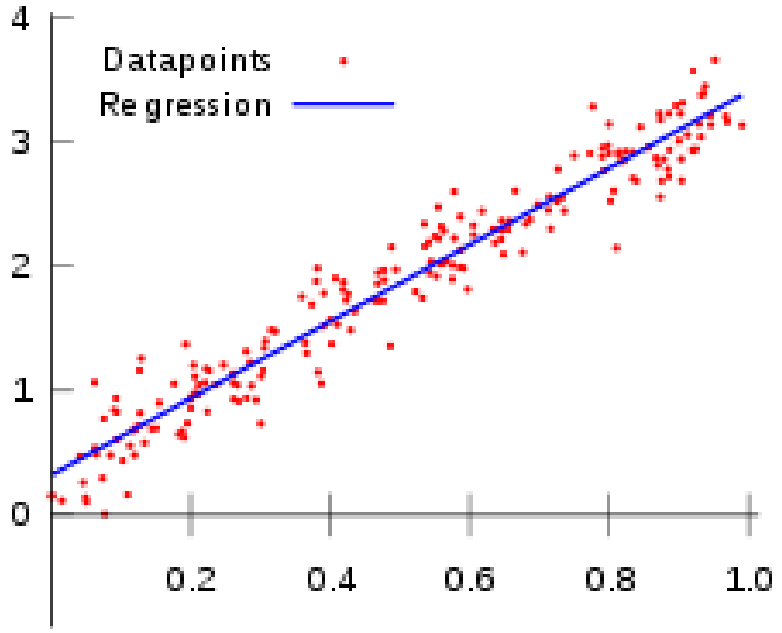
**Πλεονεκτήματα και μειονεκτήματα**

Αυτός ο αλγόριθμος είναι αρκετά απλός στην εφαρμογή του και είναι ανθεκτικός σε θορυβώδη δεδομένα εκπαίδευσης. Ακόμα κι αν τα δεδομένα εκπαίδευσης είναι μεγάλα, είναι αρκετά αποτελεσματικό. Το μόνο μειονέκτημα με τον αλγόριθμο KNN είναι ότι δεν υπάρχει ανάγκη προσδιορισμού της τιμής του K και το κόστος υπολογισμού είναι αρκετά υψηλό σε σύγκριση με άλλους αλγόριθμους.

**Περιπτώσεις χρήσης**

- Βιομηχανικές εφαρμογές για να αναζητήσετε παρόμοιες εργασίες σε σύγκριση με άλλες
- Εφαρμογές ανίχνευσης χειρόγραφου
- Αναγνώριση εικόνας
- Αναγνώριση βίντεο
- Ανάλυση μετοχών

### *2.2.5 Λογιστική παλινδρόμηση*




**Εικόνα 11: Λογιστική παλινδρόμηση (Πηγή: https://www.analyticsvidhya.com/blog/2021/07/perform-logistic-regression-with-pytorch-seamlessly/)**

Είναι ένας αλγόριθμος ταξινόμησης στη μηχανική μάθηση που χρησιμοποιεί μία ή περισσότερες ανεξάρτητες μεταβλητές για να καθορίσει ένα αποτέλεσμα. Το αποτέλεσμα μετριέται με μια διχοτομική μεταβλητή που σημαίνει ότι θα έχει μόνο δύο πιθανά αποτελέσματα.

**Τι είναι η παλινδρόμηση;**
Η ανάλυση παλινδρόμησης είναι μια ισχυρή τεχνική στατιστικής ανάλυσης. Μια εξαρτημένη μεταβλητή του ενδιαφέροντος μας χρησιμοποιείται για την πρόβλεψη των τιμών άλλων ανεξάρτητων μεταβλητών σε ένα σύνολο δεδομένων.

Χρησιμοποιεί πολλές τεχνικές για την ανάλυση και την πρόβλεψη του αποτελέσματος, αλλά η έμφαση δίνεται κυρίως στη σχέση μεταξύ εξαρτημένης μεταβλητής και μιας ή περισσότερων ανεξάρτητων μεταβλητών.

Η ανάλυση λογιστικής παλινδρόμησης προβλέπει το αποτέλεσμα σε μια δυαδική μεταβλητή που έχει μόνο δύο πιθανά αποτελέσματα.

Ο στόχος της λογιστικής παλινδρόμησης είναι να βρεθεί η καλύτερη σχέση μεταξύ της εξαρτημένης μεταβλητής και ενός συνόλου ανεξάρτητων μεταβλητών. Είναι καλύτερο από άλλους αλγόριθμους δυαδικής ταξινόμησης, όπως ο πλησιέστερος γείτονας, καθώς εξηγεί ποσοτικά τους παράγοντες που οδηγούν στην ταξινόμηση.

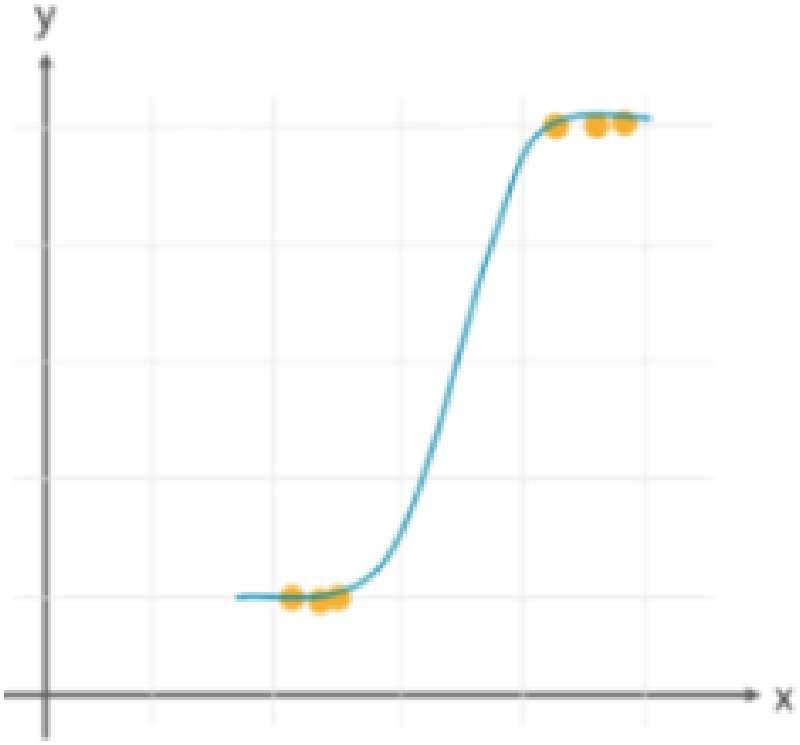
Η λογιστική παλινδρόμηση είναι μια ειδική περίπτωση γραμμικής παλινδρόμησης όπου προβλέπουμε το αποτέλεσμα μόνο σε μια κατηγορική μεταβλητή. Προβλέπει την πιθανότητα του συμβάντος χρησιμοποιώντας τη συνάρτηση καταγραφής.

Χρησιμοποιείται τη συνάρτηση/καμπύλη Sigmoid για να προβλέψει την κατηγορική τιμή. Η τιμή κατωφλίου αποφασίζει το αποτέλεσμα (νίκη/ήττα).

Εξίσωση γραμμικής παλινδρόμησης**: y = β0 + β1X1 + β2X2 …. + βnXn**

- Το Y σημαίνει την εξαρτημένη μεταβλητή που πρέπει να προβλεφθεί.
- Το β0 είναι η τομή Y, η οποία είναι βασικά το σημείο της ευθείας που αγγίζει τον άξονα y.
- Το β1 είναι η κλίση της γραμμής (η κλίση μπορεί να είναι αρνητική ή θετική ανάλογα με τη σχέση μεταξύ της εξαρτημένης μεταβλητής και της ανεξάρτητης μεταβλητής.)
- Το X εδώ αντιπροσωπεύει την ανεξάρτητη μεταβλητή που χρησιμοποιείται για την πρόβλεψη της προκύπτουσας εξαρτημένης τιμής μας.

Σιγμοειδής συνάρτηση: $\mathbf{p = 1 / 1 + e^{-y}}$




**Εικόνα 12: Σιγμοειδής συνάρτηση (Πηγή: https://www.analyticsvidhya.com/blog/2021/07/perform-logistic-regression-with-pytorch-seamlessly/)**

Εξίσωση Logistic Regression: $\mathbf{p = 1 / 1 + e^{-(\beta 0 + \beta 1X1 + \beta 2X2 \ldots. + \beta nXn)}}$

**Πλεονεκτήματα και μειονεκτήματα**

Η λογιστική παλινδρόμηση προορίζεται ειδικά για ταξινόμηση, είναι χρήσιμη για την κατανόηση του πώς ένα σύνολο ανεξάρτητων μεταβλητών επηρεάζει το αποτέλεσμα της εξαρτημένης μεταβλητής.

Το κύριο μειονέκτημα του αλγόριθμου λογιστικής παλινδρόμησης είναι ότι λειτουργεί μόνο όταν η προβλεπόμενη μεταβλητή είναι δυαδική, υποθέτει ότι τα δεδομένα δεν λείπουν από τιμές και υποθέτει ότι οι προγνωστικοί παράγοντες είναι ανεξάρτητοι μεταξύ τους.

**Περιπτώσεις χρήσης**

- Προσδιορισμός παραγόντων κινδύνου για ασθένειες
- Ταξινόμηση λέξεων
- Πρόβλεψη καιρού
- Αιτήσεις Ψηφοφορίας

### *2.2.6 Support Vector Machine (Μηχανή Διανυσμάτων Υποστήριξης)*

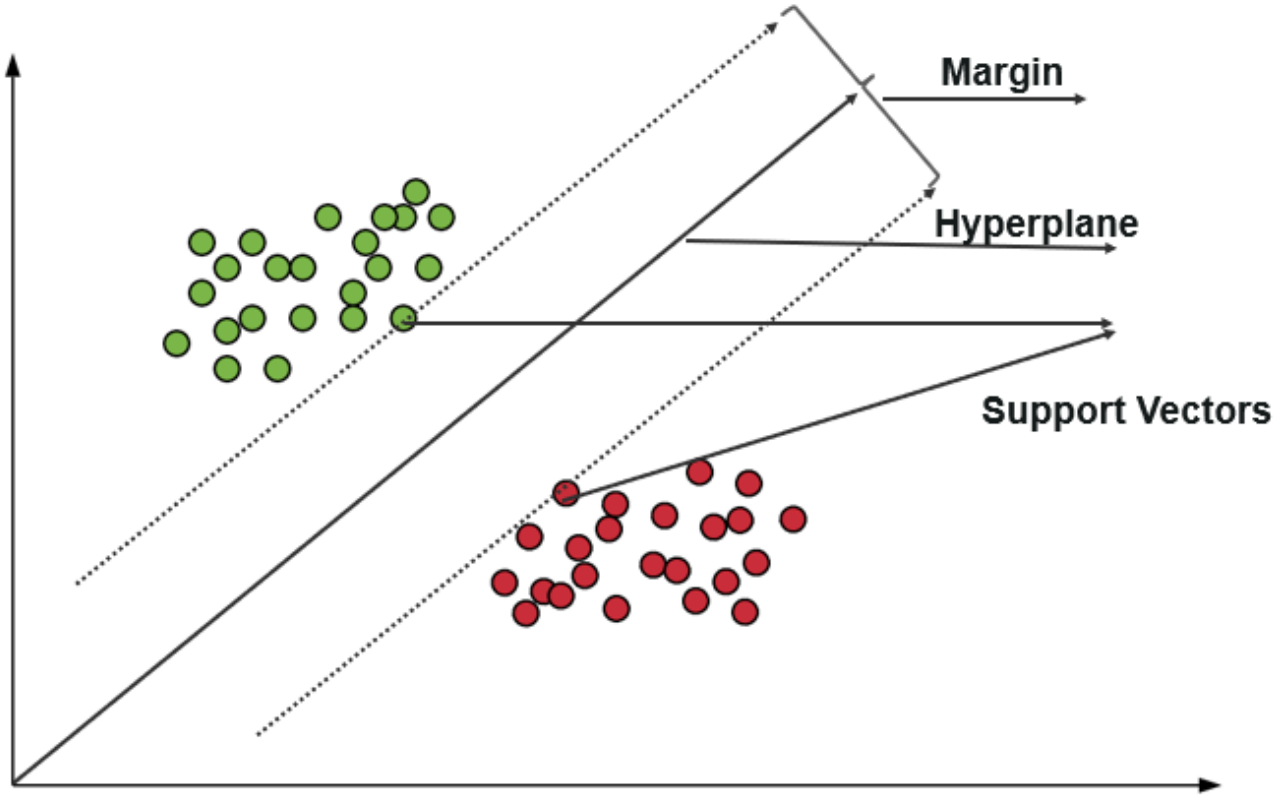
Η μηχανή διανυσμάτων υποστήριξης είναι ένας ταξινομητής που αναπαριστά τα δεδομένα εκπαίδευσης ως σημεία στο χώρο χωρισμένα σε κατηγορίες με ένα όσο το δυνατόν ευρύτερο κενό. Στη συνέχεια προστίθενται νέα σημεία στο διάστημα προβλέποντας σε ποια κατηγορία ανήκουν και σε ποιο χώρο θα ανήκουν.

**Πώς λειτουργεί το SVM;**
Ο κύριος στόχος μιας μηχανής διανυσμάτων υποστήριξης είναι να διαχωρίσει τα δεδομένα με τον καλύτερο δυνατό τρόπο. Όταν γίνει ο διαχωρισμός, η απόσταση μεταξύ των πλησιέστερων σημείων είναι γνωστή ως περιθώριο. Η προσέγγιση είναι να επιλέξετε ένα υπερεπίπεδο (hyperplane) με το μέγιστο δυνατό περιθώριο μεταξύ των διανυσμάτων υποστήριξης στα δεδομένα σύνολα δεδομένων.

Για να επιλέγει το μέγιστο υπερεπίπεδο στα δεδομένα σύνολα, η μηχανή διανυσμάτων υποστήριξης ακολουθεί τα ακόλουθα σύνολα:

- Δημιουργούνται υπερεπίπεδα που διαχωρίζουν τις κατηγορίες με τον καλύτερο δυνατό τρόπο

- Επιλέξτε το σωστό υπερεπίπεδο με τον μέγιστο διαχωρισμό από τα πλησιέστερα σημεία δεδομένων




**Εικόνα 13: SVM Διάγραμμα (Πηγή: https://www.researchgate.net/figure/Classification-of-data-by-support-vector-machine-SVM_fig8_304611323)**

**SVM Kernels**

Ένας πυρήνας SVM προσθέτει βασικά περισσότερες διαστάσεις σε έναν χώρο χαμηλών διαστάσεων για να διευκολύνει τον διαχωρισμό των δεδομένων. Μετατρέπει το αδιαχώριστο πρόβλημα σε προβλήματα διαχωρισμού προσθέτοντας περισσότερες διαστάσεις χρησιμοποιώντας το κόλπο του πυρήνα. Ένα μηχάνημα φορέα υποστήριξης υλοποιείται στην πράξη από έναν πυρήνα. Το κόλπο του πυρήνα βοηθά στη δημιουργία ενός πιο ακριβούς ταξινομητή. Ας ρίξουμε μια ματιά στους διαφορετικούς πυρήνες στη μηχανή διανυσμάτων υποστήριξης.

- Γραμμικός Kernel – Ένας γραμμικός πυρήνας μπορεί να χρησιμοποιηθεί ως κανονικό γινόμενο κουκκίδων μεταξύ οποιωνδήποτε δύο δεδομένων παρατηρήσεων. Το γινόμενο μεταξύ των δύο διανυσμάτων είναι το άθροισμα του πολλαπλασιασμού κάθε ζεύγους τιμών εισόδου. Ακολουθεί η γραμμική εξίσωση του πυρήνα.

$$f(x) = B(0) + sum(ai * (x,xi))$$

- Πολυωνυμικός Kernel – Είναι μια μάλλον γενικευμένη μορφή του γραμμικού πυρήνα. Μπορεί να διακρίνει καμπύλο ή μη γραμμικό χώρο εισόδου. Ακολουθεί η πολυωνυμική εξίσωση πυρήνα.

$$K(X_1, X_2) = (a + X_1^T X_2)^b$$

b = degree of kernel & a = constant term.

- Kernel συνάρτησης ακτινικής βάσης – Ο πυρήνας της συνάρτησης ακτινικής βάσης χρησιμοποιείται συνήθως στην ταξινόμηση SVM, μπορεί να χαρτογραφήσει το χώρο σε άπειρες διαστάσεις. Ακολουθεί η εξίσωση του πυρήνα RBF.

**Πλεονεκτήματα και μειονεκτήματα**

Χρησιμοποιεί ένα υποσύνολο σημείων εκπαίδευσης στη συνάρτηση απόφασης που το καθιστά αποδοτικό στη μνήμη και είναι εξαιρετικά αποτελεσματικό σε χώρους υψηλών διαστάσεων. Το μόνο μειονέκτημα της μηχανής διανυσμάτων υποστήριξης είναι ότι ο αλγόριθμος δεν παρέχει άμεσα εκτιμήσεις πιθανοτήτων.

**Πλεονεκτήματα του SVM**

- Αποτελεσματικό σε χώρους υψηλών διαστάσεων
- Εξακολουθεί να είναι αποτελεσματικό σε περιπτώσεις όπου ο αριθμός των διαστάσεων είναι μεγαλύτερος από τον αριθμό των δειγμάτων
- Χρησιμοποιεί ένα υποσύνολο σημείων εκπαίδευσης στη συνάρτηση απόφασης που το καθιστά αποδοτικό στη μνήμη
- Μπορούν να καθοριστούν διαφορετικές συναρτήσεις πυρήνα για τη συνάρτηση απόφασης που την καθιστά επίσης ευέλικτη

**Μειονεκτήματα του SVM**

- Εάν ο αριθμός των χαρακτηριστικών είναι πολύ μεγαλύτερος από τον αριθμό των δειγμάτων, αποφύγετε την υπερβολική προσαρμογή στην επιλογή των συναρτήσεων του πυρήνα και ο όρος τακτοποίησης είναι ζωτικής σημασίας.
- Τα SVM δεν παρέχουν άμεσα εκτιμήσεις πιθανοτήτων, αυτές υπολογίζονται χρησιμοποιώντας πενταπλάσια διασταυρούμενη επικύρωση.

**Περιπτώσεις χρήσης**

- Επιχειρηματικές εφαρμογές για σύγκριση της απόδοσης μιας μετοχής σε μια χρονική περίοδο
- Επενδυτικές προτάσεις
- Ταξινόμηση εφαρμογών που απαιτούν ακρίβεια και αποτελεσματικότητα
- Ανίχνευση προσώπου
- Κατηγοριοποίηση κειμένου και υπερκειμένου
- Ταξινόμηση εικόνων
- Βιοπληροφορική
- Αναδίπλωση πρωτεΐνης και απομακρυσμένη ανίχνευση ομολογίας
- Αναγνώριση χειρογράφου
- Γενικευμένος Προγνωστικός Έλεγχος

## *2.3 Γιατί είναι σημαντικοί οι ταξινομητές;*

Η ταξινόμηση – δηλαδή η αντιστοίχιση μιας εισαγωγής δεδομένων με μια συγκεκριμένη ετικέτα κλάσης – είναι μια θεμελιώδης λειτουργία πολλών εφαρμογών τεχνητής νοημοσύνης και οι ταξινομητές αποτελούν βασικό στοιχείο σε πολλές από αυτές τις εφαρμογές. Οι ταξινομητές χρησιμοποιούνται ευρέως για μια σειρά από περιπτώσεις κοινής χρήσης, όπως για τον προσδιορισμό εάν ένας πελάτης ανήκει σε ένα συγκεκριμένο τμήμα, για τον προσδιορισμό εάν μια χρηματοοικονομική συναλλαγή είναι δόλια ή για τον προσδιορισμό εάν ένα κομμάτι εξοπλισμού πεδίου είναι σε λειτουργική κατάσταση με βάση μια φωτογραφία ή ένα βίντεο πλάνο.

Η ταξινόμηση είναι ένας ισχυρός τομέας συνεχούς έρευνας και καινοτομίας μηχανικής μάθησης. Σημαντική ακαδημαϊκή και εμπορική προσπάθεια έχει επενδυθεί στην ανάπτυξη μιας ποικίλης επιλογής αλγορίθμων ταξινομητών βελτιστοποιημένων για διαφορετικούς τύπους προβλημάτων ταξινόμησης. Έχουν αναπτυχθεί πολυάριθμες ισχυρές μέθοδοι ταξινομητή και πολλές είναι διαθέσιμες μέσω βιβλιοθηκών ανοιχτού κώδικα, για παράδειγμα ταξινομητές Python από το Pypi.org.

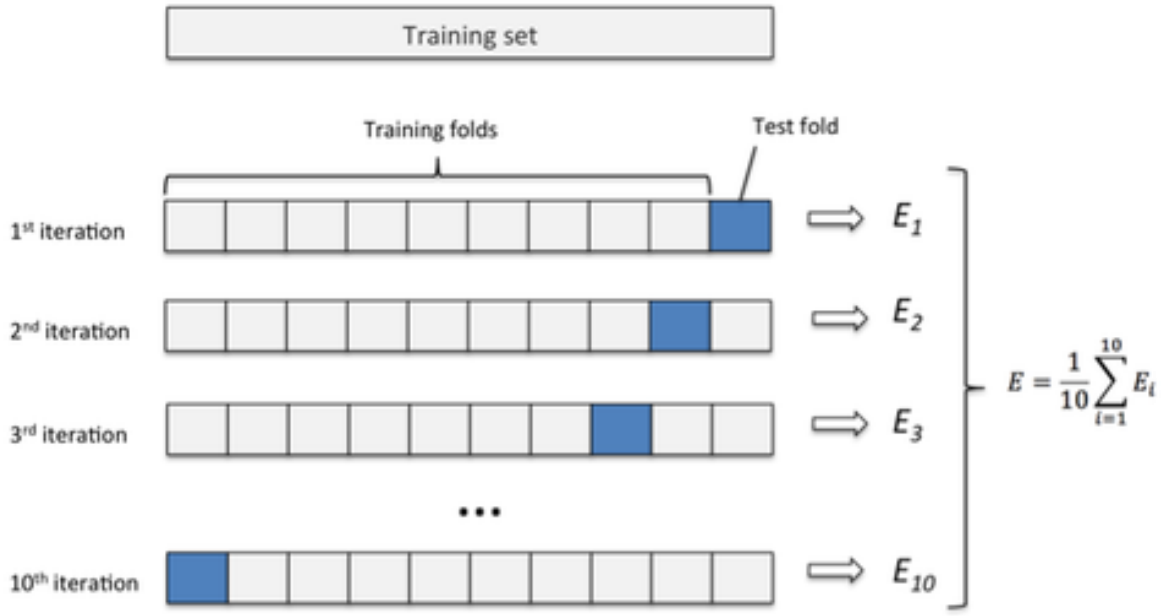
## *2.4 Αξιολόγηση ενός ταξινομητή*

Το πιο σημαντικό κομμάτι μετά την ολοκλήρωση οποιουδήποτε ταξινομητή είναι η αξιολόγηση για τον έλεγχο της ακρίβειας και της αποτελεσματικότητάς του. Υπάρχουν πολλοί τρόποι με τους οποίους μπορούμε να αξιολογήσουμε έναν ταξινομητή. Ας ρίξουμε μια ματιά σε αυτές τις μεθόδους που αναφέρονται παρακάτω.
Μετά την εκπαίδευση του μοντέλου, το πιο σημαντικό μέρος είναι η αξιολόγηση του ταξινομητή για να επαληθευτεί η εφαρμογή του.

### *2.4.1 Μέθοδος κράτησης*

Υπάρχουν πολλές μέθοδοι και η πιο κοινή μέθοδος είναι η μέθοδος κράτησης. Σε αυτή τη μέθοδο, το δεδομένο σύνολο δεδομένων χωρίζεται σε 2 διαμερίσματα ως δοκιμή και εκπαίδευση 20% και 80% αντίστοιχα. Το σετ εκπαίδευσης που θα χρησιμοποιηθεί για την εκπαίδευση του μοντέλου και τα αόρατα δεδομένα δοκιμής θα χρησιμοποιηθούν για τη δοκιμή της προγνωστικής του ισχύος.

### *2.4.2 Cross-validation (Διασταυρωμένη επικύρωση)*




**Εικόνα 14: Διασταυρωμένη επικύρωση (Πηγή: https://towardsdatascience.com/cross-validation-c4fae714f1c5)**

Η υπερβολική εφαρμογή (over-fitting) είναι ένα κοινό πρόβλημα στη μηχανική εκμάθηση που μπορεί να παρουσιαστεί στα περισσότερα μοντέλα. Μπορεί να διεξαχθεί διασταυρούμενη επικύρωση k-fold για να επαληθευτεί ότι το μοντέλο δεν έχει τοποθετηθεί υπερβολικά. Σε αυτή τη μέθοδο, το σύνολο δεδομένων διαιρείται τυχαία σε k αμοιβαία αποκλειόμενα υποσύνολα, το καθένα περίπου ίσο μέγεθος και ένα διατηρείται για δοκιμή ενώ άλλα χρησιμοποιούνται για εκπαίδευση. Αυτή η διαδικασία επαναλαμβάνεται σε όλες τις πτυχές k.

### *2.4.3* Classification Report

Μια αναφορά ταξινόμησης θα δώσει τα ακόλουθα αποτελέσματα, είναι ένα δείγμα αναφοράς ταξινόμησης ενός ταξινομητή SVM που χρησιμοποιεί ένα σύνολο δεδομένων.

- **Ακρίβεια**
  - Η ακρίβεια είναι ο λόγος της σωστά προβλεπόμενης παρατήρησης προς το σύνολο των παρατηρήσεων
  - True Positive: Ο αριθμός των σωστών προβλέψεων ότι η εμφάνιση είναι θετική.
  - True Negative: Αριθμός σωστών προβλέψεων ότι η εμφάνιση είναι αρνητική.

- **F1- Score**

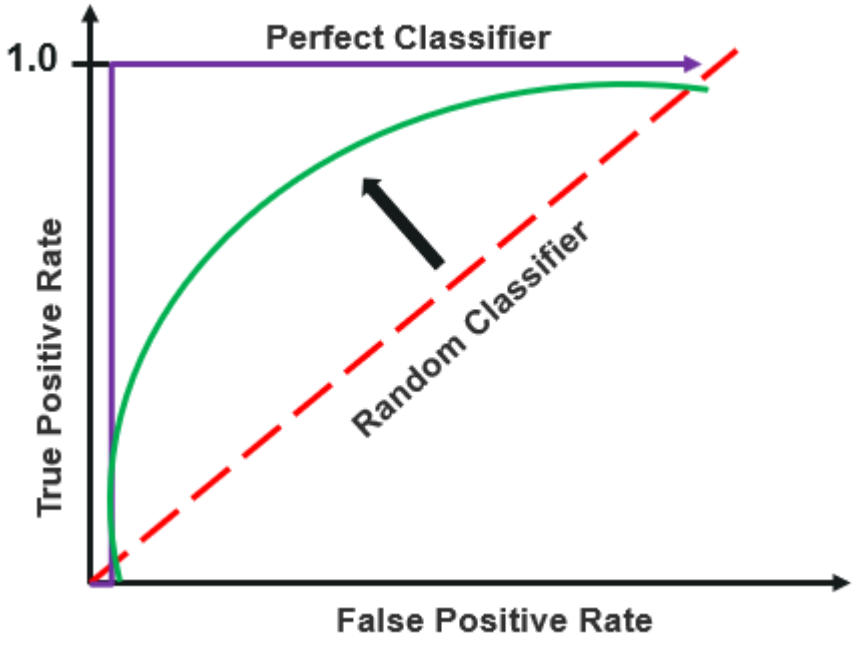
  - Είναι ο σταθμισμένος μέσος όρος ακρίβειας και ανάκλησης

- **Ακρίβεια και ανάκληση**

  - Η ακρίβεια είναι το κλάσμα των σχετικών παρουσιών μεταξύ των ανακτημένων περιπτώσεων, ενώ η ανάκληση είναι το κλάσμα των σχετικών περιπτώσεων που έχουν ανακτηθεί σε σχέση με τον συνολικό αριθμό των περιπτώσεων. Χρησιμοποιούνται βασικά ως μέτρο συνάφειας.

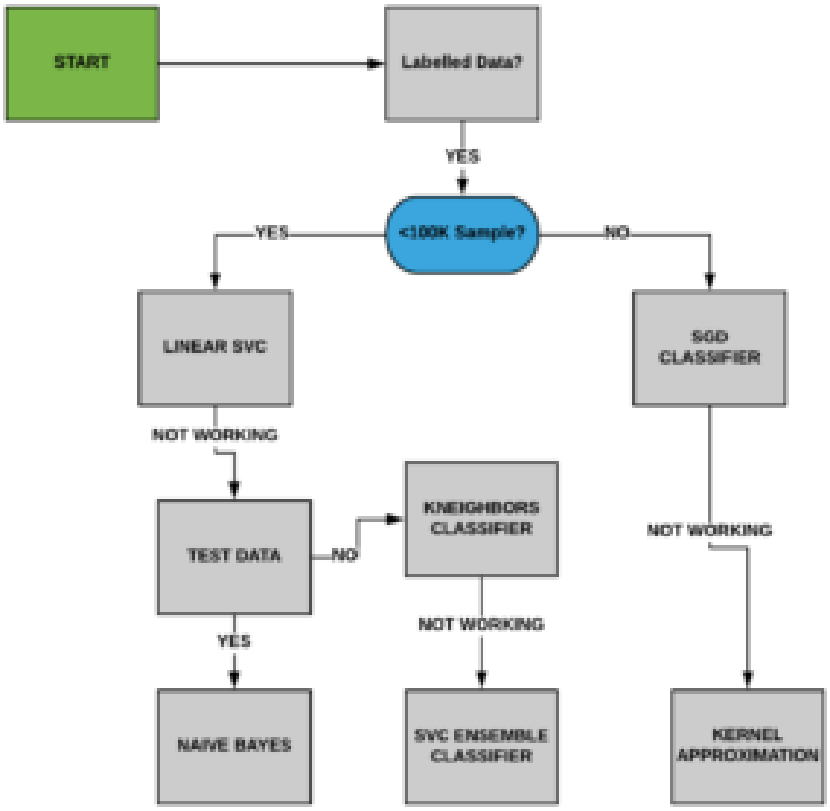
### *2.4.4 Καμπύλη ROC (Receiver Operating Characteristics) (Χαρακτηριστικά λειτουργίας δέκτη)*

Τα χαρακτηριστικά λειτουργίας του δέκτη ή η καμπύλη ROC χρησιμοποιείται για οπτική σύγκριση μοντέλων ταξινόμησης, η οποία δείχνει τη σχέση μεταξύ του πραγματικού θετικού ρυθμού και του ψευδώς θετικού ρυθμού. Η περιοχή κάτω από την καμπύλη ROC είναι το μέτρο της ακρίβειας του μοντέλου.




**Εικόνα 15: Καμπύλη ROC (Πηγή: https://towardsdatascience.com/cross-validation-c4fae714f1c5)**

## Επιλογή αλγορίθμου




**Εικόνα 16: Κατάλληλη επιλογή αλγόριθμου (Πηγή: 2 https://towardsdatascience.com/cross-validation-c4fae714f1c5)**

Εκτός από την παραπάνω προσέγγιση, μπορούν να ακολουθηθούν τα παρακάτω βήματα για να χρησιμοποιηθεί ο καλύτερος αλγόριθμος για το μοντέλο

- Ανάγνωση δεδομένων
- Δημιουργία εξαρτημένων και ανεξάρτητων συνόλων δεδομένων με βάση τα εξαρτημένα και ανεξάρτητα χαρακτηριστικά μας
- Διαχωρισμός των δεδομένων σε σετ εκπαίδευσης και δοκιμών
- Εκπαίδευση του μοντέλου χρησιμοποιώντας διαφορετικούς αλγόριθμους όπως KNN, Δέντρο αποφάσεων, SVM κ.λπ
- Αξιολόγηση του ταξινομητή
- Επιλογή του ταξινομητή με τη μεγαλύτερη ακρίβεια.

Αν και μπορεί να χρειαστεί περισσότερος χρόνος από όσο χρειάζεται για να επιλεγεί ο καλύτερος αλγόριθμος που ταιριάζει στο μοντέλο, η ακρίβεια είναι ο καλύτερος τρόπος να προχωρήσει για να γίνει το μοντέλο αποτελεσματικό.

# ΚΕΦΑΛΑΙΟ 3: ΕΦΑΡΜΟΓΗ ΣΤΗΝ ΥΓΕΙΑ

### *3.1 Ευκαιρίες και Πλεονεκτήματα για την Υγειονομική Περίθαλψη*

Η χρήση μηχανικής μάθησης για την υγειονομική περίθαλψη για τις παραπάνω εργασίες μπορεί να προσφέρει πολλές ευκαιρίες στους οργανισμούς υγειονομικής περίθαλψης. Πρώτον, επιτρέπει στους επαγγελματίες υγείας να εστιάζουν στη φροντίδα των ασθενών αντί να ξοδεύουν το χρόνο τους στην αναζήτηση ή την καταχώριση πληροφοριών.

Ο δεύτερος σημαντικός ρόλος της μηχανικής μάθησης στην υγειονομική περίθαλψη είναι η αύξηση της ακρίβειας διάγνωσης. Για παράδειγμα, η μηχανική μάθηση έχει αποδειχθεί ότι είναι 92% ακριβής στην πρόβλεψη της θνησιμότητας ασθενών με COVID-19.

Τρίτον, η χρήση μηχανικής μάθησης στην ιατρική μπορεί να βοηθήσει στην ανάπτυξη ενός πιο ακριβούς σχεδίου θεραπείας. Πολλές ιατρικές περιπτώσεις είναι μοναδικές και απαιτούν ειδική προσέγγιση για αποτελεσματική φροντίδα και μείωση των παρενεργειών. Οι αλγόριθμοι μηχανικής μάθησης μπορούν να απλοποιήσουν την αναζήτηση τέτοιων λύσεων.

Η χρήση μηχανικής μάθησης σε λειτουργίες υγειονομικής περίθαλψης μπορεί να είναι εξαιρετικά επωφελής για την εταιρεία. Η μηχανική εκμάθηση δημιουργήθηκε για να αντιμετωπίζει μεγάλα σύνολα δεδομένων και τα αρχεία ασθενών είναι ακριβώς αυτό – πολλά σημεία δεδομένων που χρειάζονται ενδελεχή ανάλυση και οργάνωση.

Επιπλέον, ενώ ένας επαγγελματίας υγείας και ένας αλγόριθμος μηχανικής μάθησης πιθανότατα θα καταλήξουν στο ίδιο συμπέρασμα με βάση το ίδιο σύνολο δεδομένων, η χρήση μηχανικής μάθησης θα έχει τα αποτελέσματα πολύ πιο γρήγορα, επιτρέποντας την έναρξη της θεραπείας νωρίτερα.

Ένα άλλο σημείο για τη χρήση τεχνικών μηχανικής μάθησης στην υγειονομική περίθαλψη είναι η εξάλειψη της ανθρώπινης συμμετοχής σε κάποιο βαθμό, γεγονός που μειώνει την πιθανότητα ανθρώπινου λάθους. Αυτό αφορά ιδιαίτερα τις εργασίες αυτοματισμού διαδικασιών, καθώς η κουραστική εργασία ρουτίνας είναι εκεί που οι άνθρωποι κάνουν τα περισσότερα σφάλματα.

## *3.2 Εφαρμογές της Μηχανικής Μάθησης στην Υγεία*

### *3.2.1 Συστήματα Υποστήριξης Κλινικών Αποφάσεων*

Τα εργαλεία υποστήριξης κλινικών αποφάσεων βοηθούν στην ανάλυση μεγάλου όγκου δεδομένων για τον εντοπισμό μιας ασθένειας, την απόφαση για το επόμενο στάδιο θεραπείας, τον προσδιορισμό τυχόν πιθανών προβλημάτων και τη συνολική βελτίωση της αποτελεσματικότητας της φροντίδας των ασθενών. Το CDSS είναι ένα ισχυρό εργαλείο που βοηθά τον ιατρό να κάνει τη δουλειά του αποτελεσματικά και γρήγορα και μειώνει τις πιθανότητες να πάρει λάθος διάγνωση ή να συνταγογραφήσει αναποτελεσματική θεραπεία.

Αυτή η χρήση της μηχανικής μάθησης στην ιατρική (υγειονομική περίθαλψη) υπάρχει εδώ και λίγο καιρό, αλλά έχει γίνει πιο διαδεδομένη τα τελευταία χρόνια. Ο λόγος πίσω από αυτό είναι η ευρύτερη αποδοχή του συστήματος ηλεκτρονικού μητρώου υγείας (EHR) και η ψηφιοποίηση διαφόρων σημείων δεδομένων, συμπεριλαμβανομένων των ιατρικών εικόνων.

### *3.2.2 Έξυπνη τήρηση αρχείων*

Η διασφάλιση ότι όλα τα αρχεία ασθενών ενημερώνονται τακτικά είναι πρόκληση, καθώς η εισαγωγή δεδομένων είναι μια μονότονη εργασία. Ωστόσο, είναι επίσης ζωτικής σημασίας για την αποτελεσματική λήψη αποφάσεων και την καλύτερη φροντίδα των ασθενών.

Μία από τις χρήσεις της μηχανικής μάθησης στην υγειονομική περίθαλψη είναι η χρήση της τεχνολογίας οπτικής αναγνώρισης χαρακτήρων (OCR) στο χειρόγραφο των ιατρών, καθιστώντας την εισαγωγή δεδομένων γρήγορη και απρόσκοπτη. Αυτά τα δεδομένα μπορούν στη συνέχεια να αναλυθούν από άλλα εργαλεία μηχανικής μάθησης για τη βελτίωση της λήψης αποφάσεων και της φροντίδας των ασθενών.

### *3.2.3 Μηχανική Μάθηση στην Ιατρική Απεικόνιση*

Για το μεγαλύτερο χρονικό διάστημα, οι ιατρικές εικόνες, όπως οι ακτίνες Χ, ήταν αναλογικές. Αυτό έχει περιορίσει τη χρήση της τεχνολογίας για την αναγνώριση ανωμαλιών, την ομαδοποίηση περιπτώσεων και τη συνολική έρευνα ασθενειών. Ευτυχώς, η ψηφιοποίηση της διαδικασίας οδήγησε σε περισσότερες ευκαιρίες με αυτούς τους τύπους ανάλυσης δεδομένων, μεταξύ άλλων με τη βοήθεια της μηχανικής μάθησης. Και, σύμφωνα με μια πρόσφατη μετα-ανάλυση, οι αλγόριθμοι μηχανικής μάθησης κάνουν τη δουλειά τους όπως (και, σε ορισμένες περιπτώσεις, ακόμη καλύτεροι) οι ειδικοί στον άνθρωπο, με 87,0% ευαισθησία και 92,5% ειδικότητα για τους αλγόριθμους βαθιάς μάθησης και 86,4% ευαισθησία και 90,5% ειδικότητα για τους γιατρούς του ανθρώπου.

Ένα από τα γνωστά επιτυχημένα παραδείγματα μηχανικής μάθησης στην υγειονομική περίθαλψη είναι το έργο InnerEye από τη Microsoft. Η αρχική εστίασή του ήταν σε τρισδιάστατες ακτινολογικές εικόνες, όπου κατασκευάστηκαν εργαλεία ML για τη διαφοροποίηση υγιών κυττάρων και όγκων.

### *3.2.4 Εξατομικευμένη Ιατρική*

Αυτό που κάνει την ιατρική έναν τόσο περίπλοκο και πλούσιο σε πόρους τομέα είναι ότι κάθε περίπτωση έχει τις ιδιαιτερότητές της. Οι άνθρωποι έχουν συχνά μια σειρά παθήσεων που απαιτούν ταυτόχρονη θεραπεία. Επομένως, πρέπει να ληφθούν περίπλοκες αποφάσεις για την κατασκευή ενός αποτελεσματικού σχεδίου θεραπείας, λαμβάνοντας υπόψη τις αλληλεπιδράσεις με τα φάρμακα και ελαχιστοποιώντας τις πιθανές παρενέργειες.

### *3.2.5 Προσαρμογές συμπεριφοράς*

Η πρόληψη είναι εξίσου σημαντική στην υγειονομική περίθαλψη με τη θεραπεία ασθενειών. Ένα από τα πιο σημαντικά μέρη της προληπτικής ιατρικής είναι η τροποποίηση της συμπεριφοράς κάποιου για να απαλλαγεί από ανθυγιεινές συνήθειες και να καθιερώσει έναν υγιεινό τρόπο ζωής.

Ένα από τα οφέλη της μηχανικής μάθησης στην υγειονομική περίθαλψη είναι ότι μπορεί να χρησιμοποιηθεί για να επισημάνουμε κάτι που δεν παρατηρούμε. Αυτό ακριβώς κάνει η Somatix. Αυτή η εφαρμογή που βασίζεται στη μηχανική μάθηση ακολουθεί την καθημερινή δραστηριότητα του ασθενούς και επισημαίνει τις ασυνείδητες συνήθειες και τη ρουτίνα του, ώστε να μπορούν να επικεντρωθούν στην απαλλαγή από αυτές.

### *3.2.6 Προγνωστική Προσέγγιση Θεραπείας*

Όταν πρόκειται για τις περισσότερες επικίνδυνες ασθένειες, ο εντοπισμός τους στα αρχικά στάδια μπορεί να αυξήσει σημαντικά τις πιθανότητες επιτυχούς θεραπείας. Αυτό βοηθά επίσης στον εντοπισμό της πιθανότητας οποιασδήποτε πιθανής επιδείνωσης της κατάστασης του ασθενούς πριν συμβεί.

Μία από τις περιπτώσεις για τη σημασία της μηχανικής μάθησης στην υγειονομική περίθαλψη είναι ότι μπορεί να χρησιμοποιηθεί για την επιτυχή πρόβλεψη ορισμένων από τις πιο επικίνδυνες ασθένειες σε ασθενείς σε κίνδυνο. Αυτό περιλαμβάνει την αναγνώριση σημείων διαβήτη (με χρήση αλγόριθμου Naïve Bayes), ηπατικών και νεφρικών παθήσεων και όγκων.

### *3.2.7 Συλλογή δεδομένων*

Μία από τις πιο σημαντικές ευθύνες για έναν γιατρό είναι να συγκεντρώνει σωστά το ιστορικό ενός ασθενούς. Αυτό μπορεί συχνά να είναι προκλητικό, καθώς ο ασθενής δεν είναι ειδικός και δεν γνωρίζει ποια δεδομένα είναι σχετικά προς αποκάλυψη.

Χρησιμοποιώντας τη μηχανική μάθηση στη διαχείριση της υγειονομικής περίθαλψης, οι επαγγελματίες υγείας μπορούν να καθορίσουν τις πιο σχετικές ερωτήσεις που πρέπει να κάνουν σε έναν ασθενή με βάση διάφορους δείκτες. Αυτό θα βοηθήσει στη συλλογή σχετικών δεδομένων και, ταυτόχρονα, θα λάβει μια πρόβλεψη των πιο πιθανών συνθηκών.

### *3.2.8 Φροντίδα ηλικιωμένων και ομάδων χαμηλής κινητικότητας*

Η μηχανική μάθηση και η ιατρική μπορούν να βοηθήσουν τις ομάδες χαμηλής κινητικότητας (συμπεριλαμβανομένων των ηλικιωμένων και των ατόμων που χρησιμοποιούν αναπηρικά αμαξίδια) να βελτιώσουν την καθημερινότητά τους με έξυπνες υπενθυμίσεις και βοήθεια προγραμματισμού, να προβλέψουν και να αποφύγουν πιθανούς τραυματισμούς εντοπίζοντας κοινά εμπόδια και προσδιορίζοντας τις βέλτιστες διαδρομές.

Αν και αυτές οι λύσεις είναι αποτελεσματικές, δεν είναι τόσο διαδεδομένες όσο χρειάζεται. Ωστόσο, οι εταιρείες υγειονομικής περίθαλψης λαμβάνουν ήδη μέτρα για να τα καταστήσουν ευρέως διαθέσιμα. Για παράδειγμα, στην Ιαπωνία, υπάρχει σχέδιο να πραγματοποιείται το 75% της φροντίδας ηλικιωμένων από AI.

### *3.2.9 Ρομποτική Χειρουργική*

Οι χειρουργικές επεμβάσεις απαιτούν μεγάλη ακρίβεια, προσαρμοστικότητα στις μεταβαλλόμενες συνθήκες και σταθερή προσέγγιση για μεγάλο χρονικό διάστημα. Ενώ οι εκπαιδευμένοι χειρουργοί έχουν όλες αυτές τις ιδιότητες, μία από τις ευκαιρίες στη μηχανική μάθηση για την υγειονομική περίθαλψη είναι τα ρομπότ να εκπληρώσουν αυτές τις εργασίες. Αυτή τη στιγμή, η ρομποτική χειρουργική μπορεί να χρησιμοποιηθεί αποτελεσματικά ως βοήθεια για τους χειρουργούς. Συγκεκριμένα, η μηχανική μάθηση μπορεί να χρησιμοποιηθεί για καλύτερη μοντελοποίηση και σχεδιασμό χειρουργικής επέμβασης, αξιολόγηση των δεξιοτήτων του χειρουργού και απλοποίηση χειρουργικών εργασιών όπως η συρραφή.

### *3.2.10 Ανακάλυψη και παραγωγή φαρμάκων*

Με βάση τα δεδομένα που αποκτήθηκαν προηγουμένως σχετικά με τα ενεργά συστατικά των φαρμάκων και τον τρόπο με τον οποίο επηρεάζουν τον οργανισμό, οι αλγόριθμοι ML μπορούν να μοντελοποιήσουν ένα ενεργό συστατικό που θα λειτουργούσε σε μια άλλη παρόμοια ασθένεια.

Μια τέτοια προσέγγιση μπορεί να χρησιμοποιηθεί για την ανάπτυξη προσωπικής φαρμακευτικής αγωγής για ασθενείς με ένα μοναδικό σύνολο ασθενειών ή ορισμένες ειδικές απαιτήσεις. Στο μέλλον, αυτό το εργαλείο μηχανικής εκμάθησης θα μπορούσε να χρησιμοποιηθεί σε συνδυασμό με τη νανοτεχνολογία για καλύτερη παροχή φαρμάκων.

### *3.2.11 Κλινική Έρευνα*

Η κλινική έρευνα και οι δοκιμές είναι δαπανηρές και χρονοβόρες διαδικασίες. Υπάρχει ένας καλός λόγος πίσω από αυτό – τα νέα φάρμακα και οι ιατρικές διαδικασίες θα πρέπει να αποδεικνύονται ασφαλή πριν χρησιμοποιηθούν ευρέως. Ωστόσο, υπάρχουν περιπτώσεις που το σκεύασμα πρέπει να κυκλοφορήσει το συντομότερο δυνατό – όπως με τα εμβόλια για τον COVID-19.

Ευτυχώς, υπάρχει τρόπος να συντομευθεί η διαδικασία με τη βοήθεια αλγορίθμων μηχανικής μάθησης. Μπορεί να χρησιμοποιηθεί για τον προσδιορισμό του καλύτερου δείγματος για τη δοκιμή, τη συλλογή περισσότερων σημείων δεδομένων, την ανάλυση των συνεχιζόμενων δεδομένων από τους συμμετέχοντες στη δοκιμή και τη μείωση των σφαλμάτων που βασίζονται σε δεδομένα.

### *3.2.12 Πρόβλεψη επιδημίας λοιμωδών νοσημάτων*

Η πανδημία του COVID-19 μας έδειξε πόσο απροετοίμαστοι ήμασταν σε μια επιδημία μολυσματικής νόσου αυτού του μεγέθους. Αξίζει να αναφερθεί ότι ειδικοί του χώρου έχουν προειδοποιήσει εδώ και χρόνια την κυβέρνηση για το ενδεχόμενο ενός τέτοιου γεγονότος. Τώρα, έχουμε εργαλεία βασισμένα στη μηχανική μάθηση που μπορούν να βοηθήσουν στον έγκαιρο εντοπισμό των σημείων μιας επιδημίας. Οι αλγόριθμοι αναλύουν τα δορυφορικά δεδομένα, τις ειδήσεις και τις αναφορές των μέσων κοινωνικής δικτύωσης, ακόμη και τις πηγές βίντεο για να προβλέψουν εάν η ασθένεια έχει τη δυνατότητα να αναπτυχθεί εκτός ελέγχου.

## *3.3 Προκλήσεις της Μηχανικής Μάθησης στην Υγεία*

### *3.3.1 Έλλειψη ποιοτικών δεδομένων για τη δημιουργία ακριβών αλγορίθμων*

Τα αποτελέσματα που λαμβάνετε από τους αλγόριθμους μηχανικής εκμάθησης εξαρτώνται από την ποιότητα των δεδομένων που τοποθετούνται σε αυτούς. Δυστυχώς, τα ιατρικά δεδομένα δεν είναι πάντα τόσο ακριβή και τυποποιημένα όσο συχνά χρειάζεται. Υπάρχουν κενά στα αρχεία, ανακρίβειες στα προφίλ και άλλες δυσκολίες.

Συνολικά, τα ηλεκτρονικά αρχεία υγείας δεν κατασκευάστηκαν για να χρησιμοποιηθούν ως πηγή δεδομένων για έναν αλγόριθμο. Επομένως, προτού εφαρμοστεί ένα εργαλείο μηχανικής εκμάθησης, θα πρέπει να αφιερωθεί χρόνος για τη συλλογή, τον καθαρισμό, την επικύρωση και τη δόμηση δεδομένων για τους σκοπούς του.

### *3.3.2 Δημιουργία εργαλείων ML φιλικά προς την ιατρική ροή εργασιών*

Υπάρχουν πολλές εξαιρετικά συγκεκριμένες περιπτώσεις χρήσης μηχανικής εκμάθησης που μπορούν να βοηθήσουν στη διάγνωση και τη θεραπεία ασθενών. Ακόμα κι αν ένα εργαλείο ML λειτουργεί καλά στη θεωρία, δεν σημαίνει απαραίτητα ότι θα υιοθετηθεί από τους γιατρούς. Επομένως, είναι ζωτικής σημασίας να αναπτυχθούν και να διατεθούν εργαλεία μηχανικής εκμάθησης που θα ήταν διαισθητικά και εύχρηστα στην καθημερινή ροή ιατρικών εργασιών. Χωρίς την απαραίτητη ανατροφοδότηση από τους ανθρώπους που θα εργαστούν με το εργαλείο, δεν θα είναι τόσο αποτελεσματικό και οι επαγγελματίες δεν θα το εμπιστεύονται.

### *3.3.3 Συγκέντρωση μεγάλων ομάδων με ευρεία σετ δεξιοτήτων σε ένα μέρος*

Εκτός από τους πρακτικούς ειδικούς στον τομέα της υγείας, μια αποτελεσματική ομάδα ανάπτυξης μηχανικής μάθησης θα πρέπει να περιλαμβάνει τέτοιους ρόλους:

Επιχειρηματικός αναλυτής
Αρχιτέκτονας δεδομένων
Μηχανικός δεδομένων
Επιστήμονας δεδομένων
Ειδικός μηχανικής μάθησης.

Επιπλέον, είναι σημαντικό να διευκολυνθούν οι διαδικασίες αποτελεσματικής συνεργασίας στην ομάδα, ώστε να είναι δυνατή η απόδοση αξίας και η απόδειξη της βιωσιμότητας του προϊόντος με την πρώτη ευκαιρία.

## *3.3 Ηθική της χρήσης της Μηχανικής Μάθησης στην Υγεία*

Η χρήση της τεχνητής νοημοσύνης αποτελεί πηγή ηθικών διλημμάτων για μεγάλο χρονικό διάστημα. Ωστόσο, ορισμένα από αυτά είναι ειδικά για τη χρήση της μηχανικής μάθησης στην υγειονομική περίθαλψη. Αναφέρονται ορισμένες από τις πιο αξιόλογες περιπτώσεις:

### *3.3.1* Απόρρητο και Ασφάλεια Δεδομένων

Στις ΗΠΑ κανονισμοί απορρήτου διασφαλίζουν την ασφάλεια των πληροφοριών του ασθενούς. Όλοι θα πρέπει να έχουν το δικαίωμα να κρατούν ιδιωτικές πληροφορίες σχετικά με την υγεία τους. Ωστόσο, πολλές διαρροές δεδομένων υγειονομικής περίθαλψης συμβαίνουν καθημερινά και έχουν ως αποτέλεσμα κυρώσεις έως και 16 εκατομμυρίων δολαρίων για τους παρόχους υγειονομικής περίθαλψης. Τα δεδομένα είναι το αίμα του οργανισμού μηχανικής μάθησης. Πώς μπορούν να συνυπάρχουν αποτελεσματικά αυτά τα σημεία;

Αυτή η πρόκληση είναι δύσκολο να ξεπεραστεί. Στις περισσότερες περιπτώσεις, η μηχανική μάθηση δεν απαιτεί πλήρες φάσμα πληροφοριών για τον ασθενή (όπως όνομα, email, αριθμό τηλεφώνου και αριθμό ασφαλιστηρίου συμβολαίου). Έτσι, μπορεί να ανωνυμοποιηθεί αποτελεσματικά, έτσι ώστε η ταυτότητα του ατόμου να μην μπορεί να αποκαλυφθεί, ενώ η ακρίβεια του αλγορίθμου ML δεν θα μειωθεί. Από την αντίθετη πλευρά πρέπει να εφαρμοστούν ειδικές προσεγγίσεις ασφάλειας δεδομένων για να διασφαλιστεί η ανωνυμία των ασθενών.

### *3.3.2* Θέματα Αυτονομίας

Η μηχανική μάθηση μπορεί να χρησιμοποιηθεί αποτελεσματικά για να βοηθήσει τους ηλικιωμένους και τα άτομα με ψυχολογικά προβλήματα να λάβουν αποφάσεις για τη βελτίωση της υγείας τους. Αυτό αφορά τη λήψη των σωστών φαρμάκων, τη δημιουργία υγιεινών συνηθειών και την παραπομπή στον ειδικό όποτε χρειάζεται.

Ωστόσο, το ηθικό ζήτημα πίσω από αυτό είναι ότι οι άνθρωποι δυνητικά θα εγκαταλείψουν την αυτονομία τους και θα ενεργήσουν όπως τους λένε. Περιορίζει το εύρος πιθανών επιλογών τους σε ορισμένες προτεινόμενες επιλογές. Επομένως, θα πρέπει να παρέχεται μια σαφής ισορροπία μεταξύ των οδηγιών από τον αλγόριθμο και της ελευθερίας της προσωπικής επιλογής.

### *3.3.3* Ασφάλεια Ασθενούς

Οι αποφάσεις που λαμβάνονται από τον αλγόριθμο μηχανικής μάθησης βασίζονται πλήρως στα δεδομένα στα οποία έχει μάθει. Εάν η εισαγωγή είναι αναξιόπιστη ή εσφαλμένη, το αποτέλεσμα θα είναι επίσης λάθος. Η εσφαλμένη απόφαση μπορεί να βλάψει τον ασθενή ή ακόμη και να προκαλέσει τον θάνατό του.

Το ηθικό δίλημμα εδώ είναι ποιος θα είναι υπεύθυνος για τον θάνατο ενός ασθενούς λόγω της απόφασης που λαμβάνεται από έναν αλγόριθμο; Αυτή τη στιγμή, αυτό παραμένει ένα ανοιχτό ερώτημα. Η τελική απόφαση για τη μέθοδο θεραπείας βρίσκεται πίσω από τον ασθενή, ο οποίος θα πρέπει να ενημερωθεί για όλα τα οφέλη και τους κινδύνους κάθε θεραπευτικής μεθόδου.

### *3.3.4* Διαφάνεια και Ενημερωμένη Συναίνεση

Οι αλγόριθμοι μηχανικής μάθησης βασίζονται σε δεδομένα. Όσο περισσότερα σχετικά δεδομένα είναι διαθέσιμα, τόσο καλύτερα λειτουργούν και τόσο πιο ακριβή αποτελέσματα και προβλέψεις θα μπορούσαν να επιτευχθούν.

Πολλές χώρες έχουν νομοθεσία που περιορίζει τη χρήση δεδομένων ασθενών χωρίς την ενημερωμένη συγκατάθεσή τους. Έτσι, η χρήση της μηχανικής μάθησης στην υγειονομική περίθαλψη θα πρέπει να συνοδεύεται από την ενημέρωση του ασθενούς σχετικά με αυτήν και σχετικά με τις προσπάθειες ασφάλειας δεδομένων που εφαρμόζονται για τη διατήρηση των δεδομένων του ασφαλή.

# ΣΥΜΠΕΡΑΣΜΑΤΑ

Η μηχανική μάθηση έχει ήδη πολλές αποτελεσματικές χρήσεις στον κλάδο της υγειονομικής περίθαλψης, αλλά έχει επίσης τη δυνατότητα να κάνει πολλά περισσότερα. Εκτός από τη διάγνωση ασθενών και την ανάπτυξη θεραπείας, μπορεί να χρησιμοποιηθεί για τη βελτίωση της ιατρικής περίθαλψης, την πρόβλεψη των αποτελεσμάτων και ακόμη και τη βοήθεια σε χειρουργικές επεμβάσεις.

Ενώ η μηχανική μάθηση έχει πολλές δυνατότητες στον κλάδο της υγειονομικής περίθαλψης, συνοδεύεται επίσης από ορισμένες προκλήσεις, όπως η ποιότητα των δεδομένων υγειονομικής περίθαλψης, η κατασκευή προϊόντων φιλικών για τους γιατρούς και η συγκέντρωση μιας τεράστιας ομάδας ειδικών δεδομένων. Υπάρχουν επίσης ορισμένες ηθικές ανησυχίες, συμπεριλαμβανομένης της ασφάλειας και της λογοδοσίας των ασθενών. Παρά ορισμένες προκλήσεις, τα οφέλη του ML στην υγειονομική περίθαλψη τα υπερτερούν σημαντικά.

Το επόμενο βήμα είναι το σύστημα απόφασης κλινικής υποστήριξης (clinical decision support) το οποία θα μπορεί να διαχειρίζεται συγκεκριμένες καταστάσεις ή τύπους ασθενών, συστάσεις και βάσεις δεδομένων που μπορούν να παρέχουν πληροφορίες σχετικές με συγκεκριμένους ασθενείς, υπενθυμίζοντας για προληπτική φροντίδα και ειδοποιώντας για δυνητικά επικίνδυνες καταστάσεις.

# ΒΙΒΛΙΟΓΡΑΦΙΑ

[1]PAVLYSHENKO, B. 2016. MACHINE LEARNING, LINEAR AND BAYESIAN MODELS FOR LOGISTIC REGRESSION IN FAILURE DETECTION PROBLEMS.

[2]SELVAKUBERAN, K., KAYATHIRI, D., HARINI, B., DEVI, INDRA M., 2011. AN EFFICIENT FEATURE SELECTION METHOD FOR CLASSIFICATION IN HEALTH CARE SYSTEMS USING MACHINE LEARNING TECHNIQUES.

[3] MULLER, A. C., GUIDO, SARAH (2017). INTRODUCTION TO MACHINE LEARNING WITH PYTHON: A GUIDE FOR DATA SCIENTISTS, O'REILLY MEDIA, INC.,1005 GRAVENSTEIN HIGHWAY NORTH, SEBASTOPOL, CA 95472.

[4] MITCHELL, T. M. (1997). MACHINE LEARNING, MCGRAW-HILL SCIENCE/ENGINEERING/MATH.

[11] AWAD. MARIETTE, K. R. (2015). EFFICIENT LEARNING MACHINES: THEORIES, CONCEPTS, AND APPLICATIONS FOR ENGINEERS AND SYSTEM DESIGNERS, APRESS, BERKELEY, CA.

[5] RUSSEL, S., NORVIG, PETER (2005). ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΜΙΑ ΣΥΓΧΡΟΝΗ ΠΡΟΣΕΓΓΙΣΗ ΔΕΥΤΕΡΗ ΕΚΔΟΣΗ, ΚΛΕΙΔΑΡΙΘΜΟΣ.

[6] GRUS, J. (2015). DATA SCIENCE FROM SCRATCH, O'REILLY MEDIA, INC., 1005 GRAVENSTEIN HIGHWAY NORTH, SEBASTOPOL, CA 95472.

[7] TJARDO D MAARSEVEEN , BSC; TIMO MEINDERINK , MSC; MARCEL J T REINDERS , PHD; JOHANNES KNITZA , MD; TOM W J HUIZINGA 1, MD; ARND KLEYER 2, 3, MD; DAVID SIMON 2, 3, MD; ERIK B VAN DEN AKKER 4, 5, PHD; RACHEL KNEVEL 1, 6, MD, MACHINE LEARNING ELECTRONIC HEALTH RECORD IDENTIFICATION OF PATIENTS WITH RHEUMATOID ARTHRITIS: ALGORITHM PIPELINE DEVELOPMENT AND VALIDATION STUDY

[8] WEI, S., ZHAO, XUEJIAO., MIAO, CHUNYAN 2018. A COMPREHENSIVE EXPLORATION TO THE MACHINE LEARNING TECHNIQUES FOR DIABETES IDENTIFICATION.

[9] BRADLEY, A. P. 1996. THE USE OF THE AREA UNDER THE ROC CURVE IN THE EVALUATION OF MACHINE LEARNING ALGORITHMS.

# ΠΑΡΑΡΤΗΜΑ :ΚΩΔΙΚΑΣ ΠΡΟΓΡΑΜΜΑΤΟΣ

```
SEED = 26062019

test_size = 0.5

import sys
sys.path.append('src/')
import scipy
from yellowbrick import __version__ as yb_vers
from sklearn import __version__ as sk_vers
import NLP_functions as func
import matplotlib.pyplot as plt
import random
import numpy as np
import pandas as pd
from sklearn import datasets, svm, model_selection, tree, preprocessing,
metrics
import sklearn.ensemble as ske
from matplotlib import __version__ as m_vers
import re
import random
import pickle
import seaborn as sns
import pyxdameraulevenshtein as pyx # can be used for typo detection

print('Current versions Modules:\n')
print('Pandas version:\t\t' + pd.__version__)
print('Matplotlib version:\t' + m_vers)
print('numpy version:\t\t' + np.__version__)
print('sklearn version:\t' + sk_vers)
print('scipy version:\t\t' + scipy.__version__)
print('yellowbrick version:\t' + yb_vers)
print('seaborn version:\t' + sns.__version__)
print('pickle version:\t\t' + pickle.format_version)
Current versions Modules:

Pandas version:         1.1.4
Matplotlib version:     3.3.1
numpy version:          1.19.5
sklearn version:        1.0.2
scipy version:          1.4.1
yellowbrick version:    1.3.post1
seaborn version:        0.11.0
pickle version:         4.0
```

***Βήμα 1: Καθαρισμός δεδομένων με μεθόδους NLP***

Για να καθαρίσουμε τα δείγματα δεδομένων μας, θα εφαρμόσουμε τους ακόλουθους αλγόριθμους επεξεργασίας φυσικής γλώσσας:

• Μετατροπή του συνόλου δεδομένων που βασίζεται σε εισαγωγές σε σύνολο δεδομένων βάσει ασθενών

• Τμηματοποίηση λέξεων: οι λέξεις χωρίζονται σε κενά -> οι ειδικοί χαρακτήρες αφαιρούνται

• Προέλευση (ή Λεμματοποίηση): οι λέξεις επιστρέφουν στη ρίζα (αν η λέξη είναι στο σώμα) - προεπιλογή= Ολλανδικά (nl)

• Τυπική διόρθωση

• Αφαίρεση αντικειμένων XML

Αυτές οι λειτουργίες μπορεί να είναι χρήσιμες, ανάλογα με την ποιότητα των δεδομένων

## *1.1 Άνοιγμα raw file*

Ο πίνακας δειγμάτων μας αποτελείται από τρεις στήλες χωρισμένες με ερωτηματικό (';'). Έχουμε τις παρακάτω στήλες:

• PATNR= αναγνωριστικό ασθενούς

• text= πεδίο ελεύθερου κειμένου από το EMR (Συμπέρασμα)

• σχολιασμός = σχετική ετικέτα:

- 'True' = RA-περίπτωση
- 'False' = δεν είναι περίπτωση RA

In [3]:

```
radiag_df = pd.read_csv(r'sample_data/dummydata.csv', sep=';')
print('nr of entries: ', len(radiag_df))
radiag_df.head()
nr of entries:  2000
```

Out[3]:

| | PATNR | annotation | text |
|---|---|---|---|
| **0** | 474.0 | False | normocytic anemia no hemoly antibody anf and... |
| **1** | 2.0 | True | supplementation polyarthritis ikv rf and anti ccp posi... |
| **2** | 423.0 | False | with ncm functional limitation li hand not explained... |

| | PATNR | annotation | text |
|---|---|---|---|
| **3** | 40.0 | True | with seropositive erosive destructive ra for which... |
| **4** | 286.0 | True | clinical for others systemic disease sl sclerod... |

## *1.2 Συγχώνευση στο αναγνωριστικό ασθενούς (patient id)*

Συμπιέστε τα πεδία ελεύθερου κειμένου ανά ασθενή

In [4]:

```
id_column='PATNR'
X_column='text'
y_column="annotation"

print('[BEFORE] nr of entries:', len(radiag_df), '\tnr of patients:',
len(radiag_df['PATNR'].unique()))
radiag_df = func.mergeOnColumn(radiag_df, id_column, X_column, y_column)
print('[AFTER] nr of entries:', len(radiag_df), '\tnr of patients:',
len(radiag_df['PATNR'].unique()))
[BEFORE] nr of entries: 2000   nr of patients: 668
[AFTER] nr of entries: 668     nr of patients: 668
```

## *1.3 Μετονομασία στηλών και ετικετών*

1. Θα αλλάξουμε τα ονόματα στηλών ' annotation' σε ' Outcome' και ' text' σε ' Text'. Από εδώ και πέρα μπορούμε να αναφερόμαστε μόνο στα προσαρμοσμένα ονόματα στηλών.
2. Επιπλέον, θα εφαρμόσουμε ένα διαφορετικό πρότυπο για τα δεδομένα της ετικέτας (TRUE -> y και FALSE -> n)

In [5]:

```
radiag_df = radiag_df.rename(columns={y_column: "Outcome", X_column:
"Text"})
radiag_df['Outcome'] = radiag_df['Outcome'].apply(lambda x: 'y' if x ==
True else 'n')
radiag_df[['Text', 'Outcome']].head()
```

Out[5]:

| | Text | Outcome |
|---|---|---|
| **0** | normocytic anemia no hemoly antibody anf... | n |
| **1** | supplementation polyarthritis ikv rf and anti ccp pos... | y |
| **2** | with ncm functional limitation li hand not explain... | n |
| **3** | with seropositive erosive destructive ra which... | y |
| **4** | clinical for others systemic disease sl sclero... | y |

## *1.4 Προεπεξεργασία - Τμηματοποίηση λέξεων ( + Διόρθωση τυπογραφικού λάθους)*

FYI: 1. Το βήμα διόρθωσης τυπογραφικών σφαλμάτων είναι επί του παρόντος σε σίγαση σκόπιμα, επειδή αυτό το βήμα ισχύει μόνο εάν εργάζεστε με δεδομένα + η υλοποίηση της python είναι αρκετά αργή (άρα ίσως προτιμάτε να ελέγξετε για τυπογραφικά λάθη με άλλο εργαλείο).

In [6]:

```
radiag_df['Text'] = radiag_df['Text'].apply(lambda x :
func.processArtefactsXML(str(x)))
radiag_df['Text'] = radiag_df['Text'].apply(lambda x :
func.simpleCleaning(x, stem=False))

# Apply TypoCorrection
  #  import time
  #  typocor = func.TypoCorrection(np.array(l_new)) # provide word list
  #  t0 = time.time()
  #  radiag_df['Text'] = radiag_df['Text'].apply(lambda x :
typocor.correct(x))
  #  t1 = time.time()
  #  print('Time for TypoCorrection (n=' + str(len(radiag_df)) + ') : ' +
str(t1-t0))
```

## *1.5 Προεπεξεργασία - Στέλεχος ή ληματοποίηση*

Κανονικοποιούμε τις λέξεις πίσω στη μορφή «ρίζα» (αφαιρώντας τις ποικίλες εγκλίσεις). Αυτό διασφαλίζει ότι τα σημασιολογικά παρόμοια χαρακτηριστικά αντιμετωπίζονται με τον ίδιο τρόπο. Με αυτήν την προσέγγιση μειώνουμε αποτελεσματικά τον αριθμό των διαστάσεων

## *Install NLTK stemmer*

Όπως μπορείτε να δείτε το NLTK υποστηρίζει μυριάδες γλώσσες. Εάν θέλετε να χρησιμοποιήσετε ένα stemmer, φροντίστε να αποσχολιάσετε τη γραμμή! (και μην ξεχάσετε να εγκαταστήσετε το nltk με pip)

In [17]:

```
#!pip install -U nltk
#from nltk.stem.snowball import SnowballStemmer
#stemmer = SnowballStemmer("dutch") # Dutch
#stemmer = SnowballStemmer("arabic") # Arabic
#stemmer = SnowballStemmer("danish") # Danish
#stemmer = SnowballStemmer("english") # English
#stemmer = SnowballStemmer("finnish") # Finnish
#stemmer = SnowballStemmer("french") # French
#stemmer = SnowballStemmer("german") # German
#stemmer = SnowballStemmer("hungarian") # Hungarian
#stemmer = SnowballStemmer("italian") # Italian
#stemmer = SnowballStemmer("norwegian") # Norwegian
#stemmer = SnowballStemmer("portuguese") # Portuguese
#stemmer = SnowballStemmer("romanian") # Romanian
#stemmer = SnowballStemmer("russian") # Russian
#stemmer = SnowballStemmer("spanish") # Spanish
#stemmer = SnowballStemmer("swedish") # Swedish
```

## *Εισαγωγή προσαρμοσμένου stemmer για άλλες γλώσσες (άλλο παράδειγμα)*

In [ ]:

```
#!pip install -U greek-stemmer # Greek
#from greek_stemmer import GreekStemmer
#stemmer = GreekStemmer() # Dutch
```

### #1.5.1 Εκτελέστε stemming (Παράδειγμα στα Ολλανδικά)

In [ ]:

```
#!pip install -U nltk
from nltk.stem.snowball import SnowballStemmer
stemmer = SnowballStemmer("dutch") # Dutch
```

In [16]:

```
radiag_df['Text'] = radiag_df['Text'].apply(lambda x : func.stemmingText(x,
stemmer=stemmer))
radiag_df=radiag_df.fillna('')

# Save autocorrected dataframe
radiag_df.to_csv(r'sample_data/autocorrect_dummydata.csv', sep='|',
index=False)
```

Out[16]:

| | Text | PATNR | Outcome |
|---|---|---|---|
| **0** | normocytair anemie gen hemoly antistof anf en... | 474.0 | n |
| **1** | suppletie polyartritis ikv rf en anti ccp pos... | 2.0 | y |
| **2** | bij ncm functiebeperk li hand niet verklar do... | 423.0 | n |
| **3** | met seropositief erosief destructief ra waarv... | 40.0 | y |
| **4** | klinisch vor ander systeemziekt sl scleroderm... | 286.0 | y |
| **...** | ... | ... | ... |
| **663** | artralgie hand voet vnl na gebruik dd toch oa... | 334.0 | y |
| **664** | weinig mogelijk medicatie gat stop met oac ac... | 515.0 | n |
| **665** | beschouw als ongedifferentieerd artritis dd s... | 0.0 | y |
| **666** | nieuw ontstanen koortsperiod met dyspnoe dysp... | 605.0 | n |

| | Text | PATNR | Outcome |
|---|---|---|---|
| **667** | prednison monotherapie reager aanvankelk slec... | 514.0 | n |

668 rows × 3 columns

### *Βήμα 2: Δόμηση των δεδομένων*

### *2.1 Ανοίξτε το προεπεξεργασμένο αρχείο*

Μπορείτε να παραλείψετε τα παραπάνω βήματα εάν έχετε ήδη ένα προεπεξεργασμένο αρχείο

In [15]:
```
radiag_df = pd.read_csv(r'sample_data/autocorrect_dummydata.csv', sep='|')
radiag_df=radiag_df.fillna('')
radiag_df[['Text', 'Outcome']].head()
```

Out[15]:

| | Text | Outcome |
|---|---|---|
| **0** | normocytair anemie gen hemoly antistof anf en... | n |
| **1** | suppletie polyartritis ikv rf en anti ccp pos... | y |
| **2** | bij ncm functiebeperk li hand niet verklar do... | n |
| **3** | met seropositief erosief destructief ra waarv... | y |
| **4** | klinisch vor ander systeemziekt sl scleroderm... | y |

### *2.2 Εξισορρόπηση τάξεων σε δεδομένα (αναλογία περιπτώσεων RA έναντι μη περιπτώσεων) - Προαιρετικό*

Ο αριθμός των μη υποθέσεων είναι μεγαλύτερος από τον αριθ. των περιπτώσεων έτσι επιλέγεται ένα τυχαίο δείγμα μη περιπτώσεων για να δημιουργηθεί μια ισορροπία μεταξύ των κλάσεων.

• Προαιρετικά: Το εάν θέλετε ή όχι να εξισορροπήσετε τις τάξεις εξαρτάται από την αναμενόμενη επικράτηση των περιπτώσεων στα δεδομένα της δοκιμής.

• Θα μπορούσατε επίσης να επιλέξετε να εξισορροπήσετε τις τάξεις με διαφορετικό τρόπο (π.χ.: 1:5), αλλά θα πρέπει πάντα να αντιστοιχεί με την αναμενόμενη επικράτηση στο σετ δοκιμών. Επιλέξτε τον επιθυμητό αριθμό δειγμάτων αλλάζοντας το nr_of_samples
FYI: τα δεδομένα δεν είναι εξισορροπημένα από προεπιλογή

In [10]:

```
balance = False
nr_of_samples = len(radiag_df[radiag_df['Outcome']=='y'])

if (balance):
    df_no_outcome =
radiag_df[radiag_df['Outcome']=='n'].sample(n=nr_of_samples,
random_state=SEED)
    equal_radiag_df = pd.concat([df_no_outcome,
radiag_df[radiag_df['Outcome']=='y']])
    radiag_df = equal_radiag_df.sample(frac=1, random_state=SEED)
```

### *2.3 Αφαιρέστε stop words*

In [11]:

```
from nltk.corpus import stopwords

stop_words = stopwords.words('dutch')
radiag_df['Text'] = radiag_df['Text'].apply(lambda x: ' '.join([item for
item in x.split(' ') if item not in stop_words]))
```

### *2.4 Διαχωρισμός συνόλου σχολιασμών τόσο στα δεδομένα κειμένου όσο και στην αντίστοιχη ετικέτα (Y)*

In [12]:

```
radiag_df = radiag_df.sample(frac=1, random_state=SEED) # random shuffle

X = radiag_df['Text'].values
y = radiag_df['Outcome'].values
y_b = np.array([func.binarize(val) for val in y])
```

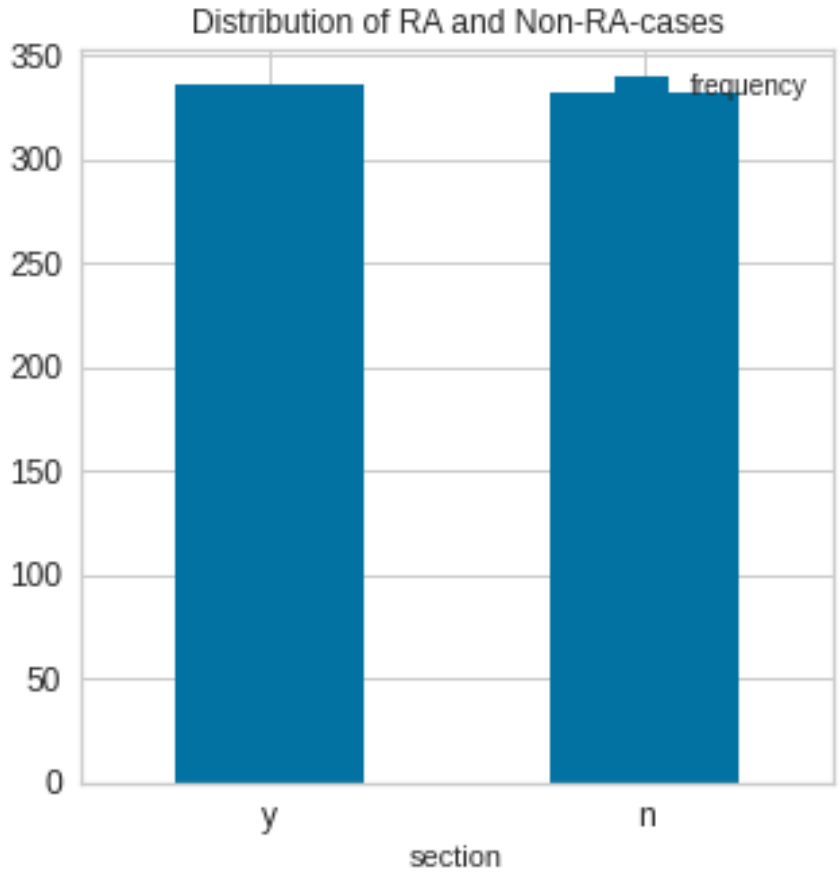
### *2.5 Οπτικοποίηση των δεδομένων*

### Plot: Prevalence of RA-cases

Θετικός έλεγχος -> οι τάξεις πρέπει να είναι σε ισορροπία

In [13]:

```
from collections import Counter

labels, values = zip(*Counter(y).items())
print(values)
df = pd.DataFrame({'section':labels, 'frequency':values})
ax = df.plot(kind='bar',  title ="Distribution of RA and Non-RA-cases",
figsize=(5, 5), x='section', legend=True, fontsize=12, rot=0)
(336, 332)
```

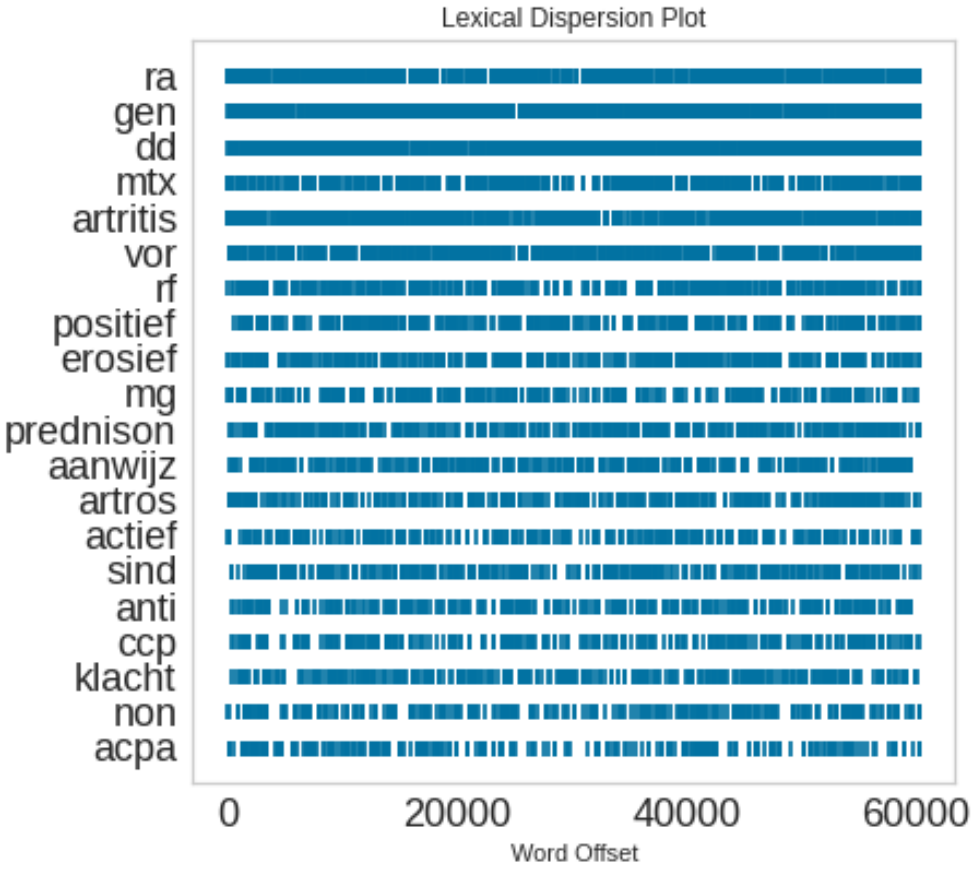



## Plot: Lexical Dispersion Plot

• Εμφανίζει την επικράτηση των χαρακτηριστικών
• Η ομοιογένεια του χαρακτηριστικού σε όλες τις εγγραφές

In [14]:

```
func.plotLexicalDispersion(X, nr_features=20, ngram_range=(1,1))
src/NLP_functions.py:573: VisibleDeprecationWarning: Creating an ndarray
from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-
or ndarrays with different lengths or shapes) is deprecated. If you meant
to do this, you must specify 'dtype=object' when creating the ndarray
  d = np.array(words)
/home/tdmaarseveen/.conda/envs/ra_clustering2/lib/python3.8/site-
packages/sklearn/utils/deprecation.py:87: FutureWarning: Function
get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and
will be removed in 1.2. Please use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
<ipython-input-14-4f591371587c>:1: FutureWarning: arrays to stack must be
passed as a "sequence" type such as list or tuple. Support for non-sequence
iterables such as generators is deprecated as of NumPy 1.16 and will raise
an error in the future.
  func.plotLexicalDispersion(X, nr_features=20, ngram_range=(1,1))
*c* argument looks like a single numeric RGB or RGBA sequence, which should
be avoided as value-mapping will have precedence in case its length matches
with *x* & *y*.  Please use the *color* keyword-argument or provide a 2-D
array with a single row if you intend to specify the same RGB or RGBA value
for all points.
```



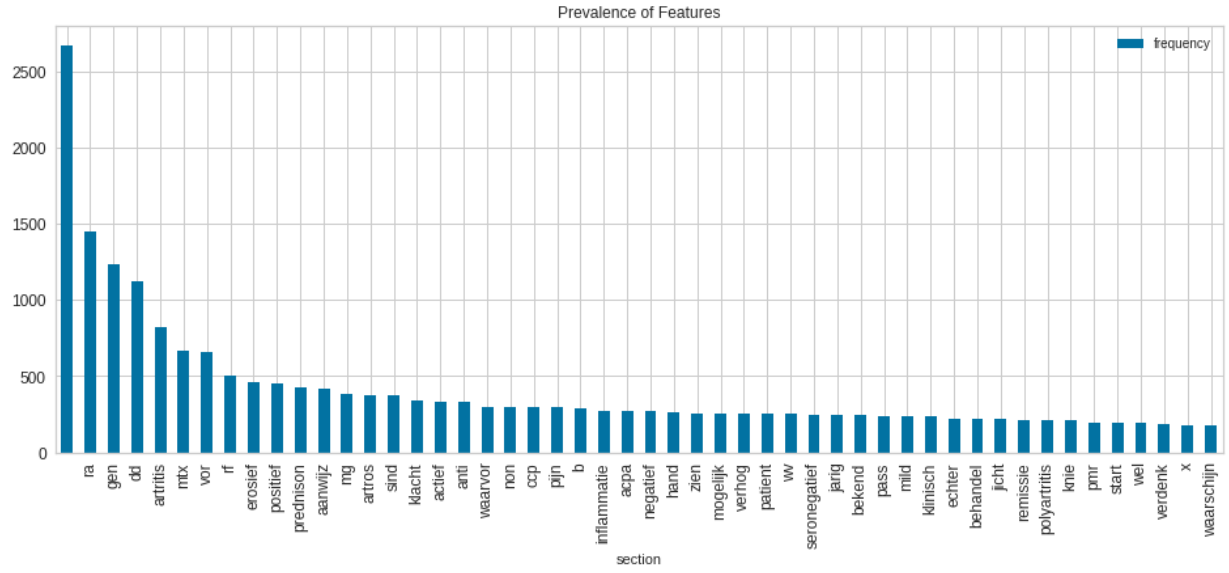

**Plot: Κατανομή χαρακτηριστικών**

In [15]:

```
func.plotSampleDistribution(X, nr_features=50)
```

Out[15]:

```
<module 'matplotlib.pyplot' from
'/home/tdmaarseveen/.conda/envs/ra_clustering2/lib/python3.8/site-
packages/matplotlib/pyplot.py'>
```



### *Βήμα 3: Κατασκευή μοντέλων αγωγών και τοποθέτησης*

εισάγετε όλους τους ταξινομητές που θέλετε να εφαρμόσετε/συγκρίνετε

### *3.1 Εισαγωγή διαφορετικών μοντέλων ταξινόμησης:*

Παρακάτω παρατίθενται τα ακόλουθα μοντέλα: Naive Bayes, Gradient Boosting, Neural Networks, Decision tree & SVM.

In [17]:

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
from sklearn.linear_model import SGDClassifier
from sklearn.svm import SVC
from sklearn.neural_network import MLPClassifier
from sklearn.neural_network import MLPRegressor
from sklearn import tree
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.dummy import DummyClassifier
```

### *3.2 Αρχικοποιήστε τα διαφορετικά μοντέλα*

## Αντιστοίχιση λέξεων

Η μέθοδος αντιστοίχισης λέξεων ταξινομείται με βάση την παρουσία των παρεχόμενων στόχων.

Παρέχονται οι ακόλουθοι στόχοι: «ρευματοειδής αρθρίτιδα», «ρευματοειδής αρθρίτιδα» και «ra»

In [18]:

```
import TextClassification as tc

l_targets = ['rheumatoid arthritis', 'reumatoide artritis', 'ra']

WordMatching = tc.CustomBinaryModel(l_targets)
```

## Αρχικοποίηση ταξινομητών

Όλες οι μέθοδοι εκτός από την απλή μέθοδο αντιστοίχισης λέξεων απαιτούν διανυσματικά δεδομένα. Ως εκ τούτου, οι ταξινομητές συνοδεύονται από μια συνάρτηση TfidfVetorizer

In [19]:

```
SEED=26062019
models = [
    # 0
    WordMatching,
    # Naive Bayes - 1
    Pipeline([('vect', TfidfVectorizer(ngram_range=(1, 3), stop_words =
stop_words)),
                ('clf', MultinomialNB()),
            ]),
    # Gradient Boosting - 2
    Pipeline([('vect', TfidfVectorizer(ngram_range=(1, 3), stop_words =
stop_words)),
                ('clf', ske.GradientBoostingClassifier(random_state=SEED))
            ]),
    # Neural Networks - 3
```

```
    Pipeline([('vect', TfidfVectorizer(ngram_range=(1, 3), stop_words =
stop_words)),
                ('clf', MLPClassifier(solver='lbfgs', random_state=SEED)),
# hidden_layer_sizes=(5, 2), ,
            ]),
    # Decision Tree - 4
    Pipeline([('vect', TfidfVectorizer(ngram_range=(1, 3), stop_words =
stop_words)),
                ('clf', tree.DecisionTreeClassifier(random_state=SEED)),
            ]),
    # SVM 5
    Pipeline([('vect', TfidfVectorizer(ngram_range=(1, 3), stop_words =
stop_words)),
                ('clf', SVC(kernel='linear', probability=True,
random_state=SEED)),
            ]),
    # Random Forest 6
    Pipeline([('vect', TfidfVectorizer(ngram_range=(1, 3), stop_words =
stop_words)),
                ('clf', RandomForestClassifier(random_state=SEED)), #
n_estimators=100, max_depth=2,
            ]),
    # Dummy 7
    Pipeline([('vect', TfidfVectorizer(ngram_range=(1, 3), stop_words =
stop_words)),
                ('clf', DummyClassifier(random_state=SEED,
strategy='stratified')),
            ]),
    ]
```

## Ονομασία των μοντέλων

1. Δώστε σε κάθε μία από τις μεθόδους σας ένα σχετικό ή αναγνωρίσιμο όνομα -> παρέχοντας μια λίστα.
2. Θα μπορούσατε επίσης να αντιστοιχίσετε την προτιμώμενη χρωματική παλέτα!
Προσοχή: κρατήστε την ίδια σειρά όπως παραπάνω

In [20]:

```
names = ['Word Matching', 'Naive Bayes', 'Gradient Boosting', 'Neural
Networks', 'Decision Tree', 'SVM', 'Random Forest', 'Dummy']
pal = ['r', 'c', 'b', 'g', 'y', 'magenta', 'indigo', 'black', 'orange']
```

## *3.3: Τοποθέτηση των ταξινομητών*

Τα μοντέλα μπορούν πλέον να εφαρμοστούν στα δεδομένα εκπαίδευσης.

### 3.3.1 Ρύθμιση παραμέτρων

- Εξετάστε το ενδεχόμενο να προσαρμόσετε τις προεπιλεγμένες παραμέτρους σύμφωνα με τις προτιμήσεις σας:
- seed: ψευδο-τυχαία γεννήτρια αριθμών, για εξασφάλιση αναπαραγωγής
- pathway : διαδρομή προς τον προτιμώμενο κατάλογο

- binarize label -> label θα αλλάξει είτε σε 0 είτε σε 1, επομένως θα πρέπει να αναφέρετε πώς κωδικοποιούνται οι αληθινές ετικέτες (προεπιλογή=y)
- παλέτα : λίστα χρωμάτων
- Επιπλέον, εφαρμόζεται η συνάρτηση assignPalette(): εκχωρεί αυτόματα έναν ταξινομητή σε ένα χρώμα, για να διασφαλίσει το ίδιο χρώμα σε πολλά διαγράμματα

In [22]:

```
import TextClassification as tc

tm = tc.TextClassification(X, y, models, names)

tm.setSeed(26062019)
tm.setOutputPath(r'output_files/')
tm.binarizeLabel(y, true_label='y')
tm.assignPalette()
{'Word Matching': 'r', 'Naive Bayes': 'y', 'Gradient Boosting': 'c',
'Neural Networks': 'b', 'Decision Tree': 'g', 'SVM': 'magenta', 'Random
Forest': 'indigo', 'Dummy': 'black'}
```

### 3.3.2 Επιλέξτε διαδικασία διασταυρούμενης επικύρωσης

Για να διασφαλίσετε μια ισχυρή αξιολόγηση απόδοσης, συνιστάται να υπολογίσετε την απόδοση σε πολλαπλές πτυχές ή/και γύρους. Παρέχουμε δύο διαδικασίες που χρησιμοποιούνται συνήθως παρακάτω:

**Πότε να χρησιμοποιείται 10 CV?**

Εάν θέλετε να έχετε μια αξιόπιστη προσέγγιση της απόδοσης του ταξινομητή σε αόρατα δεδομένα

**Πότε να χρησιμοποιείται 5x2 CV?**

Αν θέλετε να συγκρίνετε τους ταξινομητές δίκαια

In [23]:

```
approach = '5x2CV'

if approach == '10CV':
    tm.setRounds(1)
    tm.setFolds(10)
elif approach == '5x2CV':
    tm.setRounds(5)
    tm.setFolds(2)

tm.fitModels()
General settings for training/testing:
Method = Cross Validation 10-fold
        fraction test:  0.5

loading model:  Word Matching
Word Matching is assumed to be a word matching method and is therefore not
fitted
loading model:  Naive Bayes
loading model:  Gradient Boosting
loading model:  Neural Networks
loading model:  Decision Tree
```

```
loading model:  SVM
loading model:  Random Forest
loading model:  Dummy
```

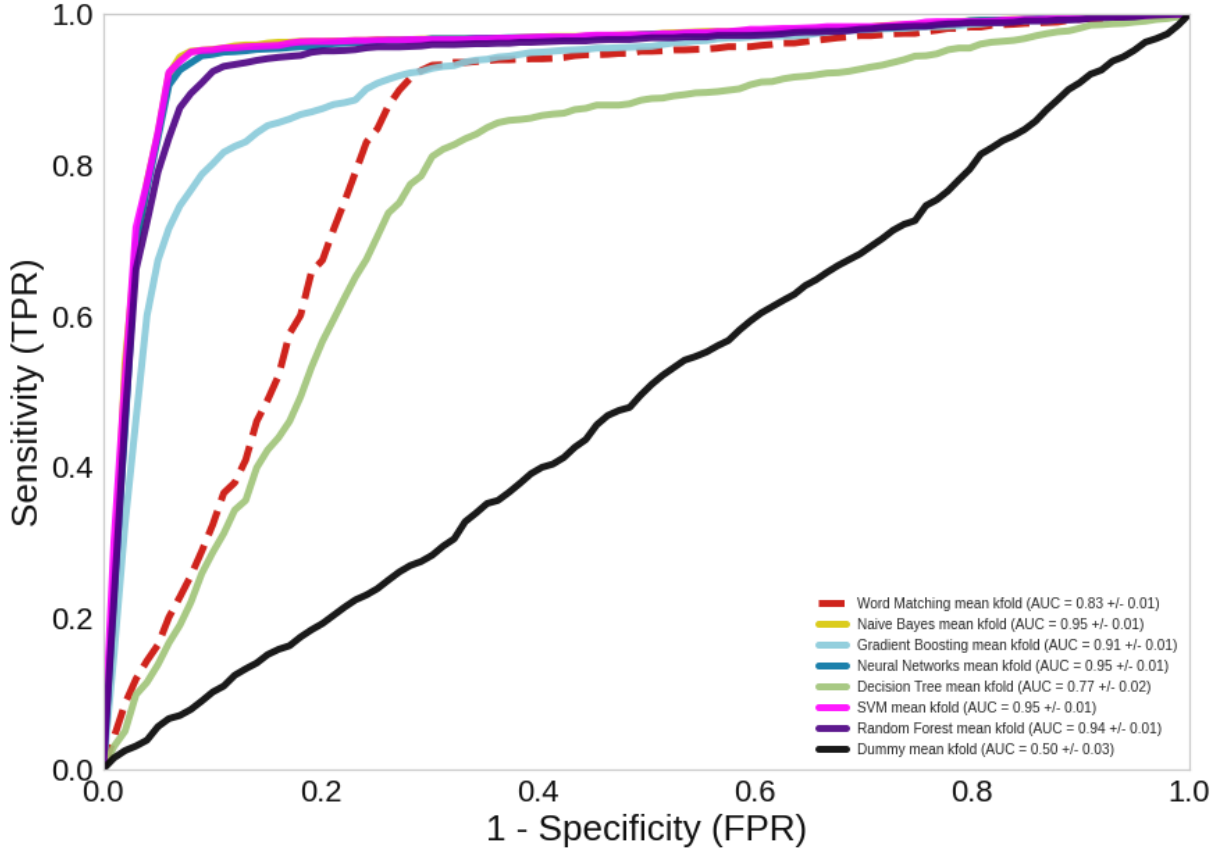
### *Βήμα 4: Αξιολογήστε την επιλογή μοντέλου*

### *4.1 ROC-AUC*

Οραματιστείτε την απόδοση των μοντέλων στην καμπύλη ROC

In [24]:

```
tm.setREF('Word Matching') # assign Word Matching as reference

plt = tm.plotROC()
plt.savefig('figures/results/ROC_curve_all_methods.png',
bbox_inches='tight')
Word Matching 0.8250898522653688 (std : +/-0.01 )
Naive Bayes 0.95270314547493 (std : +/-0.01 )
Gradient Boosting 0.9111147863850068 (std : +/-0.01 )
Neural Networks 0.9502552530902785 (std : +/-0.01 )
Decision Tree 0.7682277542855865 (std : +/-0.02 )
SVM 0.9536426717743117 (std : +/-0.01 )
Random Forest 0.9420791878024389 (std : +/-0.01 )
Dummy 0.4962128547576413 (std : +/-0.03 )
```




#### 4.1.1 Student's T-Test σε σχέση με το Naive Word Matching

Αξιολογήστε τη διαφορά στην απόδοση εφαρμόζοντας ένα t-test
• d_aucs = λεξικό με όλα τα aucs
• auc_ref = αναφορά auc (σε αυτήν την περίπτωση χρησιμοποιείται το auc της μεθόδου Word Matching)

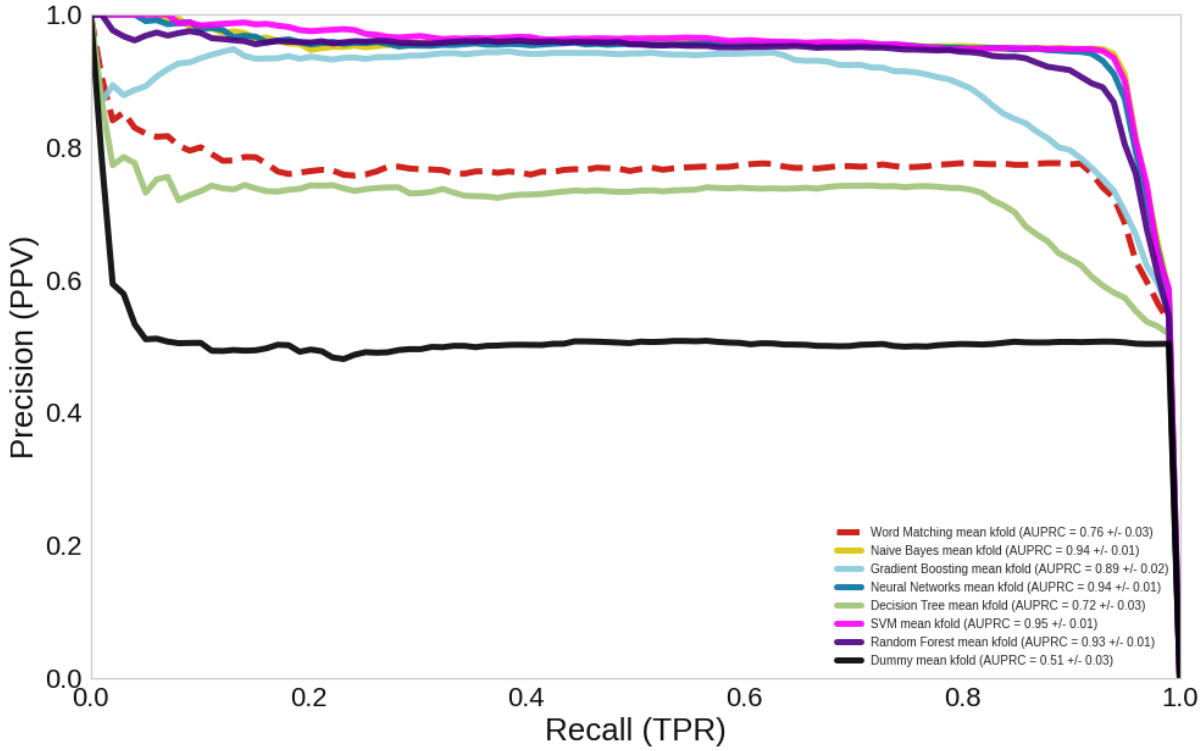
In [25]:

```
d_aucs = tm.getAUC()
auc_ref = d_aucs['Word Matching']
auc_models = d_aucs


for key in d_aucs.keys(): # classifiers with probabilities
    twosample_results = tm.ttest_5x2cv(auc_models[key], auc_ref)
    print(key, '\t(p = ' + '{:.1e}'.format(twosample_results[1]) + ')')
Word Matching  (p = 1.0e+00)
Naive Bayes    (p = 2.3e-05)
Gradient Boosting      (p = 2.0e-03)
Neural Networks        (p = 1.6e-05)
Decision Tree  (p = 4.6e-02)
SVM    (p = 1.1e-05)
Random Forest  (p = 5.9e-06)
Dummy  (p = 1.3e-04)
```

## *4.2. AUC - Καμπύλη ανάκλησης ακριβείας*

In [26]:

```
plt = tm.plotPrecisionRecall()
plt.savefig('figures/results/PR_curve_all_methods.png',
bbox_inches='tight')
Word Matching mean kfold (AUPRC = 0.76 +/- 0.03)
Naive Bayes mean kfold (AUPRC = 0.94 +/- 0.01)
Gradient Boosting mean kfold (AUPRC = 0.89 +/- 0.02)
Neural Networks mean kfold (AUPRC = 0.94 +/- 0.01)
Decision Tree mean kfold (AUPRC = 0.72 +/- 0.03)
SVM mean kfold (AUPRC = 0.95 +/- 0.01)
Random Forest mean kfold (AUPRC = 0.93 +/- 0.01)
Dummy mean kfold (AUPRC = 0.51 +/- 0.03)
```




### 4.2.2 Τεστ μαθητή σε σχέση με την αντιστοίχιση αφελών λέξεων (PR-AUC)

In [27]:

```
d_auprcs = tm.getAUPRC()
auprc_ref = d_auprcs['Word Matching']
auprc_models = d_auprcs

for key in d_aucs.keys(): # classifiers with probabilities
    twosample_results = tm.ttest_5x2cv(auprc_models[key], auprc_ref)
    print(key, '\t(p = ' + '{:.1e}'.format(twosample_results[1]) + ')')
Word Matching   (p = 1.0e+00)
Naive Bayes     (p = 4.4e-06)
Gradient Boosting       (p = 2.9e-05)
Neural Networks         (p = 5.2e-06)
Decision Tree   (p = 5.3e-03)
SVM     (p = 2.9e-06)
Random Forest   (p = 2.3e-06)
Dummy   (p = 8.1e-07)
```

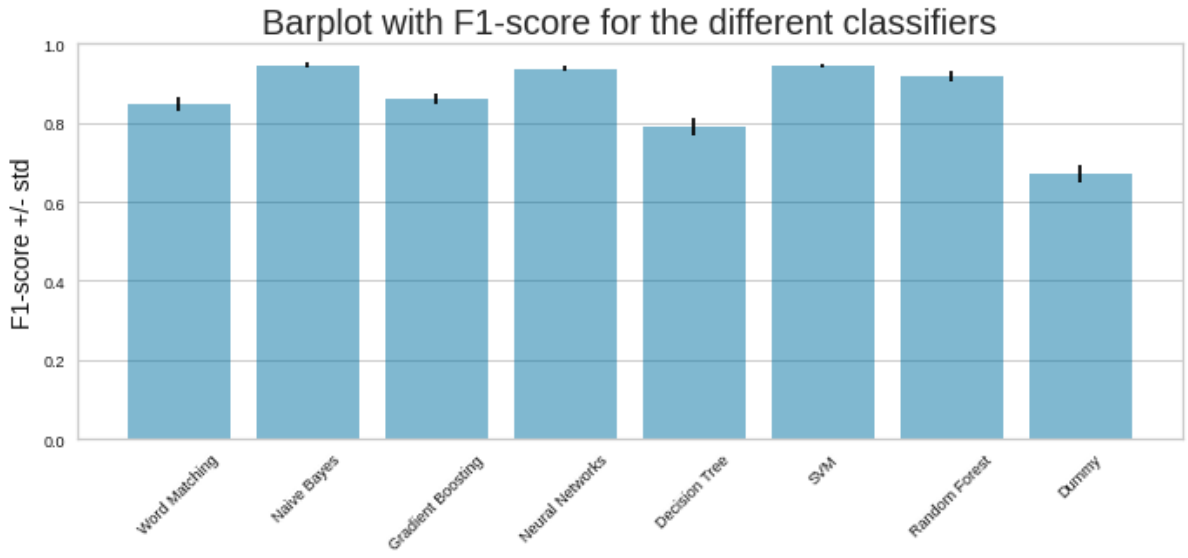
## *4.3 F1-measures*

Πολύτιμη μέτρηση όταν αντιμετωπίζετε ανισορροπία δεδομένων

In [28]:

```
import matplotlib.pylab as pylab
params = {'figure.figsize': (10,5), 'axes.titlesize':20,
'font.weight':'regular', 'xtick.labelsize':22, 'ytick.labelsize':22,
'axes.labelsize':14, 'font.size':22}#fontweight='bold'
pylab.rcParams.update(params)
```

```
plt = tm.plotF1scores(debug=True)
Word Matching : 0.85+/-0.02
Naive Bayes : 0.95+/-0.01
Gradient Boosting : 0.86+/-0.01
Neural Networks : 0.94+/-0.01
Decision Tree : 0.79+/-0.02
SVM : 0.94+/-0.01
Random Forest : 0.92+/-0.01
Dummy : 0.67+/-0.02
<Figure size 1008x1008 with 0 Axes>
```




### 4.3.2 Μαθητικό t-test για F1-scores

Συνιστάται μόνο εάν προσέγγιση = 5x2 cv για αξιόπιστο αποτέλεσμα

In [29]:

```
d_f1 = tm.getF1()
f1_ref = d_f1['Word Matching']

for key in d_f1.keys(): # classifiers with probabilities
    twosample_results = tm.ttest_5x2cv(d_f1[key], f1_ref)
    print(key, '\t(p = ' + '{:.1e}'.format(twosample_results[1]) + ')')
Word Matching   (p = 1.0e+00)
Naive Bayes     (p = 3.2e-03)
Gradient Boosting       (p = 9.3e-01)
Neural Networks         (p = 2.5e-03)
Decision Tree   (p = 2.6e-02)
SVM     (p = 1.8e-03)
Random Forest   (p = 7.7e-03)
Dummy   (p = 6.2e-05)
```

### *Βήμα 5: Διανομή μοντέλου - αποθήκευση μοντέλου ταξινόμησης*

Με το pickle μπορείτε να αποθηκεύσετε και να φορτώσετε τα μοντέλα
Πριν αποθηκεύσετε το μοντέλο σας, μπορείτε να χωρέσετε σε ολόκληρα τα δεδομένα, αυτή είναι κοινή πρακτική. Ωστόσο, εάν χρειάζεται πολύς χρόνος για να χωρέσει το μοντέλο, τότε μπορείτε επίσης να επιλέξετε να λάβετε απλώς τη διάμεση επανάληψη του καθορισμένου ταξινομητή από το διπλό βιογραφικό

In [32]:

```
best_model = tm.getTrainedClassifier('SVM')
```

```
best_model.fit(tm.X, tm.y) # fit on entire data
filename='savedModels/SVM.sav'
pickle.dump(best_model, open(filename, 'wb'))
```

## 5.1 Εισαγωγή εκπαιδευμένου μοντέλου ταξινόμησης

Εισαγάγετε το πρόσφατα αποθηκευμένο μοντέλο (μπορείτε να εφαρμόσετε αυτό το μοντέλο σε ένα νέο σύνολο χωρίς ετικέτα)

In [33]:

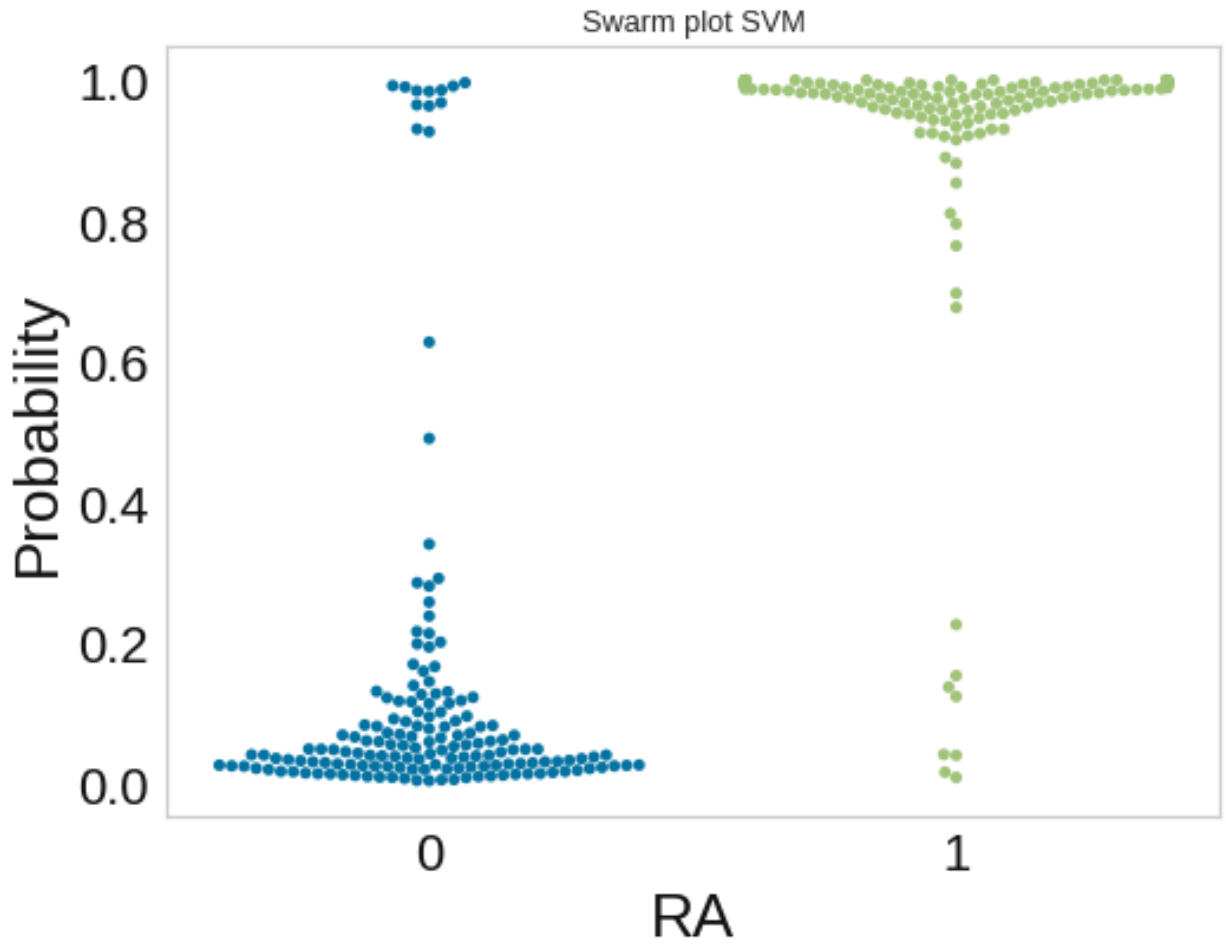
```
filename='savedModels/SVM.sav'
loaded_model = pickle.load(open(filename, 'rb'))
```

### *Βήμα 6: Αξιολογήστε το καλύτερο μοντέλο*

Οραματιστείτε την κατανομή πρόβλεψης (πιθανότητες) του ταξινομητή με μια γραφική παράσταση σμήνος.
• Θα μπορούσατε επίσης να σχεδιάσετε ένα διάγραμμα διασποράς (αποσχολιάζοντας την τρίτη γραμμή)

In [34]:

```
plt = tm.plotSwarm('SVM')
plt.savefig('figures/results/swarmplot_SVM.png', bbox_inches='tight')
# tm.plotScatter('SVM')
/home/tdmaarseveen/.conda/envs/ra_clustering2/lib/python3.8/site-
packages/seaborn/categorical.py:1296: UserWarning: 37.4% of the points
cannot be placed; you may want to decrease the size of the markers or use
stripplot.
  warnings.warn(msg, UserWarning)
```



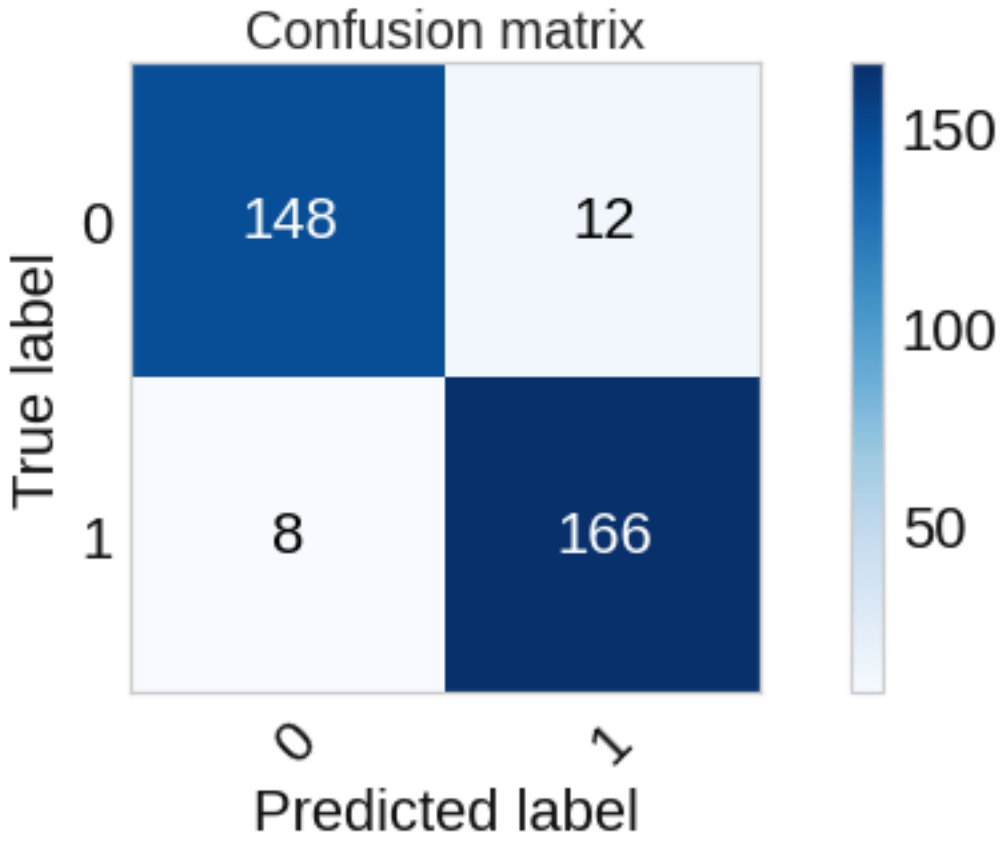

### *6.2 Ορίστε τη βέλτιστη αποκοπή*

#### Παράδειγμα: Βέλτιστο όριο όταν επιθυμείται Ευαισθησία 0,9

Επιστρέφει το όριο που αντιστοιχεί με ευαισθησία (TPR στον κωδικό) 0,9 και την υψηλότερη δυνατή ακρίβεια. Άλλες επιλογές είναι: tpr, ppv και f1

In [35]:

```
import matplotlib.pylab as pylab
params = {'figure.figsize': (10,5), 'axes.titlesize':20,
'font.weight':'regular', 'xtick.labelsize':22, 'ytick.labelsize':22,
'axes.labelsize':22, 'font.size':22}#fontweight='bold'
pylab.rcParams.update(params)

tm.getConfusionMatrix('SVM', desired=.9, most_val='tpr')
Generating confusion matrix for SVM based on median Iteration (AUPRC):  4
Other weighing variables:  ['prc']
Thresh: 0.68
PRC:    0.93
Sens:   0.95
Spec:   0.93
F1:     0.94
NPV:    0.9487179487179487
ACC:    0.9401197604790419
```

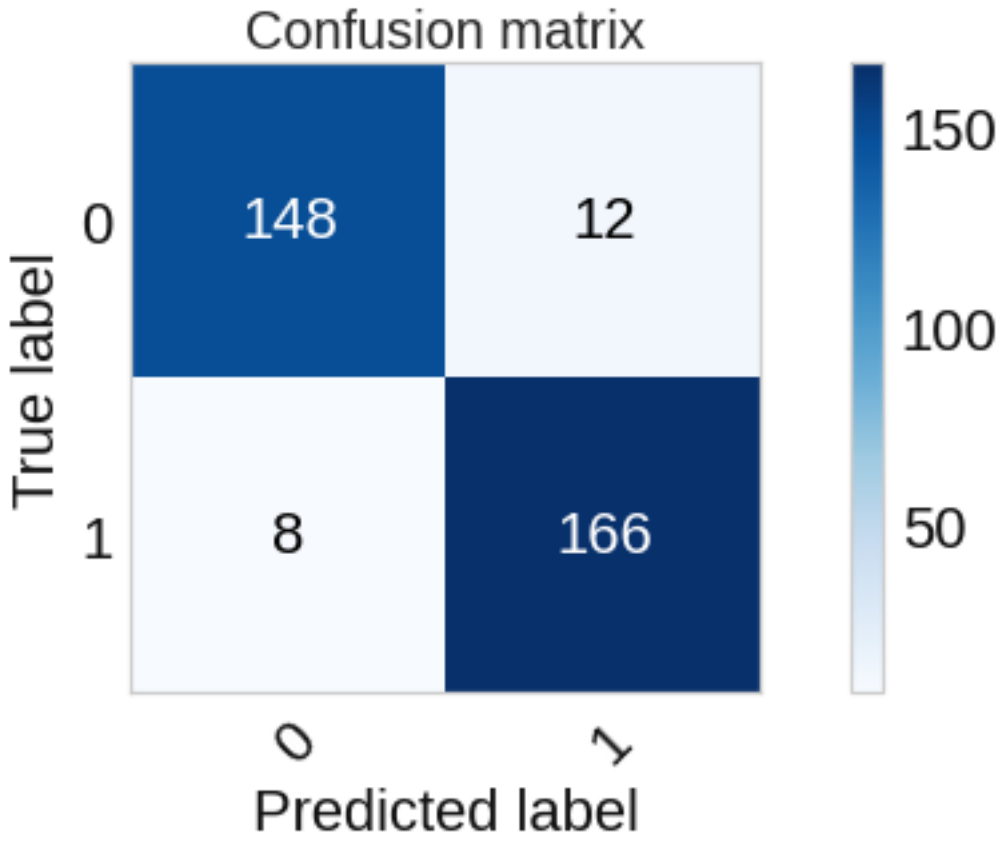



### Παράδειγμα: Βέλτιστη αποκοπή όταν επιθυμείται Ακρίβεια 0,9

Επιστρέφει το όριο που αντιστοιχεί με ακρίβεια 0,8 και την υψηλότερη δυνατή ευαισθησία

In [36]:

```
import matplotlib.pylab as pylab
params = {'figure.figsize': (10,5), 'axes.titlesize':20,
'font.weight':'regular', 'xtick.labelsize':22, 'ytick.labelsize':22,
'axes.labelsize':22, 'font.size':22}#fontweight='bold'
pylab.rcParams.update(params)

tm.getConfusionMatrix('SVM', desired=0.9, most_val='prc')
Generating confusion matrix for SVM based on median Iteration (AUPRC):  4
Other weighing variables:  ['tpr']
Thresh: 0.68
PRC:    0.93
Sens:   0.95
Spec:   0.93
F1:     0.94
NPV:    0.9487179487179487
ACC:    0.9401197604790419
```




### *[Επιπλέον ανάλυση] 6.3 Επίδραση του επιπολασμού στην Ανάκληση Ακριβείας*

Εφαρμογή ενόςtrainingsset με διαφορετικό επιπολασμό κάθε φορά

In [298]:

```
params = {'figure.figsize': (8,6), 'axes.titlesize':20,
'font.weight':'regular', 'xtick.labelsize':22, 'ytick.labelsize':22,
'axes.labelsize':22, 'font.size':22}#fontweight='bold'
pylab.rcParams.update(params)

plt, d_aucs = tm.plotPrevalencePR('SVM', cv=False)
```

### *[Επιπλέον ανάλυση] 6.4 Επίδραση του μεγέθους του δείγματος*

Αυξήστε σταδιακά το μέγεθος του δείγματος του σετ εκπαίδευσης με ένα προκαθορισμένο μέγεθος βημάτων. Επιπλέον, θα μπορούσατε επίσης να επιλέξετε να εφαρμόσετε διασταυρούμενη επικύρωση (cv) για πιο αξιόπιστες μετρήσεις.

**ROC**

In [299]:

```
plt, d_aucs = tm.samplingCurveROC('SVM', stepsize=100 , cv=True)
```

**Ανάκληση ακριβείας**

In [300]:

```
plt, d_aucs = tm.samplingCurvePR('SVM', stepsize=100 , cv=True)
```

### *6.5 Τα πιο σημαντικά χαρακτηριστικά*

### SVM

In [38]:

```
tm.plot_coefficients('SVM', top_features=10, chunks=3, negative=False)
```

**Βήμα 7: Χρησιμοποιήστε το μοντέλο σε ανεξάρτητο σετ δοκιμών**

Εφαρμόστε ταξινομητή σε ανεξάρτητο σύνολο δοκιμών για αμερόληπτη αξιολόγηση

• toDO: γραφική παράσταση μήτρας σύγχυσης με αντίστοιχες μετρήσεις με αποκοπή (Ppv, npv, sens, spec, f1, ακρίβεια)

• plot pr AUC & roc AUC

### *7.1 load & prepare test set*

### *Εισαγωγή δεδομένων δοκιμής & τελικό μοντέλο*

In [39]:

```
file_name='sample_data/dummydata_gold.csv'
model_name='savedModels/SVM.sav'

valid_df = pd.read_csv(file_name, sep=';')
loaded_model = pickle.load(open(model_name, 'rb'))
valid_df.head()
```

Out[39]:

| | PATNR | annotation | text |
|---|---|---|---|
| **0** | 51.0 | True | w folaatzuur mg w prednison mg afspraak w acti... |
| **1** | 219.0 | False | spronggewricht echter geen artritis objectivee... |
| **2** | 230.0 | False | in verleden minimaal calciumintaak geen osteop... |
| **3** | 58.0 | True | van depomedrol ct bdz bij verdenking quervain ... |
| **4** | 288.0 | True | reeds gestarren suppletie via huisarts doormaa... |

### *Ορίστε τις σχετικές στήλες:*

- • X στήλη = στήλη κειμένου
- • y στήλη = ετικέτα για πρόβλεψη (αν υπάρχει) -> διαφορετικά μπορείτε να παρέχετε μια κενή συμβολοσειρά
- id column = patient id

In [40]:

```
id_column='PATNR'
X_column='text'
y_column="annotation"
```

### *Προετοιμάστε δεδομένα δοκιμής*

In [42]:

```
print('[BEFORE] nr of entries:', len(radiag_df), '\tnr of patients:',
len(radiag_df['PATNR'].unique()))
valid_df = func.mergeOnColumn(valid_df, id_column, X_column, y_column)
print('[AFTER] nr of entries:', len(radiag_df), '\tnr of patients:',
len(radiag_df['PATNR'].unique()))

valid_df[X_column] = valid_df[X_column].apply(lambda x :
func.processArtefactsXML(str(x)))
valid_df[X_column] = valid_df[X_column].apply(lambda x :
func.simpleCleaning(x, stem=False))
valid_df[X_column] = valid_df[X_column].apply(lambda x :
func.stemmingText(x, stemmer=stemmer))
valid_df = valid_df.fillna('')
valid_df.to_csv(r'validation/prepped_gold_df.csv', sep='|', index=False)
[BEFORE] nr of entries: 668    nr of patients: 668
[AFTER] nr of entries: 668    nr of patients: 668
```

### *7.2 Πραγματοποίηση προβλέψεων*

In [44]:

```
import pickle

# get EMR text
valid_X = valid_df[X_column].values

# apply built model on provided text
probas_ = loaded_model.predict_proba(valid_X)
pred = probas_[:,1]

# add predictions to table & save predictions
valid_df['prediction'] = valid_df['PATNR'].copy()
valid_df['prediction'] = pred
valid_df.to_csv(r'validation/gold_predictions.csv', sep='|', index=False)

valid_df.head()
```

Out[44]:

| | text | PATNR | annotation | prediction |
|---|---|---|---|---|

| | text | PATNR | annotation | prediction |
|---|---|---|---|---|
| **0** | w folaatzur mg w prednison mg afsprak w acti... | 51.0 | True | 0.870694 |
| **1** | spronggewricht echter gen artritis objective... | 219.0 | False | 0.040848 |
| **2** | in verled minimal calciumintak gen osteoporo... | 230.0 | False | 0.080816 |
| **3** | van depomedrol ct bdz bij verdenk quervain t... | 58.0 | True | 0.953309 |
| **4** | red gestarr suppletie via huisart doormak ne... | 288.0 | True | 0.030657 |

### *7.3 Αξιολογήστε το μοντέλο*

Μπορείτε να αξιολογήσετε το μοντέλο εάν έχετε σχολιάσει το σύνολο δοκιμής.

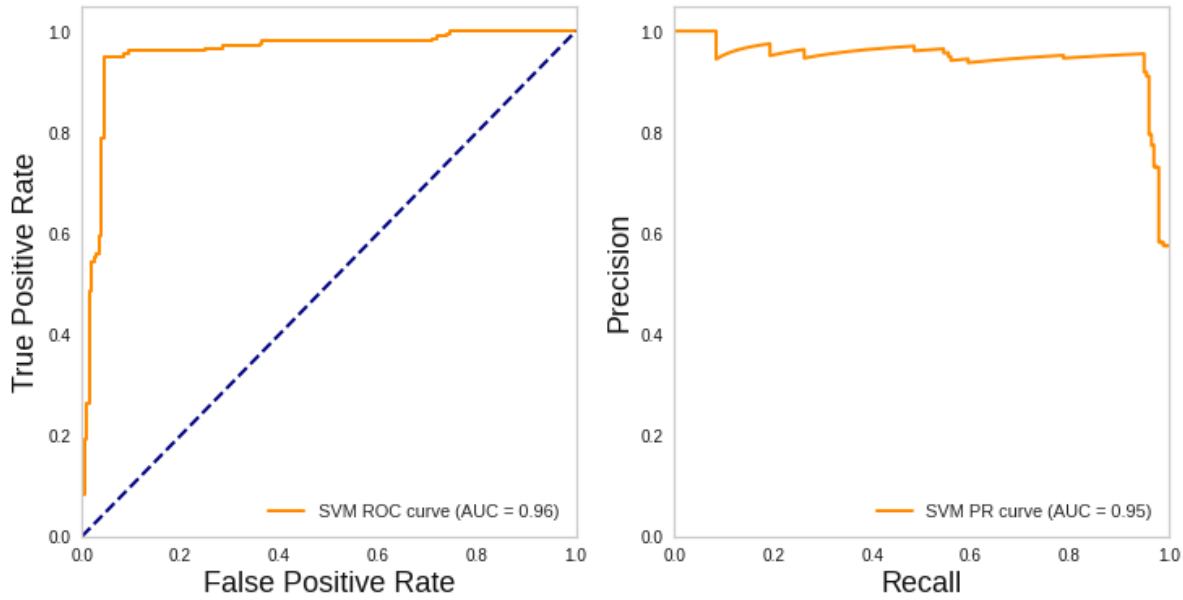
## Σχεδιάστε την καμπύλη ROC

Αξιολογήστε την απόδοση του μοντέλου στο σύνολο χρυσού προτύπου EHR συγκρίνοντας τις προβλέψεις του μοντέλου με τον σχολιασμό (y_column)

In [45]:

```
from sklearn.metrics import roc_curve, auc, precision_recall_curve

fpr, tpr, _ = roc_curve(valid_df[y_column], valid_df['prediction'])
prec, recall, _ = precision_recall_curve(valid_df[y_column],
valid_df['prediction'])

pr_auc = auc(recall, prec)
roc_auc = auc(fpr, tpr)

fig1, ax1 = plt.subplots(1,2,figsize=(12,6))

clf_name = re.match(r'.*/(.*).sav', model_name).group(1)

lw = 2
## ROC
ax1[0].plot(fpr, tpr, color='darkorange',
         lw=lw, label='%s ROC curve (AUC = %0.2f)' % (clf_name, roc_auc))
ax1[0].plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
ax1[0].set_xlim([0.0, 1.0])
ax1[0].set_ylim([0.0, 1.05])
ax1[0].set_xlabel('False Positive Rate', fontsize=18)
ax1[0].set_ylabel('True Positive Rate', fontsize=18)
ax1[0].legend(loc="lower right", fontsize=11)
ax1[0].tick_params(axis='both', which='major', labelsize=10)
ax1[0].tick_params(axis='both', which='minor', labelsize=8)

## PR
```

```
ax1[1].plot(recall, prec, color='darkorange',
         lw=lw, label='%s PR curve (AUC = %0.2f)' % (clf_name, pr_auc))
ax1[1].set_xlim([0.0, 1.0])
ax1[1].set_ylim([0.0, 1.05])
ax1[1].set_xlabel('Recall', fontsize=18)
ax1[1].set_ylabel('Precision', fontsize=18)
ax1[1].legend(loc="lower right", fontsize=11)
ax1[1].tick_params(axis='both', which='major', labelsize=10)
ax1[1].tick_params(axis='both', which='minor', labelsize=8)

plt.show()
```




### *7.4 Εφαρμογή αποκοπής*

Οραματιστείτε την αντίστοιχη μήτρα σύγχυσης και αξιολογήστε εάν ο ταξινομητής είναι σε θέση να διατηρήσει παρόμοια υψηλή ακρίβεια/ευαισθησία στο σύνολο δοκιμής.

In [ ]:

```
import importlib as imp
cut_off = 0.68 # defined on the trainingsset

tm.classificationReport(valid_df[y_column], valid_df['prediction'],
threshold=0.68)
```

### *Extra: (Βήμα 8) Οπτικοποιήστε τα πιο σημαντικά χαρακτηριστικά*

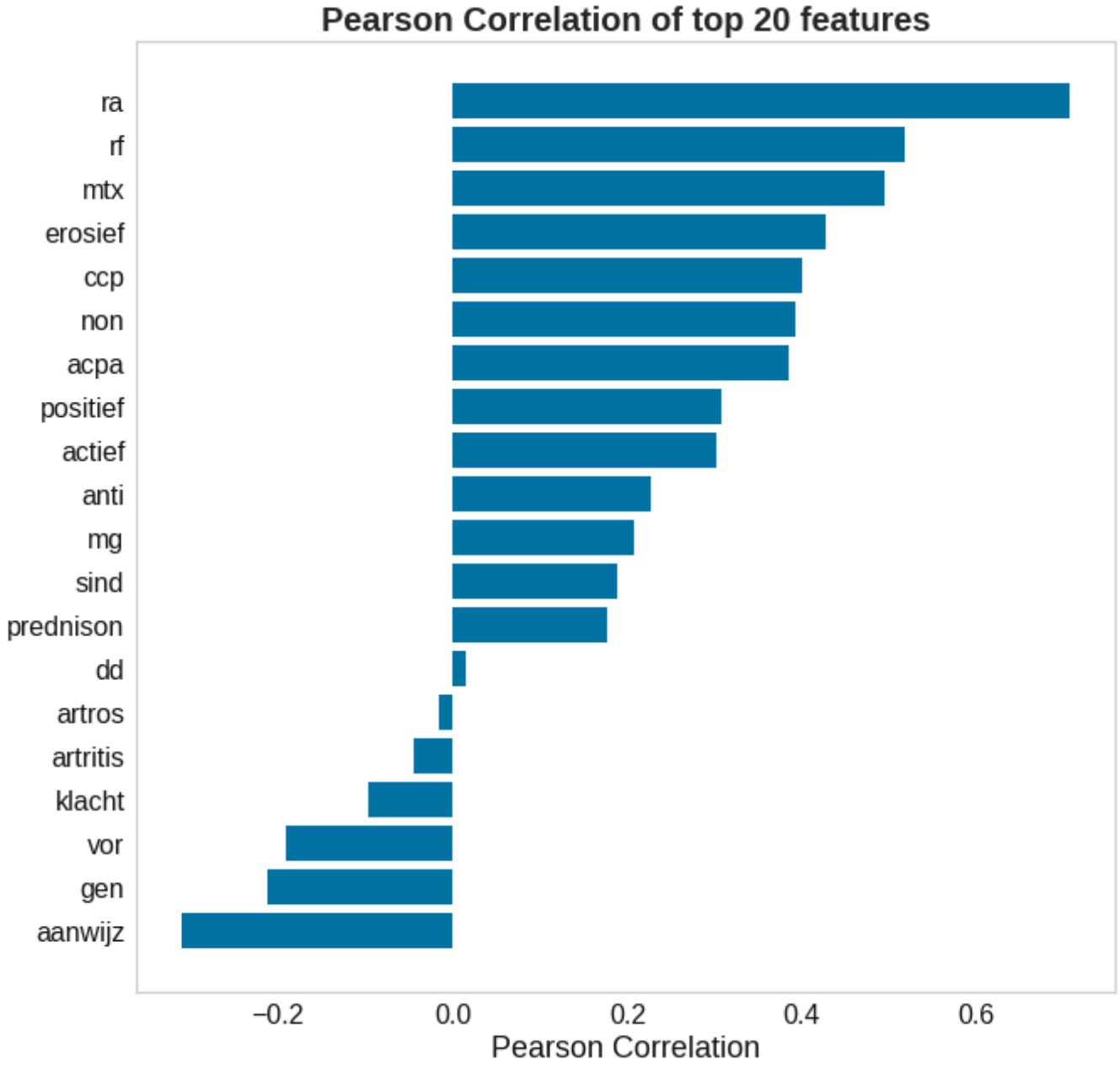
Προαιρετικές οπτικοποιήσεις

Λάβετε υπόψη: η σημασία του χαρακτηριστικού δεν λειτουργεί αυτήν τη στιγμή -> επομένως τα παρακάτω παραδείγματα είναι γενικές απεικονίσεις

### *8.1 Συσχέτιση Pearson ανά χαρακτηριστικό*

In [47]:

```
clf, pred, x_train, y_train = tm.getTrainedClassifier('SVM', clf=False)
plt = func.plotFeatureCorrelation(x_train, y_train, nr_features=20,
ngram_range=(1,3))
locs, labels = plt.yticks()

plt.show()
/home/tdmaarseveen/.conda/envs/ra_clustering2/lib/python3.8/site-
packages/sklearn/utils/deprecation.py:87: FutureWarning: Function
get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and
will be removed in 1.2. Please use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```




### *8.2 Είναι το σετ εκπαίδευσης αντιπροσωπευτικό για το σετ δοκιμής*

Θα πρέπει να είναι συγκρίσιμο -> διαφορετικά δεν έχετε αντιπροσωπευτικό train/test set

### *Input:*

- nr_features = αριθμός χαρακτηριστικών προς σχεδίαση στο διάγραμμα διανομής

### *Disclaimer*

Σε αυτήν την περίπτωση και οι δύο διανομές είναι σχεδόν ακριβώς ίδιες: αυτό δεν συμβαίνει σχεδόν ποτέ με πραγματικά δεδομένα (χρησιμοποιήσαμε εικονικά δεδομένα). Θα πρέπει να είστε δύσπιστοι εάν τα σύνολα δεδομένων ευθυγραμμίζονται τέλεια.

In [48]:

```
import matplotlib.pyplot as plt

# remove stop words in test set
radiag_df['Text'] = radiag_df['Text'].apply(lambda x: ' '.join([item for
item in x.split(' ') if item not in stop_words]))
valid_X = radiag_df['Text']

# Assess most prevalent features
func.plotTrainTestDistribution(X, valid_X, nr_features=50)
```