



**ΕΦΑΡΜΟΓΕΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ**

**ΜΑΛΛΗΣ ΒΑΣΙΛΕΙΟΣ 44513**

**ΠΑΠΑΘΑΝΑΣΗΣ ΘΩΜΑΣ 44703**

**ΌΝΟΜΑ ΕΠΙΒΛΕΠΟΝΤΑ ΚΑΘΗΓΗΤΗ**

**ΓΡΗΓΟΡΙΟΣ ΝΙΚΟΛΑΟΥ**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ**

**ΤΜΗΜΑ ΑΥΤΟΜΑΤΙΣΜΟΥ**

**ΣΕΠΤΕΜΒΡΙΟΣ 2019**

## Περιεχόμενα

Κατάλογος εικόνων .....	4
Περίληψη .....	10
Abstract.....	11
Ευχαριστίες.....	12
Κεφάλαιο 1: .....	13
Εισαγωγή .....	13
Τεχνητή νοημοσύνη (Α.Ι.).....	14
Έμπειρα Συστήματα (Expert Systems).....	17
Οι «Έξυπνες» Ιατρικές Συσκευές.....	18
Τα Συστήματα Υποστήριξης Αποφάσεων.....	19
Τα Συστήματα Μάθησης .....	19
Τεχνητή νοημοσύνη (Α.Ι.) και Ιατρική .....	20
Ορισμοί.....	22
Θεωρία αποφάσεων .....	22
Ιστορική Αναδρομή .....	24
Κεφάλαιο 2: .....	35
Εισαγωγή .....	35
Μηχανική μάθηση .....	35
Ιστορία και η σύνδεση με άλλους τομείς: .....	36
Διαδικασίες μάθησης:.....	43
Μάθηση με Εκπαιδευτή.....	43
Μάθηση χωρίς Εκπαιδευτή .....	44
Ενισχυτική Μάθηση: .....	44
Μη Επιβλεπόμενη Μάθηση.....	45
Αλγόριθμοι Μηχανικής Μάθησης.....	47
Παράμετροι συντονισμού: .....	49
Συναρτήσεις ενεργοποίησης:.....	53
Βηματική συνάρτηση ενεργοποίησης: .....	53
Μη γραμμική συνάρτηση ενεργοποίησης: .....	53
Κεφάλαιο 3 .....	72
Εισαγωγή .....	72

Τι είναι το Matlab;.....	72
Πλεονεκτήματα του MATLAB .....	73
Μειονεκτήματα του MATLAB .....	73
Εφαρμογές: .....	74
Εργαλειοθήκες – Toolboxes .....	74
Αριθμητική Ταξινόμηση (Numerical Classification) .....	75
Breast Cancer Wisconsin.....	75
Fisher Iris .....	85
Ταξινόμηση εικόνων (Image Classification).....	88
Το πρόβλημα της ταξινόμησης εικόνων.....	89
Σύγκριση και ταξινόμηση.....	90
Αναπαράσταση του Bag of Features: .....	91
Παράδειγμα Dog & cat.....	93
Γραφικό περιβάλλον Classification Learner .....	98
Αποτελέσματα .....	103
Συμπεράσματα.....	104
Βιβλιογραφία .....	105

## Κατάλογος εικόνων

Εικόνα 1.1
Εικόνα 1.2 Ο μυθικός Τάλως αποτελεί την πρώτη μορφή ρομπότ στην ιστορία της Ανθρωπότητας (ασημένιο δίδραχμο του 3ου αι. π.Χ.).
Εικόνα 1.3 Η λέξη “ρομπότ” πρωτοχρησιμοποιείται στο θεατρικό έργο R.U.R. του Τσέχου συγγραφέα Κάρελ Τσάπεκ
Εικόνα 1.4 Τζορτζ Μπουλ (1815-1864)
Εικόνα 1.5 I Robot
Εικόνα 1.6 Τζον Μακάρθι (1927-2011), ο “πνευματικός πατέρας” της τεχνητής νοημοσύνης.
Εικόνα 1.7 Shakey
Εικόνα 1.8 TI Explorer
Εικόνα 1.9 Furby
Εικόνα 1.10 AIBO
Εικόνα 1.11 VW Tuareg R5 (Stanford Racing Team)
Εικόνα 2.1
Εικόνα 2.2 Stellario Cama
Εικόνα 2.3 Computer Laboratory, University of Cambridge
Εικόνα 2.4 Alan Turing statue at Bletchley Park
Εικόνα 2.5 Marvin Minsky at MIT in 1968
<a href="https://ichef.bbci.co.uk/images/ic/496xn/p067jpdr.jpg">https://ichef.bbci.co.uk/images/ic/496xn/p067jpdr.jpg</a>
Εικόνα 2.6 Sir James Lighthill, author of 'Artificial Intelligence: A General Survey'
Εικόνα 2.7 Garry Kasparov lost his second match to a computer

Εικόνα 2.8 Neural network image recognition via backpropagation
Εικόνα 2.9 DeepMind artificial intelligence moving an animated figure
Εικόνα 2.10 Go was considered a difficult game
Εικόνα 2.11 BBC Click's Spencer Kelly and a humanoid robot
Εικόνα 2.12 Learning processes
Εικόνα 2.14 Διαχωρισμός σημείων ενδιαφέροντος
Εικόνα 2.15: Αριστερά: Χαμηλή τιμή κανονικοποίησης/ Δεξιά: Υψηλή τιμή κανονικοποίησης
Εικόνα 2.16 Υψηλό Gamma
Εικόνα 2.17 Κακό περιθώριο (πολύ κοντά στη μια περιοχή) - Καλό περιθώριο (ίσες αποστάσεις)
Εικόνα 2.18 Οπίσθια πιθανότητα
Εικόνα 2.19 Παράδειγμα πιθανότητας καιρού
Εικόνα 2.20 Αποτέλεσμα πιθανότητας
Εικόνα 2.21 Πίνακες συχνότητας - πιθανότητας
Εικόνα 2.22 Παράμετροι της πιθανότητας
Εικόνα 2.23 Προβλέψεις του καιρού μέσω δέντρου απόφασης
Εικόνα 2.24 Εντροπία
Εικόνα 2.25 Εντροπία 2
Εικόνα 2.26 Εντροπία 3
Εικόνα 2.27 Περιπτώσεις κέρδους
Εικόνα 2.28 Υπολογισμός κέρδους
Εικόνα 2.29 Μεγαλύτερο κέρδος πληροφοριών
Εικόνα 2.30 Γράφημα αποτελέσματος
Εικόνα 2.31 Γράφημα αποτελέσματος 2

Εικόνα 2.32 Δέντρο αποφάσεων
Εικόνα 2.33 Αλγόριθμος πλησιέστερων γειτόνων
Εικόνα 2.34 Δέντρο αποφάσεων
Εικόνα 2.35 Απόσταση Hamming
Εικόνα 2.36 Γράφημα πρόβλεψης δανείου
Εικόνα 2.37 Euclidean απόσταση
Εικόνα 2.38 Τυποποιημένη μεταβλητή
Εικόνα 2.39 Γραμμική παλινδρόμηση (Linear Regression)
Εικόνα 3.1 Βιβλιοθήκες του Matlab
Εικόνα 3.2 Χαρακτηριστικές τιμές γνωρισμάτων του καρκίνου
Εικόνα 3.3 Πίνακας εισαγωγής των τιμών
Εικόνα 3.4 Διαχωρισμός σε στήλες και εισαγωγή σε διάταξη πίνακα
Εικόνα 3.5 Καρτέλα ταξινομητών
Εικόνα 3.6 Predictions model
Εικόνα 3.7 Predictions model
Εικόνα 3.8 Predictions model
Εικόνα 3.9 Confusion matrix
Εικόνα 3.10 ROC curve
Εικόνα 3.11 Parallel Coordinates Plot
Εικόνα 3.12 Αποτελέσματα πρόβλεψης
Εικόνα 3.13 Predictions Model
Εικόνα 3.14 Confusion matrix
Εικόνα 3.15 ROC Curve
Εικόνα 3.16 ROC Curve

Εικόνα 3.17 ROC Curves
Εικόνα 3.18 Parallel Coordinates Plot
Εικόνα 3.19 Πίνακας Dataset
Εικόνα 3.20 Scatter plot
Εικόνα 3.21 Confusion Matrix
Εικόνα 3.22 ROC curve 72,4%
Εικόνα 3.23 Parallel Coordinates plot
Εικόνα 3.24 Ροή εκτέλεσης εντολών 1
Εικόνα 3.25 Ροή εκτέλεσης εντολών 2
Εικόνα 3.26 Ροή εκτέλεσης εντολών 3

**ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

Ο/Η κάτωθι υπογεγραμμένος/η Μάκης Βασιλείος, του Νικολαίου, φοιτητής του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής του Πανεπιστημίου Δυτικής Αττικής, πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε, ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα, σε περίπτωση που το ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασή της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση Π.Ε με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού βμήνου από την ημερομηνία ανάθεσής της.

Ο Δηλών



Ημερομηνία

10/09/2019



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ (1<sup>η</sup> ή 2<sup>η</sup> σελίδα)

**ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ**

Ο/Η κάτωθι υπογεγραμμένος/η Θαράς - Θεοφάνης Παπαθανάσης, του Θεοφάνη, φοιτητής του Τμήματος Μηχανικών Βιομηχανικής Σχεδίασης και Παραγωγής του Πανεπιστημίου Δυτικής Αττικής, πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε, ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα, σε περίπτωση που το Ίδρυμα του έχει απονεμίσει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφασή της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση Π.Ε με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού βμήνου από την ημερομηνία ανάθεσής της.

Ο Δηλών



Ημερομηνία

10/09/2019

## Περίληψη

Ο σκοπός της παρούσας πτυχιακής εργασίας είναι να παρουσιάσει το θέμα των εφαρμογών Τεχνητής Νοημοσύνης και Μηχανικής Μάθησης σε απλές μορφές.

Πρώτα απ' όλα, ξεκινάει με μία επεξήγηση των όρων της Τεχνητής Νοημοσύνης και της Μηχανικής Μάθησης και κατόπιν θα γίνει μία σύντομη ιστορική αναδρομή των επιστημών από την αρχαιότητα μέχρι σήμερα καθώς και την ραγδαία ανάπτυξη της σε πολλούς τομείς.

Κατόπιν, παρουσιάζουμε την λειτουργία τους μέσω δύο ειδών ταξινόμησης, την αριθμητική και ταξινόμηση γνωρισμάτων εικόνων και αναγνώριση αυτών. Για την επεξεργασία των εικόνων και την εξόρυξη των γνωρισμάτων χρησιμοποιήθηκε το πρόγραμμα Matlab.

Τέλος, γίνεται αξιολόγηση των αποτελεσμάτων που είχαμε ως τιμές καθώς και της απόδοσης του.

## **Abstract**

The purpose of this diploma thesis is to present the subject of applications of Artificial Intelligence and Machine Learning in simple forms.

First of all, it starts with an explanation of the terms of Artificial Intelligence and Machine Learning, then there will be a brief historical overview of the sciences behind Artificial Intelligence, from antiquity to the present and its rapid development in many areas.

Then, present their function with two types of sorting, numeric and classification of the characteristic images and their identification. The Matlab program was used for image editing and feature extraction.

Finally, an evaluation of the results as well as its performance is made.

## Ευχαριστίες

Θεωρούμε υποχρέωσή μας να ευχαριστήσουμε τον επιβλέποντα καθηγητή κύριο Νικολάου Γρηγόριο κυρίως για την πολύτιμη καθοδήγηση και βοήθειά του για την επίλυση διάφορων θεμάτων που πρόέκυψαν κατά τη διάρκεια υλοποίησης της πτυχιακής εργασίας. Θα θέλαμε επίσης να απευθύνουμε τις ευχαριστίες μας στους γονείς μας, καθώς στήριξαν τις σπουδές μας με ποικίλους τρόπους, φροντίζοντας έτσι την καλύτερη δυνατή μόρφωση μας.

## Κεφάλαιο 1:

### Εισαγωγή

Βασικός στόχος μας ως ανθρώπινο είδος από την αρχαιότητα έως και σήμερα είναι η προσπάθεια μας να διαμορφώσουμε τις κατάλληλες συνθήκες, οι οποίες θα μας εξασφαλίσουν μια καλή ποιότητα ζωής. Για την επίτευξη αυτού του σκοπού το ανθρώπινο είδος στράφηκε προς την επιστήμη και την τεχνολογία ως άξια μέσα επίτευξης αυτού του στόχου.

Απο τα βάθη της αρχαιότητας λοιπόν έως και σήμερα και το πιο απλό καθημερινό εργαλείο μέχρι και οι πιο περιπλοκές συσκευές η εφευρέσεις είναι ένα μέρος της ιστορίας της ανθρώπινης εξέλιξης. Κάθε επιθυμία μας για την ανακάλυψη ενός νέου και πιο εξελιγμένου εξοπλισμού αντικατοπτρίζει τις ανάγκες της εκάστοτε εποχής. Για παράδειγμα, δημιουργήσαμε εργαλεία για να πολλαπλασιάσουμε την φυσική μας δύναμη, είτε ενισχύοντας την με αυτή ενός ζώου είτε αντικαθιστώντας την με μια εφεύρεση οπού θα μας βοηθούσε πολύ για αρκετά χρονιά. Μια τέτοια εφεύρεση αποτέλεσε και η ανακάλυψη του τροχού ως μέσω κίνησης. Έτσι λοιπόν ο άνθρωπος έλυσε ποικίλα θέματα εκείνης της εποχής που τον κρατούσαν "πίσω" από τους στόχους του, οι οποίοι δεν είναι άλλοι από την αναβάθμιση του βιοτικού του επιπέδου ζωής.

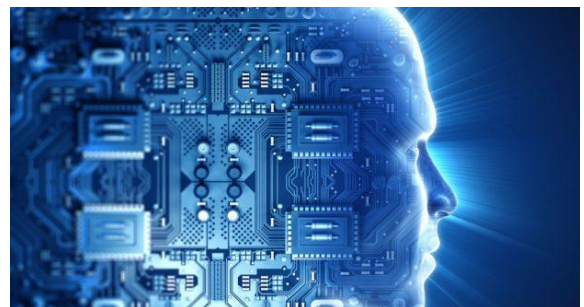
Από μικρή ηλικία μάθαμε πως το είδος μας ονομάζεται homo sapiens (άνθρωπος ο σοφός), καθώς οι νοητικές μας ικανότητες είναι πολύ σημαντικές για το είδος μας διότι αποτελούν τον καθοριστικό παράγοντα της εξέλιξης. Χιλιάδες χρόνια τώρα προσπαθούμε να κατανοήσουμε το τρόπο σκέψης μας καθώς αποτελεί μεγάλο γεγονός ότι μια χούφτα ύλη μπορεί να αντιληφθεί, να κατανοήσει να προβλέψει και να χειριστεί ένα πελώριο και πολύπλοκο κόσμο.

Στο σημείο αυτό έρχεται η επιστήμη της τεχνητής νοημοσύνης (artificial intelligence), η οποία επιχειρεί όχι μόνο να κατανοήσει άλλα και να κατασκευάσει νοήμονες οντότητες. Επιπλέον αποτελεί μια από τις νεότερες επιστήμες σήμερα καθώς συνδυάζει μια τεράστια ποικιλία επιμέρους πεδίων, τα οποία καλύπτουν ένα φάσμα που ξεκινά από γενικούς τομείς, όπως είναι η μάθηση και η αντίληψη και φτάνει έως και τη διάγνωση ασθενειών, το οποίο αποτελεί και το αντικείμενο της παρούσας πτυχιακής. Τέλος, μπορεί να αυτοματοποιήσει την ανθρώπινη αντίληψη και για τον λόγο αυτό την συναντάμε όλο και πιο συχνά στην καθημερινότητά μας.

Σήμερα υπάρχουν πολλοί που ισχυρίζονται πως η τεχνητή νοημοσύνη (A.I.) ήρθε για να αντικαταστήσει τον άνθρωπο, απ την άλλη μεριά υπάρχουν εκείνοι οι οποίοι πιστεύουν πως δημιουργήθηκε για την εξυπηρέτηση του ανθρώπου και χρίζει συζήτησης περί αντικατάστασης του ανθρώπου. Στην συνέχεια θα αναλύσουμε τί ακριβώς είναι και πού φέρει χρήση ο τομέας της τεχνητής νοημοσύνης (A.I.) και να ξεκαθαρίσουμε τι ακριβώς ισχύει.

## Τεχνητή νοημοσύνη (A.I.)

Ο όρος τεχνητή νοημοσύνη (A.I.) αναφέρεται στον κλάδο της πληροφορικής ο οποίος ασχολείται με τη σχεδίαση και την υλοποίηση υπολογιστικών συστημάτων που μιμούνται στοιχεία της ανθρώπινης συμπεριφοράς τα οποία υπονοούν έστω και στοιχειώδη ευφυΐα: μάθηση, προσαρμοστικότητα, εξαγωγή συμπερασμάτων, κατανόηση από συμφραζόμενα, επίλυση προβλημάτων κλπ. Ο Τζον Μακάρθι όρισε τον τομέα αυτόν ως «επιστήμη και μεθοδολογία της δημιουργίας νοούντων μηχανών».



Εικόνα 2.1 <https://www.pestola.gr/wp-content/uploads/project-brainwave-microsoft-600x400.jpg>

Η τεχνητή νοημοσύνη (A.I.) αποτελεί σημείο τομής μεταξύ πολλαπλών επιστημών όπως της πληροφορικής, της ψυχολογίας, της φιλοσοφίας, της νευρολογίας, της γλωσσολογίας και της επιστήμης μηχανικών, με στόχο τη σύνθεση ευφυούς συμπεριφοράς, με στοιχεία συλλογιστικής, μάθησης και προσαρμογής στο περιβάλλον, ενώ συνήθως εφαρμόζεται σε μηχανές ή υπολογιστές ειδικής κατασκευής.

Διαιρείται στη συμβολική τεχνητή νοημοσύνη, η οποία επιχειρεί να εξομοιώσει την ανθρώπινη νοημοσύνη αλγοριθμικά χρησιμοποιώντας σύμβολα και λογικούς κανόνες υψηλού επιπέδου, και στην υποσυμβολική τεχνητή νοημοσύνη, η οποία προσπαθεί να αναπαράγει την ανθρώπινη ευφυΐα χρησιμοποιώντας στοιχειώδη αριθμητικά μοντέλα που συνθέτουν επαγωγικά νοήμονες συμπεριφορές με τη διαδοχική αυτοοργάνωση απλούστερων δομικών συστατικών («συμπεριφορική τεχνητή νοημοσύνη»), προσομοιώνουν πραγματικές βιολογικές διαδικασίες όπως η εξέλιξη των ειδών και η λειτουργία του εγκεφάλου

(«υπολογιστική νοημοσύνη»), ή αποτελούν εφαρμογή στατιστικών μεθοδολογιών σε προβλήματα τεχνητής νοημοσύνης.

Η διάκριση σε συμβολικές και υποσυμβολικές προσεγγίσεις αφορά τον χαρακτήρα των χρησιμοποιούμενων εργαλείων, ενώ δεν είναι σπάνια η σύζευξη πολλαπλών προσεγγίσεων (διαφορετικών συμβολικών, υποσυμβολικών, ή ακόμα συμβολικών και υποσυμβολικών 2 μεθόδων) κατά την προσπάθεια αντιμετώπισης ενός προβλήματος. Με βάση τον επιθυμητό επιστημονικό στόχο η Τεχνητή νοημοσύνη (A.I.) κατηγοριοποιείται σε άλλου τύπου ευρείς τομείς, όπως επίλυση προβλημάτων, μηχανική μάθηση, ανακάλυψη γνώσης, συστήματα γνώσης κλπ. Επίσης υπάρχει επικάλυψη με συναφή επιστημονικά πεδία όπως η μηχανική όραση, η επεξεργασία φυσικής γλώσσας ή η ρομποτική, τα οποία μπορούν να τοποθετηθούν μες στο ευρύτερο πλαίσιο της σύγχρονης τεχνητής νοημοσύνης ως ανεξάρτητα πεδία της.

Η λογοτεχνία και οι ταινίες επιστημονικής φαντασίας από τη δεκαετία του 1920 μέχρι σήμερα έχουν δώσει στο ευρύ κοινό την αίσθηση ότι η TN αφορά την προσπάθεια κατασκευής μηχανικών ανδροειδών ή αυτοσυνείδητων προγραμμάτων υπολογιστή (*ισχυρή TN*), επηρεάζοντας μάλιστα ακόμα και τους πρώτους ερευνητές του τομέα. Στην πραγματικότητα οι περισσότεροι επιστήμονες της τεχνητής νοημοσύνης προσπαθούν να κατασκευάσουν λογισμικό ή πλήρεις μηχανές οι οποίες να επιλύουν με αποδεκτά αποτελέσματα ρεαλιστικά υπολογιστικά προβλήματα οποιουδήποτε τύπου (*ασθενής TN*), αν και πολλοί πιστεύουν ότι η εξομοίωση ή η προσομοίωση της πραγματικής ευφυΐας, η *ισχυρή TN*, πρέπει να είναι ο τελικός στόχος.

Η σύγχρονη τεχνητή νοημοσύνη (A.I.) αποτελεί ένα από τα πλέον «μαθηματικοποιημένα» και ραγδαίως εξελισσόμενα πεδία της πληροφορικής. Σήμερα, ο τομέας αξιοποιεί περισσότερο υποσυμβολικές μεθόδους και εργαλεία καταγόμενα από τα εφαρμοσμένα μαθηματικά και τις επιστήμες μηχανικών, παρά από τη θεωρητική πληροφορική και τη μαθηματική λογική όπως συνέβαινε πριν το 1990. Σε ακαδημαϊκό επίπεδο η τεχνητή νοημοσύνη (A.I.) μελετάται επίσης από την ηλεκτρονική μηχανική, ενώ συνιστά ένα από τα σημαντικότερα θεμελιακά συστατικά του διεπιστημονικού γνωστικού πεδίου της επιστήμης.

Εκτενέστερα, ένα τυπικό AI αντιλαμβάνεται το περιβάλλον του και αναλαμβάνει δράσεις που μεγιστοποιούν την πιθανότητα επιτυχίας των στόχων του. Η σκοπούμενη λειτουργία στόχου ενός AI μπορεί να είναι απλή ("1 εάν το AI κερδίσει ένα παιχνίδι Go, 0 διαφορετικά") ή σύνθετο ("Κάνετε ενέργειες μαθηματικά παρόμοιες με τις ενέργειες που σας έδωσαν ανταμοιβές στο παρελθόν").

Οι στόχοι μπορούν να οριστούν σαφώς ή να προκληθούν. Εάν η TN είναι προγραμματισμένο για την "ενίσχυση της μάθησης", οι στόχοι μπορούν να προκληθούν σιωπηρά επιβραβεύοντας κάποιους τύπους συμπεριφοράς και τιμωρώντας άλλους. Εναλλακτικά, ένα εξελικτικό σύστημα μπορεί να προκαλέσει στόχους χρησιμοποιώντας μια "λειτουργία προπόνησης" για να μεταλλαχθεί και να αναπαραγάγει κατά προτίμηση - παρακολούθηση συστημάτων TN. Αυτό είναι παρόμοιο με το πώς τα ζώα εξελίχθηκαν για να επινοήσουν γενικά ορισμένους στόχους όπως η εξεύρεση τροφής ή πώς μπορούν να εκτραφούν τα σκυλιά μέσω τεχνητής επιλογής για να έχουν τα επιθυμητά χαρακτηριστικά.

Ορισμένα συστήματα TN, όπως ο πλησιέστερος γείτονας (nearest-neighbor), αντί να λογίζονται κατ'αναλογία, αυτά τα συστήματα δεν έχουν γενικά στόχους, εκτός από το βαθμό που οι στόχοι είναι κάπως σιωπηλοί στα εκπαιδευτικά τους δεδομένα. Τέτοια συστήματα μπορούν ακόμα να συγκρίνουν, εάν το σύστημα μη στόχου είναι πλαισιωμένο ως ένα σύστημα του οποίου ο «στόχος» είναι να ολοκληρώσει επιτυχώς την λειτουργία ταξινόμησης του.

Η TN συχνά περιστρέφεται γύρω από τη χρήση αλγορίθμων. Ένας αλγόριθμος είναι ένα σύνολο από σαφείς οδηγίες που μπορεί να εκτελέσει ένας υπολογιστής. Ένας πολύπλοκος αλγόριθμος είναι συχνά χτισμένος πάνω από άλλους, απλούστερους, αλγόριθμους.

Πολλοί αλγόριθμοι TN είναι ικανοί να μάθουν από τα δεδομένα, μπορούν να αναβαθμιστούν με την εκμάθηση νέων ευρετικών (στρατηγικές ή «κακοήθεις κανόνες» που έχουν δουλέψει καλά στο παρελθόν) ή μπορούν να γράψουν και άλλους αλγορίθμους. Κάποιοι από τους "μαθητές" που περιγράφονται παρακάτω, συμπεριλαμβανομένων των Bayesian δικτύων, των δέντρων αποφάσεων και του πλησιέστερου γείτονα (nearest-neighbor), θα μπορούσαν θεωρητικά, εάν δοθούν άπειρα δεδομένα, χρόνος και μνήμη, να μάθουν να προσεγγίζουν οποιαδήποτε λειτουργία, συμπεριλαμβανομένου οποιουδήποτε συνδυασμού μαθηματικών λειτουργιών.

Θα μπορούσαν επομένως, θεωρητικά, να αντλήσουν όλες τις πιθανές γνώσεις, εξετάζοντας κάθε πιθανή υπόθεση και ταιριάζοντάς την με τα δεδομένα. Στην πράξη, σχεδόν ποτέ δεν είναι δυνατόν να εξεταστεί κάθε πιθανότητα, λόγω του χρόνου που απαιτείται για την επίλυση ενός προβλήματος (αυξάνεται εκθετικά).

Μεγάλο μέρος της έρευνας TN περιλαμβάνει τη διαπίστωση του πώς μπορεί να εντοπιστεί και να αποφευχθεί η εξέταση ευρέων ορίων δυνατοτήτων που είναι απίθανο να γίνουν



καρποφόρες. Για παράδειγμα, κατά την προβολή ενός χάρτη και την αναζήτηση της συντομότερης οδικής διαδρομής από το Ντένβερ προς τη Νέα Υόρκη στην Ανατολή, στις περισσότερες περιπτώσεις μπορεί κανείς να παραλείπει να ψάχνει σε οποιοδήποτε μονοπάτι μέσα από το Σαν Φρανσίσκο ή σε άλλες περιοχές μακριά προς τη Δύση. Έτσι, ένα σύστημα TN που χρησιμοποιεί έναν αλγόριθμο pathfinding μπορεί να αποφύγει τη χρονική σπατάλη που θα προέκυπτε αν έπρεπε να εξεταστεί κάθε δυνατή διαδρομή.

Η παλαιότερη (και ευκολότερη κατανόηση) προσέγγιση του συστήματος TN ήταν ο συμβολισμός: "Εάν ένας κατά τα άλλα υγιής ενήλικας έχει πυρετό, τότε μπορεί να έχει γρίπη". Μια δεύτερη, πιο γενική προσέγγιση είναι Bayesian συμπεράσματα: "Εάν ο τρέχων ασθενής έχει πυρετό, προσαρμόστε την πιθανότητα να έχει τη γρίπη έτσι και έτσι". Η τρίτη μείζων προσέγγιση, εξαιρετικά δημοφιλής στις συνήθεις επιχειρηματικές εφαρμογές TN, είναι αναλυτές όπως ο SVM και ο πλησιέστερος γείτονας (nearest-neighbor): "Μετά την εξέταση των αρχαίων παλαιότερων ασθενών των οποίων η θερμοκρασία, τα συμπτώματα, η ηλικία και άλλοι παράγοντες ταιριάζουν περισσότερο με τον τρέχοντα ασθενή, των ασθενών αυτών έπασχαν από γρίπη ». Μια τέταρτη προσέγγιση είναι πιο δύσκολο να κατανοηθεί διαισθητικά, αλλά είναι εμπνευσμένη από το πώς λειτουργούν τα μηχανήματα του εγκεφάλου: η προσέγγιση του τεχνητού νευρικού δικτύου χρησιμοποιεί τεχνητούς "νευρώνες" που μπορούν να μάθουν συγκρίνοντας την επιθυμητή παραγωγή και μεταβάλλοντας τις δυνάμεις των συνδέσεων μεταξύ των εσωτερικών νευρώνων για να "ενισχύσει" τις συνδέσεις που φαινόταν χρήσιμες.

Αυτές οι τέσσερις κύριες προσεγγίσεις μπορούν να αλληλεπικαλύπτονται μεταξύ τους και με εξελικτικά συστήματα. για παράδειγμα, τα νευρικά δίκτυα μπορούν να μάθουν να κάνουν συμπεράσματα, να γενικεύσουν και να κάνουν αναλογίες.

### **Έμπειρα Συστήματα (Expert Systems).**

Τα έμπειρα συστήματα αναφέρονται και σαν Συστήματα Γνώσης (Knowledge Based Systems). Μερικοί θεωρούν ότι τα έμπειρα συστήματα δεν πρέπει να κατηγοριοποιούνται στην περιοχή της TN. Στην περιοχή αυτή, εκτός των έμπειρων συστημάτων, περιλαμβάνονται και οι παρακάτω κλάδοι τεχνολογιών, όπως:

- Νευρωνικά δίκτυα (neural networks) είναι υψηλά απλοποιημένο μοντέλο του ανθρώπινου νευρικού συστήματος, το οποίο επιδεικνύει ικανότητες, όπως εκμάθηση, γενίκευση και υπεξαίρεση.

- Συστήματα Προοπτικής (Perspective systems) είναι συνηθισμένες εικόνες και ακουστικά σήματα που καθοδηγούν τους υπολογιστές με άλλες συσκευές. ^ Μάθηση (learning) εμπεριέχει όλες τις ενέργειες οι οποίες διευκολύνουν έναν υπολογιστή ή μια συσκευή να αποκτήσει επιπλέον γνώση σε σχέση με αυτή που ήδη έχει στην μνήμη του από τους κατασκευαστές ή προγραμματιστές.
- Η μάθηση μηχανής ή αλλιώς μηχανική μάθηση (machine learning) είναι ιδιαίτερα χρήσιμη για αναγνώριση σύνθετων προτύπων και σχέσεων δεδομένων.
- Ρομποτική (robotics) αποτελείται από συσκευές οι οποίες ελέγχονται από υπολογιστή οι οποίες μιμούνται τις κινητήριες ενέργειες του ανθρώπου. Η ρομποτική χρησιμοποιείται όλο και περισσότερο στη βιομηχανία, σε εργασίες ρουτίνας, μηχανικής συναρμολόγησης προϊόντων και έχει επιτύχει μεγάλη αύξηση της παραγωγικότητας με παράλληλη μείωση του κόστους.
- Υλικό της Τεχνητής Νοημοσύνης (AI Hardware) περιέχει τις φυσικές συσκευές οι οποίες στοχεύουν στην εφαρμογή της ΤΝ.
- Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing) επιτρέπει στους χρήστες να επικοινωνούν με τον υπολογιστή χρησιμοποιώντας γλώσσες όπως Ελληνικά, Ισπανικά, Γερμανικά, Αγγλικά, Γαλλικά, Κινέζικα κ.λ.π.
- Fuzzy Logic είναι μια τεχνική βασισμένη στην τεχνολογιών λογικών κανόνων. Η θεωρία αυτή αναφέρει ότι οι διαχειριστικές αποφάσεις πρέπει να βασίζονται μερικώς σε πραγματικά δεδομένα, αντιλήψεις ή καταστάσεις οι οποίες μπορεί να μην είναι πάντα σωστές ή λάθος

### Οι «Εξυπνες» Ιατρικές Συσκευές

Η επιστήμη αυτή μελετά τις ιδέες που δίνουν την δυνατότητα στα συστήματα των υπολογιστών να συμπεριφέρονται με ανθρώπινη εξυπνάδα. Η μηχανή για την ζωντανή συνομιλία, μέσω του Διαδικτύου, είναι ένα παράδειγμα έξυπνης μηχανής. Η Τεχνητή νοημοσύνη (A.I.) είναι δυνατό με κατάλληλη κλίμακα, να συγκριθεί. Έτσι μπορούμε να λέμε ότι ένας υπολογιστής με προγράμματα επεξεργασίας κειμένου και εξυπνότερος της απλής γραφομηχανής ή ότι ένα άτομο με μεγαλύτερο IQ είναι εξυπνότερο άλλου.

Από τις πολλές εφαρμογές που έχουν δημιουργηθεί και ελεγχθεί, πολύ λίγες είναι πρακτικά χρήσιμες, για να χρησιμοποιηθούν στην καθημερινή μας ζωή. Η ανάπτυξη της τεχνολογίας

επιτρέπει την ενσωμάτωση της «έξυπνάδας» σε πάρα πολλές ηλεκτρονικές συσκευές, από τα κινητά τηλέφωνα μέχρι τις έξυπνες κάρτες υγείας. Πολλές έξυπνες μη παρεμβατικές συσκευές όπως για παράδειγμα ο ψηφιακός μετρητής αρτηριακής πίεσης, οι αυτόματοι ανιχνευτές αρρυθμίας, το ηλεκτρονικό θερμόμετρο, το στήθοσκόπιο κ.α., σχεδιάστηκαν για να διευκολύνουν κυρίως τους ασθενείς που πάσχουν από χρόνια νοσήματα. Πολλά έξυπνα ιατρικά μηχανήματα όπως ο μετρητής σακχάρου, ο ηλεκτροκαρδιογράφος, για παράδειγμα χρησιμοποιούνται για την κατ' οίκον παρακολούθηση ασθενών με τέτοιο τρόπο ώστε τα αποτελέσματά τους να μεταφέρονται στον υπολογιστή του θεράποντα ιατρού τους μέσω της τηλεφωνικής γραμμής. Πολλές ακόμα έξυπνες συσκευές χρησιμοποιούνται στα νοσοκομεία, στις Μονάδες Εντατικής Θεραπείας, στα μηχανήματα που δείχνουν τις τιμές εργαστηριακών εξετάσεων και αλλού. Πολλές εφευρέσεις και μηχανήματα προβάλλονται σαν έξυπνες όπως για παράδειγμα οι έξυπνες ιατρικές μπλούζες που θα είναι ταυτόχρονα συλλέκτες, φορείς και πομποί πληροφοριών. Οι μπλούζες αυτές που έχουν αισθητήρες, καλώδια, οπτικές ίνες και μικροεπεξεργαστές, θα καταγράφουν πολλές ζωτικές ενδείξεις ενός οργανισμού, όπως τον καρδιακό του παλμό, την αρτηριακή του πίεση, τον αναπνευστικό του ρυθμό και θα τις μεταδίδουν σε κατάλληλα κέντρα αναλύσεων.

#### **Τα Συστήματα Υποστήριξης Αποφάσεων.**

Τα Συστήματα Υποστήριξης Αποφάσεων (DSS - Decision Support Systems) είναι τα προγράμματα που χρησιμοποιούν την γνώση των εμπειρογνομώνων για να δώσουν συμβουλές που θα υποστηρίξουν Διαγνωστικές και Θεραπευτικές κυρίως αποφάσεις. Ο ρόλος τους είναι χρήσιμος και στην προληπτική ιατρικά, όπως για παράδειγμα τα συστήματα υποστήριξης αποφάσεων για την ανίχνευση των πρώτων σταδίων του καρκίνου. Τα συστήματα DSS βρίσκουν ανταπόκριση στον ιατρικό κόσμο και θεωρούνται από πολλούς ιατρούς συνώνυμα της Τεχνητής Νοημοσύνης. Χρησιμοποιούνται ιδιαίτερα σε μέρη που δεν υπάρχουν ειδικοί εμπειρογνώμονες ή σε μέρη στα οποία υπάρχουν λίγοι ιατροί που αντιμετωπίζουν μόνοι τους πολλές ασθένειες και διάφορες επιδημίες.

#### **Τα Συστήματα Μάθησης**

Ο τομέας έρευνας της ΤΝ που ασχολείται κυρίως με τις τεχνικές που δίνουν την ικανότητα στον υπολογιστή να βγάζει συμπεράσματα, να αυξάνει τις γνώσεις του και μαθαίνει από τα λάθη του, είναι τα Συστήματα Μάθησης. (Learning Systems). Τα Συστήματα Μάθησης είναι προϋπόθεση για οποιαδήποτε προχωρημένη μορφή νοημοσύνης. Δεν υπάρχουν μηχανές που μπορεί να σκέπτονται, αλλά υπάρχουν μηχανές που μπορούν να μάθουν να σκέπτονται. Η βασική μέθοδος μάθησης, είναι η μέθοδος δοκιμής και λάθους (trial or error), που

επιδεικνύεται με το μηχανικό ποντίκι. Όταν ένα μηχανικό ποντίκι τοποθετηθεί στο εσωτερικό ενός λαβύρινθου που έχει διαμορφωθεί από διαδοχικούς συνδυασμούς δεξιόστροφων και αριστερόστροφων στροφών, προσπαθώντας και βρίσκοντας μετά από προσπάθειες, τον σωστό συνδυασμό εξόδου του από τον λαβύρινθο, τον απομνημονεύει και τον χρησιμοποιεί σε οποιαδήποτε μελλοντική του προσπάθεια. Το μηχανικό ποντίκι στις προσπάθειες του μπορεί να μαθαίνει από τα λάθη του, ώστε να μην κάνει ένα συγκεκριμένο λάθος περισσότερο από μία φορά. Το ποντίκι μαθαίνει αποτυπώνοντας ενέργειες που οδηγούν σε ικανοποιητικά αποτελέσματα και διαγράφοντας τις άλλες. Μπορεί ακόμα να αντιμετωπίζει νέες καταστάσεις βασιζόμενο στην προηγούμενη εμπειρία του και να γίνεται «εξυπνότερο», με όσες περισσότερες καταστάσεις αντιμετωπίζει επιτυχώς. Με παρόμοιο τρόπο οι υπολογιστές, όπως και οι άνθρωποι, αντιμετωπίζουν τις νέες καταστάσεις βασιζόμενοι στην «εμπειρία» που έχουν αποκτήσει παλαιότερα επιλύοντας πολλές ανάλογες εφαρμογές. Οι τεχνικές μάθησης χρησιμοποιούνται και στα συστήματα εμπειρογνομόνων καθώς και στα συστήματα Ασφάλειας ώστε να μπορούν να ανιχνεύουν και να αντιμετωπίζουν αυτόματα τα σχετικά προβλήματα που ανακύπτουν.

### **Τεχνητή νοημοσύνη (A.I.) και Ιατρική**

Το άτομο προσπαθεί να αυξήσει τις δυνατότητές του με το να δημιουργεί εργαλεία. Από την εφεύρεση του ροπαλού για ενισχύσει το χτύπημά του, μέχρι το ηλεκτρονικό μικροσκόπιο για να ακονίσει το όραμά του, τα εργαλεία έχουν επεκτείνει τη δυνατότητά τους μέχρι και να αισθάνονται και να χειρίζονται τον κόσμο αντί για τον άνθρωπο. Σήμερα στεκόμαστε στο κατώφλι των νέων τεχνικών εξελίξεων που θα αυξήσουν τον ανθρώπινο συλλογισμό. Ο υπολογιστής και οι μέθοδοι προγραμματισμού που επινοούνται για αυτόν τον σκοπό είναι τα νέα εργαλεία που θα πραγματοποιήσουν αυτήν την αλλαγή. Η ιατρική είναι ένας τομέας στον οποίο τέτοια βοήθεια απαιτείται αυστηρά. Οι αυξανόμενες προσδοκίες μας για την υψηλότερη ποιοτική υγειονομική περίθαλψη και η ταχεία ανάπτυξη πάντα της πιο λεπτομερούς ιατρικής γνώσης, αφήνουν τον ιατρό χωρίς επαρκή χρόνο για να αφιερώσει σε κάθε ασθενή και να αγωνίζεται να συμβαδίσει με τις νεώτερες εξελίξεις στον τομέα του. Ελλείψει του χρόνου, οι περισσότερες ιατρικές αποφάσεις πρέπει να βασιστούν σε γρήγορες κρίσεις για την περίπτωση του ασθενούς που στηρίζονται στην αβοήθητη μνήμη του ιατρού. Μόνο σε σπάνιες καταστάσεις μπορεί μια γραπτή ή άλλη εκτεταμένη έρευνα να ληφθεί υπόψη για να βεβαιώσει το γιατρό (και τον ασθενή) ότι η πιο πρόσφατη γνώση εφαρμόζεται σε οποιαδήποτε ιδιαίτερη περίπτωση.

Οι συνεχείς διαδικασίες κατάρτισης και ανανέωσης του πτυχίου ενθαρρύνουν τους ιατρούς να κρατήσουν περισσότερο τις σχετικές πληροφορίες συνεχώς στο μυαλό τους, αλλά οι θεμελιώδεις περιορισμοί της ανθρώπινης μνήμης και η επανάκληση πληροφοριών συνδεδεμένη με την αύξηση της γνώσης, βεβαιώνουν ότι το μεγαλύτερο μέρος από αυτό που είναι γνωστό δεν μπορεί να μαθευτεί από τα περισσότερα άτομα. Αυτή είναι η ευκαιρία για τα νέα υπολογιστικά εργαλεία: να βοηθήσουν να οργανωθεί, να αποθηκευτεί και να ανακτηθεί η κατάλληλη ιατρική γνώση που απαιτείται από τον επαγγελματία όσον αφορά κάθε δύσκολη περίπτωση και να προτείνουν τις κατάλληλες διαγνωστικές, προγνωστικές και θεραπευτικές τεχνικές. Σε ένα επαναληπτικό άρθρο του ο Schwartz το 1990 μιλά για τη δυνατότητα που ο υπολογιστής ως διανοητικό εργαλείο έχει να αναδιαμορφώσει το παρόν σύστημα της υγειονομικής περίθαλψης, αλλάζοντας πλήρως το ρόλο του ιατρού και αλλάζοντας βαθιά τη φύση της ιατρικής στρατολόγησης εργατικού δυναμικού και της ιατρικής εκπαίδευσης. Εν ολίγοις, αναφέρει ότι το σύστημα υγειονομικής περίθαλψης στο μέλλον θα είναι διαφορετικό από αυτό που είναι σήμερα.

Οι βασικές τεχνικές εξελίξεις που οδηγούν σε αυτήν την αναδιαμόρφωση σχεδόν βεβαία, θα περιλαμβάνουν την εκμετάλλευση του υπολογιστή ως «διανοούμενο», «παραγωγικό» όργανο, ένα σύμβουλο που χτίζεται στην ίδια την δομή του υγειονομικού συστήματος και που αυξάνει ή αντικαθιστά πολλές παραδοσιακές δραστηριότητες του ιατρού. Πράγματι, φαίνεται πιθανό ότι στο πολύ κοντινό μέλλον ο ιατρός και ο υπολογιστής θα συμμετάσχουν σε ένα συχνό διάλογο, τον υπολογιστή που συνεχώς θα κρατά σημειώσεις στο ιστορικό για τα φυσικά συμπεράσματα, τα εργαστηριακά στοιχεία και άλλα παρόμοια, προειδοποιώντας τον ιατρό στις πιθανότερες διαγνώσεις και προτείνουν το κατάλληλο και ασφαλέστερο σχέδιο δράσης. Αυτό το όραμα μόνο αργά έρχεται στην πραγματικότητα. Οι τεχνικές που απαιτούνται για να εκτελέσουν τα προγράμματα υπολογιστών για να επιτευχθούν αυτοί οι στόχοι είναι ακόμα αόριστες και πολλοί άλλοι παράγοντες επηρεάζουν την αποδοχή των προγραμμάτων αυτών. Αυτή η εργασία είναι μια εισαγωγή στον τομέα της τεχνητής νοημοσύνης στην ιατρική.

## Ορισμοί

Τι σημαίνει «τεχνητή νοημοσύνη (Α.Ι.) στην ιατρική»; Ένα εισαγωγικό εγχειρίδιο καθορίζει την τεχνητή νοημοσύνη (Α.Ι.) (αποκαλούμενη ΑΙ) με αυτό τον τρόπο: Τεχνητή νοημοσύνη (Α.Ι.) είναι η μελέτη των ιδεών που επιτρέπει στους υπολογιστές να κάνουν τα πράγματα που κάνουν τους ανθρώπους να φαίνονται ευφυείς... Οι κεντρικοί στόχοι της τεχνητής νοημοσύνης είναι να κατασταθούν οι υπολογιστές πιο χρήσιμοι και να γίνουν κατανοητές οι αρχές που καθιστούν τη νοημοσύνη πιθανή. Αυτό είναι ένας μάλλον απλός καθορισμός, αλλά ενσωματώνει ορισμένες υποθέσεις για την ιδέα της νοημοσύνης και τη σχέση μεταξύ του ανθρώπινου συλλογισμού και του υπολογισμού που είναι, σε μερικούς κύκλους, αρκετά αμφισβητούμενες. Η σύζευξη της μελέτης για το πώς να καταστήσει τους υπολογιστές χρήσιμους με τη μελέτη των αρχών που κρύβονται κάτω από την ανθρώπινη νοημοσύνη σαφώς υπονοεί ότι ο ερευνητής αναμένει τα δύο να είναι συνδεδεμένα. Πράγματι, στον πρόσφατα-αναπτυσσόμενο τομέα της γνωστικής επιστήμης, τα πρότυπα υπολογιστών της σκέψης χρησιμοποιούνται ρητά για να περιγράψουν τις ανθρώπινες ικανότητες.

## Θεωρία αποφάσεων

Η θεωρία αποφάσεων είναι μια μαθηματική θεωρία της λήψης αποφάσεων κάτω από την αβεβαιότητα. Λαμβάνοντας υπόψη αυτά τα στοιχεία, η θεωρία αποφάσεων προσφέρει μια κανονιστική, λογική θεωρία της βέλτιστης λήψης απόφασης που ωθείται από τους επαγγελματίες της ως αποτελεσματική τεχνική για τα ιατρικά προβλήματα λήψης αποφάσεων. Υπάρχουν αρκετά στοιχεία για τα οποία οι περισσότεροι άνθρωποι που είναι υπεύθυνοι για τη λήψη αποφάσεων, δεν έχουν εκπαιδευτεί πάνω σε αυτά και παρεκκλίνουν από το πρότυπο κατά τη διάρκεια λήψης αποφάσεων. Η θεωρία είναι εντούτοις να απευθυνθεί για βοήθεια για να καταστήσει ρητές τις βάσεις της λήψης αποφάσεων και να ενημερωθεί για οποιεσδήποτε υπάρχουσες διαφωνίες μεταξύ των υπεύθυνων για τη λήψη αποφάσεων. Πολυάριθμα προγράμματα υπολογιστών για τη λήψη αποφάσεων στις μικρές περιοχές της ιατρικής έχει υιοθετήσει το θεωρητικό φορμαλισμό απόφασης. Τα κύρια μειονεκτήματα της θεωρητικής προσέγγισης των αποφάσεων είναι οι δυσκολίες από τη λήψη των λογικών εκτιμήσεων των πιθανοτήτων για μια ιδιαίτερη ανάλυση. Αν και τεχνικές όπως η ανάλυση ευαισθησίας βοηθούν πολύ στην υπόδειξη ποιες πιθανές ανακρίβειες είναι ασήμαντες, η έλλειψη επαρκών στοιχείων συχνά αναγκάζει τις τεχνητές απλοποιήσεις του προβλήματος και μειώνει την εμπιστοσύνη στην έκβαση της ανάλυσης. Οι προσπάθειες να

επεκταθούν αυτές οι τεχνικές στους μεγάλους ιατρικούς τομείς στις οποίες μπορούν να συμβούν πολλές αναταραχές, δεν ήταν επιτυχής. Η χαρακτηριστική γλώσσα της πιθανότητας και χρησιμότητα δεν είναι αρκετά πλούσια ώστε να συζητήσει τέτοια ζητήματα και η επέκτασή του μέσα στο αρχικό πνεύμα οδηγεί σε τεράστια προβλήματα αποφάσεων. Παραδείγματος χάριν, κάποιος θα μπορούσε να χειριστεί το πρόβλημα των πολλαπλών αναταραχών με την εξέταση όλων των πιθανών υποσυνόλων των αρχικών αναταραχών ως αμοιβαία ανταγωνιστική υπόθεση. Το σύνολο των κύριων και υποθετικών πιθανοτήτων που απαιτούνται για μια τέτοια ανάλυση είναι, εντούτοις, εκθετικά μεγαλύτερος από αυτόν που απαιτείται για το αρχικό πρόβλημα, και αυτός είναι απαράδεκτος. Μια δεύτερη δυσκολία για την ανάλυση αποφάσεων είναι η σχετικά μυστήρια αιτιολόγηση ενός προγράμματος θεωρητικών εξηγήσεων που πρέπει να κατανοηθεί με αριθμητικούς χειρισμούς που περιλαμβάνονται στους αναμενόμενους υπολογισμούς αξίας, η οποία δεν είναι ένας φυσικός τρόπος για τους περισσότερους ανθρώπους.



## Ιστορική Αναδρομή

Η ιστορία της τεχνητής νοημοσύνης και της ρομποτικής ξεκινά ήδη από την Αρχαία Ελλάδα και μάλιστα στη μυθολογία. Το πρώτο “ρομπότ”, δηλαδή μηχανικός άνθρωπος, θα μπορούσε να θεωρηθεί ο μυθικός Τάλως. Ένας ανθρωπόμορφος γίγαντας, με σώμα από χαλκό, ο οποίος σύμφωνα με τον Απολλόδωρο, κατασκευάστηκε από το θεός Ηφαιστο με αποστολή να φυλάει την μινωική Κρήτη. Τα πρώτα αυτόματα κατασκευάζονται από το μηχανικό και γεωμέτρη, Ήρωνα, τον 1ο αιώνα π.Χ. Ο Ήρων ο Αλεξανδρεύς κατασκεύασε μηχανές όπως το κινητό αυτόματο (αυτόνομη συσκευή, ικανή να κινείται από μόνη της) και τον μηχανικό κερματοδέκτη που λειτουργούσε με νομίσματα και έδινε νερό. Από την άλλη, η θεμελίωση των κανόνων της λογικής γίνεται από τον Αριστοτέλη τον 4ο αιώνα π.Χ.



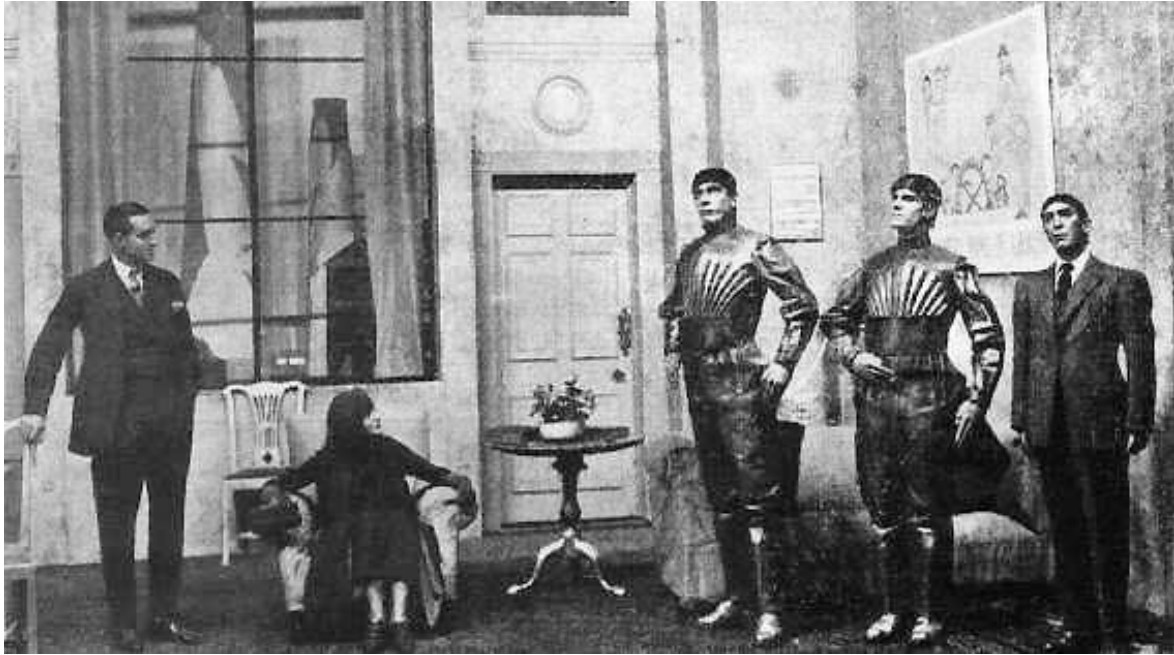
Εικόνα 1.2 Ο μυθικός Τάλως αποτελεί την πρώτη μορφή ρομπότ στην ιστορία της Ανθρωπότητας (ασημένιο δίδραχμο του 3ου αι. π.Χ.).

<http://www.explorecrete.com/mythology/images/talos-coin-1.gif>

Πρώτος αυτός διατύπωσε τους νόμους της ανθρώπινης, αλλά και τεχνητής, νόησης και τους τρόπους του συλλογισμού, στο Όργανον (“Περί Ερμηνείας”, “Κατηγορίαι”, “Αναλυτικά Πρότερα”, “Αναλυτικά Ύστερα”, “Τοπικοί και Σοφιστικοί Έλεγχοι”).

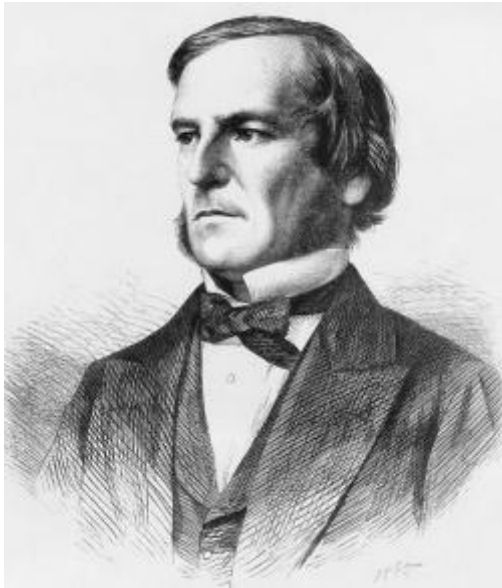
Τους επόμενους αιώνες, φιλόσοφοι, μαθηματικοί και μηχανικοί, με απαρχή τους Άραβες, κατασκευάζουν αυτόματες μηχανές, βάζοντας τις βάσεις για τα ρομπότ και τους ηλεκτρονικούς υπολογιστές, που θα έρθουν πολύ αργότερα, τον 20ο αιώνα. Τέτοια παραδείγματα, τα μουσικά αυτόματα του Άραβα μηχανικού Αλ Τζαζάρι τον 12ο αιώνα, τα μηχανικά ζώα του Βενετσιάνου μηχανικού Τζιοβάνι Φοντάνα, τον 13ο αιώνα και βέβαια ο “μηχανικός ιππότης” (ανάμεσα στα πολλά άλλα αυτόματα) του Λεονάρντο Ντα Βίντσι, το 1495. Ίσως τον πιο σημαντικό σταθμό σε όλη αυτήν την ιστορική περίοδο, αποτελεί η “Πασκαλίνα“. Μηχανική αριθμομηχανή του Γάλλου μαθηματικού Μπλεζ Πασκάλ, η οποία κατασκευάστηκε μεταξύ του 1641 και 1644, και μπορούσε να κάνει πράξεις πρόσθεσης και αφαίρεσης. Θα πρέπει βέβαια να αναφερθούμε και στην Διαφορική Μηχανή του Τσαρλς Μπάμπατζ, η οποία μπορεί να έμεινε ανολοκλήρωτη, αλλά αποτελεί αναμφίβολα τον προπάτορα όλων των σύγχρονων υπολογιστών.





Εικόνα 1.3 Η λέξη “ρομπότ” πρωτοχρησιμοποιείται στο θεατρικό έργο R.U.R. του Τσέχου συγγραφέα Κάρελ Τσάπεκ  
<https://www.mixanitouxronou.gr/wp-content/uploads/2014/12/Image.jpg>

**Στη μυθοπλασία**, όπου και συχνά τίθενται οι αρχές για την εξέλιξη της ανθρώπινης τεχνολογίας, οι αυτόματες μηχανές και τα ρομπότ αποτελούν αγαπημένο αντικείμενο πασίγνωστων έργων. Τους πρώτους αιώνες τα συναντάμε σε ένα ομιχλώδες τοπίο μυστικισμού, όπως το Γκόλεμ της εβραϊκής μυθολογίας (που ο ραβί Λεβ υποτίθεται ότι καταφέρνει να κατασκευάσει τον 17ο αιώνα στην Πράγα) και το μικρό τεχνητό πλάσμα Homunculus του Παράκελσου, “φτιαγμένος από μαγνητισμό, αλχημεία και σπέρμα”. Η λογοτεχνία, στα πρώτα της βήματα στην επιστημονική φαντασία από τον 18ο αιώνα είναι γεμάτη με αυτόματες μηχανές. Η Μηχανή (The Engine) αποτελεί ίσως την πρώτη αναφορά σε έναν νοήμονα υπολογιστή, στο πασίγνωστο βιβλίο του Τζόνθαν Σουίφτ “Τα ταξίδια του Γκιούλιβερ” (1726). Βιβλίο-σταθμό αποτελεί βέβαια ο “Φρανκενστάιν” της Μέρι Σέλεϊ (1816), όπου μπορεί το “τέρας” του Δρ. Βίκτορ Φρανκενστάιν μπορεί να μην είναι μηχανικό, δεν παύει ωστόσο να αποτελεί ένα άγνυχο αυτόματο που παίρνει ζωή μέσω ηλεκτρισμού. Εκεί μάλιστα τίθενται και τα πρώτα φιλοσοφικά ερωτήματα, σχετικά με την ηθική στον αγώνα για τον “τεχνητό άνθρωπο”. Τέλος, το 1921 είναι η χρονιά που ακούγεται για πρώτη φορά η λέξη “ρομπότ“, στην πρεμιέρα του θεατρικού έργου R.U.R. (Rossum’s Universal Robots) του Τσέχου συγγραφέα Κάρελ Τσάπεκ, το οποίο εκτυλίσσεται σε ένα εργοστάσιο κατασκευής “συνθετικών ανθρώπων”, που αποκαλούνται “ρομπότ”.



Εικόνα 1.4 Τζορτζ Μπουλ (1815-1864): Άγγλος μαθηματικός και φιλόσοφος. Πρωτοπόρος της συστηματικής μελέτης της λογικής. Στην Άλγεβρα Μπουλ θεμελιώνεται η λειτουργία των υπολογιστών

[https://upload.wikimedia.org/wikipedia/commons/thumb/6/6c/George\\_Boole.jpg/220px-George\\_Boole.jpg](https://upload.wikimedia.org/wikipedia/commons/thumb/6/6c/George_Boole.jpg/220px-George_Boole.jpg)

Στον τομέα των επιστημών, η αναζήτηση των αρχών της τεχνητής νοημοσύνης, πέρα από την αριστοτέλεια “λογική”, μας οδηγεί στο 1679, όταν ο Γερμανός φιλόσοφος και μαθηματικός, ο πρώτος ίσως επιστήμονας της πληροφορικής, Γκότφριντ Λάμπνιτς διατυπώνει τις αρχές του δυαδικού συστήματος (0 και 1). Πάνω σε αυτό θα θεμελιωθεί η Άλγεβρα Μπουλ, το 1854, όταν ο Βρετανός μαθηματικός Τζορτζ Μπουλ θα επιχειρήσει να «διερευνήσει τους θεμελιώδεις νόμους των μηχανισμών που διέπουν την νόηση στην ανάπτυξη της συλλογιστικής, για να δώσει έκφραση σε αυτούς στη συμβολική γλώσσα του λογισμού». Η Άλγεβρα Μπουλ είναι θεμελιώδους σημασίας για την επιστήμη της Πληροφορικής και αποτελεί την βάση για την θεωρητική μελέτη του πεδίου της λογικής σχεδίασης. Λίγο νωρίτερα, το 1837, ο μαθηματικός

και φιλόσοφος Μπέρναρντ Μπολτζάνο αναπτύσσει τις κεντρικές έννοιες της λογικής και την έννοια της λογικής συνέπειας. Θεωρείται πρόδρομος της σύγχρονης συμβολικής λογικής και της σημασιολογίας. Άλλος ένας σημαντικός σταθμός, το τρίτομο έργο “Principia Mathematica” (1913) των βρετανών μαθηματικών Μπέρτραντ Ράσελ και Άλφρεντ ΝορθΓουάιτχεντ, όπου επιχειρείται η κωδικοποίηση όλων των έως τότε μαθηματικών διατυπώσεων μέσα από ένα συνεπές και πλήρες σύστημα συμβολικής λογικής.

Έως τις αρχές του 20ου αιώνα, όλες οι βάσεις για την ανάπτυξη της τεχνητής νοημοσύνης και της σύγχρονης ρομποτικής έχουν τεθεί. Απομένει η έμπρακτη υλοποίησή τους, η οποία, ειδικά μετά την κατασκευή των πρώτων ηλεκτρονικών υπολογιστών, δεν θα αργήσει.

## ΧΡΟΝΟΛΟΓΙΟ

1943

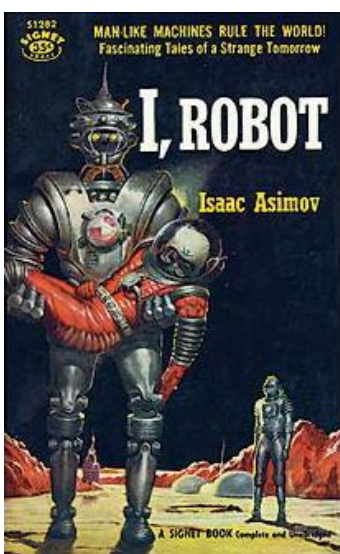
Οι Γουόρεν Μακάλοχ και Γουόλτερ Πιτς προτείνουν ένα αλγόριθμο μοντελοποίησης ενός τεχνητού νευρωνικού δικτύου, βασισμένοι στα μαθηματικά. Το Τεχνητό Νευρωνικό Δίκτυο (ANN – Artificial Neural Network) αποτελεί το ισοδύναμο του ανθρώπινου εγκεφάλου για τα ρομπότ. Είναι ικανό να αποθηκεύει πληροφορίες, να αναγνωρίζει σχήματα και να κάνει συσχετισμούς.

1944

Η Θεωρία των Παιγνίων, ένας πολύτιμος αρωγός στην έρευνα της Τεχνητής Νοημοσύνης, διατυπώνεται για πρώτη φορά από τον μαθηματικό Τζον φον Νόιμαν και τον οικονομολόγο Όσκαρ Μόργκενστερν.

1948

Ο καθηγητής μαθηματικών στο MIT, Νόρμπερτ Βίνερ, θεμελιώνει τον κλάδο της Κυβερνητικής με το σύγγραμμα “Κυβερνητική: Η έλεγχος και επικοινωνία στα ζώα και τις μηχανές”. Ο Βίνερ διατυπώνει τις θεωρίες του βασισμένος στην εμπειρία του κατά τον Β΄ Παγκόσμιο Πόλεμο στα συστήματα αντιαεροπορικής άμυνας, που έπρεπε να προβλέπουν την κίνηση των αεροσκαφών. Η Κυβερνητική ή αλλιώς «Επιστήμη των Συστημάτων», θα επηρεάσει καταλυτικά την έρευνα για την τεχνητή νοημοσύνη (A.I.) (A.I.).



1950

Ο Αμερικανός συγγραφέας Ισαάκ Ασίμοφ δημιουργεί τη συλλογή εννέα διηγημάτων επιστημονικής φαντασίας, “I Robot“. Σε αυτές τις ιστορίες το μέλλον κατά τον Ασίμοφ περιγράφεται ως ένας κόσμος όπου οι άνθρωποι και τα ρομπότ συμβιώνουν και σχεδόν δεν διακρίνονται μεταξύ τους. Εκεί διατυπώνονται οι Τρεις βασικοί Νόμοι της Ρομποτικής, που θεωρούνται έως και σήμερα αξιωματικοί ακόμη και στο χώρο των επιστημών:

Εικόνα 1.5 I Robot  
[http://upload.wikimedia.org/wikipedia/en/8/8e/I\\_Robot\\_-\\_Runaround.jpg](http://upload.wikimedia.org/wikipedia/en/8/8e/I_Robot_-_Runaround.jpg)

1. *Το ρομπότ δε θα κάνει κακό σε άνθρωπο*
2. *Το ρομπότ εκτελεί τις διαταγές που του δίνουν οι άνθρωποι, εκτός αν αυτές έρχονται σε αντίθεση με τον πρώτο νόμο*
3. *Το ρομπότ οφείλει να προστατεύει την ύπαρξή του, εφόσον αυτό δεν συγκρούεται με τον πρώτο και τον δεύτερο νόμο*

Ο Κλοντ Σάνον, ο πατέρας της Θεωρίας της Πληροφορίας, δημοσιεύει ένα άρθρο για τον προγραμματισμό υπολογιστή στο σκάκι (Programming a Computer for Playing Chess). Ο αλγόριθμος του Σάνον είναι ένας από τους πρώτους αλγόριθμους τεχνητής νοημοσύνης και περιγράφει τον τρόπο που θα μπορούσε να προγραμματιστεί ένας υπολογιστής ώστε να παίζει σκάκι με ανθρώπινο αντίπαλο.

Ο Άλαν Τούρινγκ απαντά με μία δημοσίευση, στο ερώτημα που θέτει ο ίδιος “μπορούν οι μηχανές να σκέπτονται;”. Στο άρθρο “Computing Machinery and Intelligence” προτείνει ένα τεστ, στο οποίο ένας κριτής θα πρέπει να μην μπορεί να ξεχωρίσει αν οι απαντήσεις, σε ερωτήσεις που διατυπώνονται στην ανθρώπινη γλώσσα, προέρχονται από άνθρωπο ή μηχανή. Μέχρι σήμερα κανένας υπολογιστής δεν έχει καταφέρει να περάσει το τεστ Τούρινγκ.

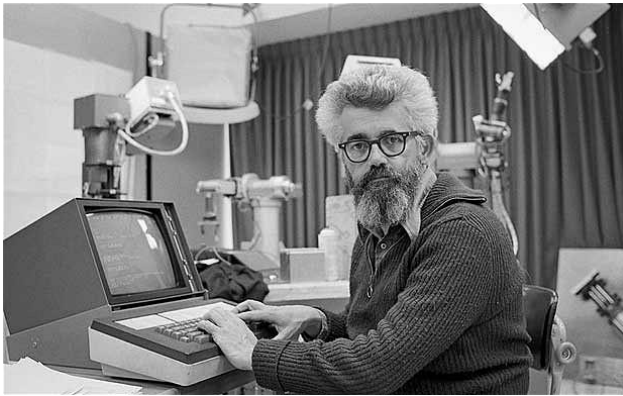
### 1951

Τα πρώτα λειτουργικά προγράμματα τεχνητής νοημοσύνης σχεδιάζονται στον υπολογιστή Ferranti Mark 1, του Πανεπιστημίου του Μάντσεστερ. Ένα παιχνίδι ντάμα κι ένα παιχνίδι σκάκι.

### 1954

Κατασκευάζεται από τους Άλεν Νιούελ, Κλιφ Σο και Χέρμπερτ Σάιμον, η γλώσσα IPL (Information Processing Language). Θεωρείται η πρώτη γλώσσα τεχνητής νοημοσύνης και βασίζεται στα θεωρήματα που περιλαμβάνονται στο βιβλίο “Principia Mathematica” των Ράσελ και Γουάιχεντ.

1956



Εικόνα 1.6 Τζον Μακάρθι (1927-2011), ο “πνευματικός πατέρας” της τεχνητής νοημοσύνης.  
[http://sansimeracomputers.files.wordpress.com/2013/11/john\\_mccarthy.jpg?w=620&h=388](http://sansimeracomputers.files.wordpress.com/2013/11/john_mccarthy.jpg?w=620&h=388)

Η έκφραση “Τεχνητή νοημοσύνη (A.I.)” πρωτοδιατυπώνεται στον τίτλο συνεδρίου που οργανώνει ο Τζον Μακάρθι, στο Ντάρτμουθ, των ΗΠΑ. Ο ορισμός του Μακάρθι για την τεχνητή νοημοσύνη (A.I.) είναι «η επιστήμη και μεθοδολογία της

δημιουργίας νοούντων μηχανών»

Αν και έχουν προηγηθεί προγράμματα σκάκι και ντάμας, το πρόγραμμα “Logic

Theorist” που έγραψε ο Νιούελ τη διετία 1955-56, θεωρείται το πρώτο που είναι ειδικά φτιαγμένο για να μιμείται τον άνθρωπο στην επίλυση απλών προβλημάτων. Σε αυτό εισάγεται ο όρος “ευριστική” (heuristics) που αποτελεί το βασικότερο τομέα έρευνας στην τεχνητή νοημοσύνη (A.I.) (A.I.).

1958

Ο Τζον Μακάρθι αναπτύσσει τη LISP (List Processing). Η δεύτερη γλώσσα προγραμματισμού υψηλού επιπέδου, μετά τη FORTRAN, είναι ιδανική για την έρευνα στην τεχνητή νοημοσύνη (A.I.) (A.I.). Η LISP πρωτοπόρησε στην εισαγωγή πολλών ιδεών στην επιστήμη των υπολογιστών, όπως οι δομές δένδρων, η αυτόματη διαχείριση αποθήκευσης δεδομένων, οι δυναμικοί τύποι, ο αντικειμενοστραφής προγραμματισμός, και ο μεταγλωττιστής που μεταγλωττίζει τον εαυτό του.

1961

Το Unimate, το πρώτο ρομπότ βιομηχανικής παραγωγής μπαίνει σε λειτουργία, στη γραμμή παραγωγής αυτοκινήτων της General Motors και συγκεκριμένα στο τμήμα χύτευσης.

1963

Ο Τζον Μακάρθι ιδρύει το πρώτο πανεπιστημιακό εργαστήριο τεχνητής νοημοσύνης. Το AI Lab του Στάνφορντ.

1965



Ο Τζόζεφ Γουαϊζενμπάουμ κατασκευάζει, στο MIT, την ELIZA. Το πρώτο διαδραστικό πρόγραμμα τεχνητής νοημοσύνης που μπορεί (θεωρητικά) να διεξάγει λογικούς διαλόγους στην αγγλική γλώσσα.

Μία ομάδα στο Στάνφορντ, υπό τον Εντ Φαϊγκενμπάουμ, κατασκευάζει το DENDRAL. Το πρώτο έμπειρο σύστημα (expert system). Δηλαδή, ένα πρόγραμμα που μιμείται την ικανότητα του εμπειρογνώμονα στη λήψη αποφάσεων, αντί να εκτελεί μία ακριβή και προδιαγεγραμμένη διαδικασία, όπως στο συμβατικό προγραμματισμό. Με μία σειρά ερωτημάτων “if-then” το DENDRAL επιχειρούσε να εξακριβώσει την μοριακή δομή οργανικών ενώσεων. Τα έμπειρα συστήματα αποτελούν θεμελιώδη κλάδο της τεχνητής νοημοσύνης.

### 1969

Κατασκευάζεται το ρομπότ Shakey στο πανεπιστήμιο του Στάνφορντ. Το πρώτο κινητό ρομπότ γενικής χρήσης που μπορεί να αντιλαμβάνεται, αναλύει και εκτελεί απλές εντολές. Ο “Shakey το ρομπότ” αποτελείται από μονάδα επεξεργασίας, κάμερα, δύο αποστασιόμετρα και αισθητήρες σύγκρουσης, ηλεκτρικό μοτέρ και κεραία ραδιοζεύξης για την επικοινωνία με σταθερό υπολογιστή.



Εικόνα 1.7 Shakey

### 1972

Αναπτύσσεται η πρώτη λογική γλώσσα προγραμματισμού, η Prolog (Programation et Logique). Ειδικά για την έρευνα στην τεχνητή νοημοσύνη (A.I.) (A.I.), οι λογικές γλώσσες μπορούν να διατυπώσουν ένα πλαίσιο κανόνων και δεδομένων μέσα στο οποίο εκτελείται ο αλγόριθμος.

<http://sansimeracomputers.files.wordpress.com/2013/11/shakey1.jpg?w=250&h=250>

### 1976

Κατασκευάζεται ο Freddy II στο Πανεπιστήμιο του Εδιμβούργου. Μία από τις πρώτες υλοποιήσεις ρομπότ συναρμολόγησης που μπορεί να αναγνωρίζει βασικά σχήματα και να “διδάσκεται” νέες διαδικασίες.

### 1979



Εικόνα 1.8 TI Explorer

[http://sansimeracomputers.files.wordpress.com/2013/11/ti\\_explorer1.jpg?w=250&h=250](http://sansimeracomputers.files.wordpress.com/2013/11/ti_explorer1.jpg?w=250&h=250)

Το Stanford Cart, ένα ερευνητικό πρότζεκτ ρομποτικού οχήματος που έχει ξεκινήσει από τις αρχές της δεκαετίας του '60 στο πανεπιστήμιο του Στάνφορντ, γίνεται το πρώτο αυτόνομο κινούμενο όχημα. Μπόρεσε να διασχίσει ένα δωμάτιο γεμάτο με εμπόδια (καρέκλες) χωρίς ανθρώπινη παρέμβαση. Κατασκευάζονται τα πρώτα LISP machines. Προσωπικοί υπολογιστές-workstations που τρέχουν σε περιβάλλον LISP. Οι υπολογιστές που βασίζονται στη LISP αναπτύσσονται, μετά τα μέσα της δεκαετίας του '70, από το MIT και τη Xerox. Η γραμμή του MIT οδηγεί και σε υπολογιστές μαζικής κατασκευής, όπως τα Symbolics 3600/3640/X1200 και TI Explorer(φωτογραφία). Η ανάπτυξη εφαρμογών τεχνητής νοημοσύνης αρχίζει να γίνεται καταναλωτικό προϊόν.

## 1982

Μπαίνει σε λειτουργία το πρώτο σύστημα αυτόματης αναγνώρισης δακτυλικών αποτυπωμάτων (AFIS – Automated Fingerprint Identification System), κατασκευασμένο από τη NEC.

## 1985

Γίνεται επίδειξη του προγράμματος AARON. Μία εφαρμογή που αναπτύσσει για παραπάνω από μία δεκαετία ο σχεδιαστής Άαρον Κοέν και μπορεί να ζωγραφίζει, χωρίς την παρέμβαση ανθρώπου, σύνθετα σχήματα που αναπαριστούν πραγματικά αντικείμενα.

## 1987

Ολοκληρώνεται η πρώτη υλοποίηση MCL (Macintosh Common Lisp), μέσω της οποίας ο Apple Macintosh μετατρέπεται σε LISP machine.

## 1991

Στον “Πόλεμο του Κόλπου” μπαίνει σε λειτουργία το πρόγραμμα DART (Dynamic Analysis and Replanning Tool). Η πρώτη εφαρμογή τεχνητής νοημοσύνης για στρατιωτική χρήση, που προγραμματίζει μία πλειάδα ενεργειών Διοικητικής Μέριμνας (διαχείριση και μεταφορά προσωπικού, υλικών, πολεμοφοδίων και καυσίμων).

## 1994

Γίνεται με επιτυχία η επίδειξη του συστήματος VaMP που μετατρέπει ένα αυτοκίνητο σε αυτόνομο ρομποτικό όχημα. Ένα μοντέλο της Mercedes μπόρεσε να καλύψει απόσταση άνω των 1.000 χιλιομέτρων σε αυτοκινητόδρομο της Γαλλίας και σε συνθήκες βαριάς κίνησης, χωρίς την παρέμβαση ανθρώπου και με ταχύτητες έως και 130 χ.α.ω.

### 1997

Ο Deep Blue, ο υπερυπολογιστής της IBM που προγραμματίστηκε για να παίζει σκάκι με ανθρώπινο αντίπαλο, καταφέρνει να κερδίσει τον παγκόσμιο πρωταθλητή Γκάρι Κασπάροφ.

### 1998



Εικόνα 1.9 Furby

<http://sansimeracomputers.files.wordpress.com/2013/11/furby.jpg?w=250&h=250>

Κυκλοφορεί ο Furby. Κατασκευασμένο από την Tiger Electronics, είναι το πρώτο ρομποτικό παιχνίδι βασισμένο στις αρχές της τεχνητής νοημοσύνης, που μπαίνει στη μαζική παραγωγή. Θα ακολουθήσει μία νέα γενιά προγραμματιζόμενων ηλεκτρονικών παιχνιδιών που μπορούν να εκτελούν εντολές και να αντιλαμβάνονται περιβαλλοντικά ερεθίσματα. Τα λεγόμενα smart toys (έξυπνα παιχνίδια).

### 1999

Κυκλοφορεί το ρομποτικό σκυλάκι AIBO από τη Sony, χαρακτηριζόμενο από την εταιρεία ως “ρομπότ οικιακής διασκέδασης”. Από το 1999 και κάθε χρόνο, έως το 2005, η Sony παρουσιάζει ένα νέο μοντέλο AIBO. Το 2006, όταν πια έχει ολοκληρωθεί η παραγωγή του, το AIBO περνά στο “Robot Hall of Fame” του Πανεπιστημίου Κάρνεγκι Μέλον, με την παρατήρηση “αντιπροσωπεύει το πιο εξελιγμένο ρομπότ μαζικής κατανάλωσης”.



Εικόνα 1.10 AIBO

<https://i.ytimg.com/vi/rnR0rVif5yM/maxresdefault.jpg>



## 2000

Ο ASIMO της Honda κάνει το ντεμπούτο του. Ένα από τα πιο εξελιγμένα ανθρωπόμορφα ρομπότ ικανό να διεξάγει βασικής μορφής επικοινωνία με τον άνθρωπο. Ο ASIMO μπορεί να αναγνωρίζει φωνητικές εντολές, κινήσεις, ακόμη και εκφράσεις, σχήματα και πρόσωπα.

Το ρομπότ Nomad τίθεται σε λειτουργία σε απομονωμένες περιοχές της Ανταρκτικής. Αυτόνομα κινούμενο, το Nomad ερευνά και ταξινομεί δείγματα πετρωμάτων από μετεωρίτες.

## 2002

Άλλη μία καινοτομία στην αξιοποίηση της τεχνητής νοημοσύνης για οικιακή χρήση. Η Roomba, της iRobot, είναι η πρώτη ρομποτική ηλεκτρική σκούπα που μπορεί να καθαρίσει πατώματα κινούμενη αυτόνομα και αντιλαμβανόμενη φυσικά εμπόδια.

## 2004

Οργανώνεται το πρώτο DARPA Grand Challenge. Ένας αγώνας δρόμου για αυτόνομα κινούμενα οχήματα, χρηματοδοτούμενος από την DARPA (Defense Advanced Research Projects Agency). Στον πρώτο αγώνα που διεξάγεται στην έρημο Μοχάβε, κανένα όχημα δεν



Εικόνα 1.11 VW Tuareg R5 (Stanford Racing Team)

[http://sansimeracomputers.files.wordpress.com/2013/11/darpa\\_tuareg.jpg?w=300&h=191](http://sansimeracomputers.files.wordpress.com/2013/11/darpa_tuareg.jpg?w=300&h=191)

καταφέρνει να ολοκληρώσει τη διαδρομή. Τ 2005 ωστόσο, ένα ειδικά διαμορφωμένο VW Touareg R5 από 65 φοιτητές, καθηγητές και ερευνητές του πανεπιστημίου του Στάνφορντ, καταφέρνει να ολοκληρώσει τη διαδρομή και να κερδίσει το βραβείο των 2 εκατ. δολαρίων. Θα τερματίσουν ακόμη 4 αυτοκίνητα από τις 23 συμμετοχές.

## 2006

Ολοκληρώνεται η πρώτη φάση δημιουργίας του πρώτου συνθετικού εγκεφάλου, αντίγραφο του εγκεφάλου των θηλαστικών σε επίπεδο μορίου. Ένα φιλόδοξο πρόγραμμα του Πολυτεχνείου της Λωζάνης, το οποίο αξιοποιεί τον υπερυπολογιστή Blue Gene της IBM σε συνδυασμό με το λογισμικό NEURON, μέσω του οποίου μοντελοποιούνται σύνθετα νευρωνικά δίκτυα.

## 2010

Ολοκληρώνεται στον Καναδά η πρώτη χειρουργική επέμβαση από ρομπότ. Πιο συγκεκριμένα μία προστατεκτομή εκτελείται εξολοκλήρου (σ.σ. από τη φάση της αναισθησίας) από ρομποτικά συστήματα.

## 2011

Ο υπολογιστής Watson της IBM, ειδικά κατασκευασμένος να απαντά σε ανθρώπινα ερωτήματα διατυπωμένα σε φυσική γλώσσα, καταφέρνει να κερδίσει σε διαγωνισμό τους νικητές του γνωστού παιχνιδιού γνώσεων της αμερικάνικης τηλεόρασης, Jeopardy.

## ΣΗΜΕΡΑ

Οι εφαρμογές τεχνητής νοημοσύνης βρίσκονται παντού. Το μικρότερο και πιο καθημερινό δείγμα, με πολλαπλές τέτοιες εφαρμογές είναι το κοινό smartphone. Εφαρμογές που μπορούν να πάρουν βασικές αποφάσεις, να υποδείξουν λύσεις σε συγκεκριμένα προβλήματα και βέλτιστες διαδρομές, να κάνουν έξυπνες αναζητήσεις, ακόμη και να επικοινωνήσουν σε φυσική γλώσσα. Δύο τέτοια παραδείγματα τα apps Siri και Google Now, για τα λειτουργικά συστήματα iOS και Android αντίστοιχα.

Ακόμη όμως και στην εποχή των PetaFLOPS (σ.σ. υπολογιστές που μπορούν να εκτελέσουν ένα τετράκις εκατομμύρια πράξεις το δευτερόλεπτο), καμία μηχανή δεν έχει καταφέρει να κερδίσει το Τεστ Τούρινγκ. Να πείσει δηλαδή ότι μπορεί να απαντά σε σύνθετα ερωτήματα όπως ακριβώς ένας άνθρωπος. Σύμφωνα με τις προβλέψεις πάντως, εντός της δεκαετίας θα γίνει εφικτή η προσομοίωση του ανθρώπινου εγκεφάλου από υπερυπολιστής.

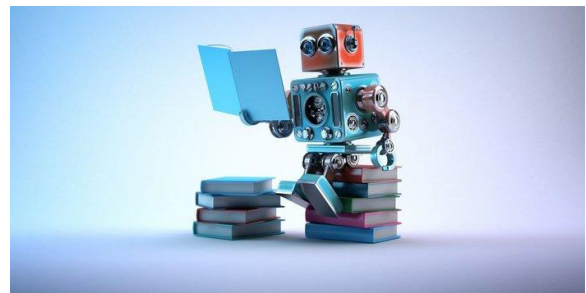
## Κεφάλαιο 2:

### Εισαγωγή

Στο προηγούμενο κεφάλαιο έγινε εκτενείς αναφορά στην τεχνητή νοημοσύνη (Α.Ι.) και στην προϊστορία της γύρω από τον άνθρωπο. Στο παρόν κεφάλαιο θα εμβαθύνουμε λίγο περισσότερο στην μηχανική μάθηση και θα εξηγήσουμε κάποιες πολύ σημαντικές πτυχές της, καθώς αποτελεί βασικότατο εργαλείο για την παρούσα πτυχιακή.

### Μηχανική μάθηση

Μηχανική μάθηση είναι υποπεδίο της επιστήμης των υπολογιστών που αναπτύχθηκε από τη μελέτη της αναγνώρισης προτύπων και της υπολογιστικής θεωρίας μάθησης στην τεχνητή νοημοσύνη (Α.Ι.). Το 1959, ο Arthur Samuel ορίζει τη μηχανική μάθηση ως "Πεδίο μελέτης που δίνει στους υπολογιστές την ικανότητα να μαθαίνουν,



Εικόνα 2.1 [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQFr4eKX-\\_KC\\_r0XUt74KollLyxgOmfF7MBefelpUGHTVFlb65m](https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQFr4eKX-_KC_r0XUt74KollLyxgOmfF7MBefelpUGHTVFlb65m)

χωρίς να έχουν ρητά προγραμματιστεί". Η μηχανική μάθηση διερευνά τη μελέτη και την κατασκευή αλγορίθμων που μπορούν να μαθαίνουν από τα δεδομένα και να κάνουν προβλέψεις σχετικά με αυτά. Τέτοιοι αλγόριθμοι λειτουργούν κατασκευάζοντας μοντέλα από πειραματικά δεδομένα, προκειμένου να κάνουν προβλέψεις βασιζόμενες στα δεδομένα ή να εξάγουν αποφάσεις που εκφράζονται ως το αποτέλεσμα.

Η μηχανική μάθηση είναι στενά συνδεδεμένη και συχνά συγχέεται με υπολογιστική στατιστική, ένας κλάδος, που επίσης επικεντρώνεται στην πρόβλεψη μέσω της χρήσης των υπολογιστών. Έχει ισχυρούς δεσμούς με την μαθηματική βελτιστοποίηση, η οποία παρέχει μεθόδους, τη θεωρία και τομείς εφαρμογής. 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.

Στο πεδίο της ανάλυσης δεδομένων, η μηχανική μάθηση είναι μια μέθοδος που χρησιμοποιείται για την επινόμενη πολύπλοκων μοντέλων και αλγορίθμων που οδηγούν στην πρόβλεψη. Τα αναλυτικά μοντέλα επιτρέπουν στους ερευνητές, τους επιστήμονες δεδομένων, τους μηχανικούς και τους αναλυτές να παράγουν αξιόπιστες αποφάσεις και αποτελέσματα και να αναδείξουν αλληλοσυσχετίσεις μέσω της μάθησης από ιστορικές σχέσεις και τάσεις στα δεδομένα.

### **Ιστορία και η σύνδεση με άλλους τομείς:**

Ως επιστημονικό εγχείρημα, η μηχανική μάθηση αναπτύχθηκε από την αναζήτηση για την τεχνητή νοημοσύνη (A.I.) (A.I.). Ήδη από την πρώιμη περίοδο της έρευνας στον τομέα της τεχνητής νοημοσύνης σε ακαδημαϊκό επίπεδο, το ζήτημα της κατασκευής μηχανών που θα μάθαιναν από δεδομένα απασχόλησε τους ερευνητές. Προσπάθησαν να προσεγγίσουν το πρόβλημα με διάφορες συμβολικές μεθόδους, καθώς και με τα λεγόμενα νευρωνικά δίκτυα. Αυτά ήταν ως επί το πλείστον perceptrons και μοντέλα, που όπως διαπιστώθηκε αργότερα ήταν επανεφευρέσεις των γενικευμένων γραμμικών μοντέλων της στατιστικής. Επίσης χρησιμοποιήθηκε η πιθανοθεωρητική λογική, ιδιαίτερα στην αυτοματοποιημένη ιατρική διάγνωση.

Ωστόσο, μια αυξανόμενη έμφαση σε προσεγγίσεις που βασίζονται στην λογική γνώση προκάλεσε ένα ρήγμα μεταξύ Τεχνητής Νοημοσύνης και μηχανικής μάθησης. Τα πιθανοθεωρητικά συστήματα μαστίζονταν από θεωρητικά και πρακτικά προβλήματα απόκτησης δεδομένων και αναπαράστασής τους. Από το 1980, έμπειρα συστήματα επικράτησαν στο πεδίο της Τεχνητής Νοημοσύνης, και ο ρόλος της στατιστικής υποχώρησε. Η εργασία σε συμβολική/βασισμένη σε γνώση εκμάθηση συνεχίστηκε εντός της TN, οδηγώντας στον επαγωγικό λογικό προγραμματισμό, αλλά οι κατευθυντήριες γραμμές της στατιστικής ήταν τώρα έξω από το χώρο της τεχνητής νοημοσύνης, στην αναγνώριση προτύπων και στην ανάκτηση πληροφοριών. Η έρευνα για νευρωνικά δίκτυα εγκαταλείφθηκε από την TN και την Επιστήμη Υπολογιστών τον ίδιο περίπου καιρό. Η ίδια επίσης κατεύθυνση ακολουθήθηκε πέρα από την TN και την πληροφορική, από ερευνητές άλλων ειδικοτήτων, συμπεριλαμβανομένων των Hopfield, Rumelhart και Hinton. Η επιτυχία ήρθε στα μέσα της δεκαετίας του 1980 με την επανεφεύρεση της μεθόδου ανάστροφης μετάδοσης(αλλιώς οπισθοδιάδοσης) - (backpropagation).

Η Μηχανική μάθηση, αναδιοργανώθηκε ως ένα ξεχωριστό πεδίο, που άρχισε να ακμάζει κατά τη δεκαετία του 1990. Η προσοχή μετατοπίστηκε από τις συμβολικές προσεγγίσεις που κληρονόμησε από την Τεχνητή νοημοσύνη (A.I.), που στόχο είχαν την αντιμετώπιση

επιλύσεων προβλημάτων πρακτικής φύσης, και δόθηκε έμφαση σε μεθόδους και μοντέλα της στατιστικής και της θεωρίας πιθανοτήτων. Επίσης επωφελήθηκε από την διαθεσιμότητα ψηφιοποιημένων πληροφοριών και της δυνατότητας να διανεμηθούν μέσω του Διαδικτύου.

Η Μηχανική μάθηση και η εξόρυξη δεδομένων συχνά χρησιμοποιούν τις ίδιες μεθόδους και επικαλύπτονται σημαντικά. Μπορούν να διακριθούν ως εξής:

Η μηχανική μάθηση εστιάζει στην πρόβλεψη, που βασίζεται σε γνωστές ιδιότητες που απορρέουν από το σύνολο εκπαίδευσης.

Η εξόρυξη δεδομένων εστιάζει στην ανακάλυψη ιδιοτήτων μη γνωστών εκ των προτέρων. Αυτό είναι το βήμα ανάλυσης στην Ανακάλυψη Γνώσης από βάσεις δεδομένων.

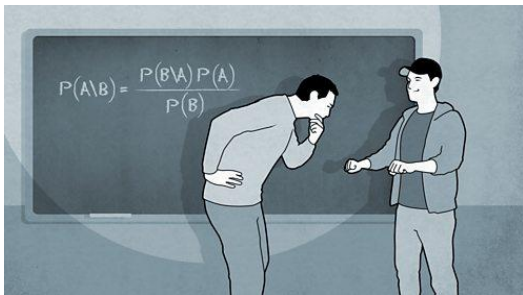
Οι δύο τομείς επικαλύπτονται με πολλούς τρόπους. Η εξόρυξη δεδομένων χρησιμοποιεί πολλές μεθόδους μηχανικής μάθησης, αλλά συχνά με διαφορετικούς στόχους. Από την άλλη πλευρά και η μηχανική μάθηση χρησιμοποιεί μεθόδους εξόρυξης δεδομένων, όπως η μη επιτηρούμενη μάθηση, ή στο στάδιο προεπεξεργασίας για να βελτιώνει την ακρίβεια της μάθησης. Ένα μεγάλο μέρος της σύγχυσης μεταξύ των δύο ερευνητικών τομέων (που συχνά έχουν ξεχωριστά συνέδρια και περιοδικά, με το ECML PKDD να αποτελεί σημαντική εξαίρεση) προκύπτει από τις βασικές υποθέσεις πάνω στις οποίες και οι δύο δουλεύουν. Όμως, στην μηχανική μάθηση η απόδοση συνήθως αξιολογείται ως προς την ικανότητα αναπαραγωγής γνώσης, την οποία ήδη κατέχουμε, ενώ στην ανακάλυψη γνώσης και την εξόρυξη δεδομένων το κλειδί είναι η ανακάλυψη γνώσης που δεν προκατέχουμε. Στην πρώτη περίπτωση μια μέθοδος επιτηρούμενης μάθησης μπορεί να έχει καλύτερα αποτελέσματα, ενώ σε μία τυπική διεργασία Ανακάλυψης Γνώσης και Εξόρυξης δεδομένων οι επιτηρούμενες μέθοδοι μάθησης δεν λειτουργούν εξαιτίας της μη διαθεσιμότητας συνόλου εκπαίδευσης

Η μηχανική μάθηση συνδέεται επίσης με την βελτιστοποίηση, πολλά προβλήματα μάθησης διατυπώνονται ως η ελαχιστοποίηση της συνάρτησης απώλειας από ένα σύνολο δεδομένων εκπαίδευσης. Η συνάρτηση απώλειας εκφράζει τη διαφορά μεταξύ των προβλέψεων του εκπαιδευμένου μοντέλου και των πραγματικών καταστάσεων του προβλήματος. Η διαφορά των δύο τομέων απορρέει από τον στόχο της γενίκευσης, ενώ οι αλγόριθμοι βελτιστοποίησης μπορούν να ελαχιστοποιήσουν την απώλεια ενός συνόλου εκπαίδευσης, η μηχανική μάθηση εστιάζει στην ελαχιστοποίηση της απώλειας σε άγνωστες καταστάσεις.

## ΧΡΟΝΟΛΟΓΙΟ

Στις μέρες μας παρατηρούμε ολοένα και περισσότερα τα αποτελέσματα της μηχανικής μάθησης σε πολλές εφαρμογές, όπως αυτόνομο αυτοκίνητο, ενεργοποιημένοι φωνητικά βοηθοί και στα κοινωνικών μέσων ενημέρωσης. Ωστόσο, οι ιδέες της μηχανικής μάθησης έχουν μακρά ιστορία και βασίζονται σε μαθηματικά από εκατοντάδες χρόνια πριν και στις τεράστιες εξελίξεις στον τομέα της πληροφορικής τα τελευταία 70 χρόνια.

1940



Εικόνα 2.2 Stellario Cama

<https://ichef.bbci.co.uk/images/ic/496xn/p03fy6mz.jpg>

Πολλές από τις μαθηματικές βάσεις της σύγχρονης μηχανικής μάθησης προηγούνται των υπολογιστών και προέρχονται από τις στατιστικές. Σημαντικές ανακαλύψεις περιλαμβάνουν το έργο του Thomas Bayes τον 18ο αιώνα, το οποίο οδήγησε τον Pierre-Simon Laplace να καθορίσει το θεώρημα του Bayes (1812). Ο Adrien-Marie Legendre ανέπτυξε επίσης τη μέθοδο Least Squares για την τοποθέτηση δεδομένων (1805) και ο Αντρέι Μάρκοφ περιέγραψε τις τεχνικές ανάλυσης που αργότερα αποκαλούν Markov Chains (1913). Αυτές οι τεχνικές είναι όλες θεμελιώδεις στη σύγχρονη μηχανική μάθηση

1948



Εικόνα 2.3 Computer Laboratory, University of Cambridge

<https://ichef.bbci.co.uk/images/ic/496xn/p067jqr8.jpg>

Στα τέλη της δεκαετίας του 1940 οι εργασίες προχώρησαν στην ανάπτυξη υπολογιστών με αποθηκευμένο πρόγραμμα που κρατούν τις οδηγίες τους (προγράμματα) στην ίδια μνήμη που χρησιμοποιείται για δεδομένα. Οι πρώτοι υπολογιστές αυτού του τύπου ξεκίνησαν τη σύγχρονη επανάσταση στον υπολογισμό. Ήταν η πειραματική μηχανή Μικροκλίμακας του Μάντσεστερ (με το παρατσούκλι «Baby») το 1948, το EDSAC του Cambridge και το Μάρτιο του Μάρκου 1 το 1949 και το EDVAC του Πανεπιστημίου της Πενσυλβανίας το 1951.



1950



Εικόνα 2.4 Alan Turing statue at Bletchley Park

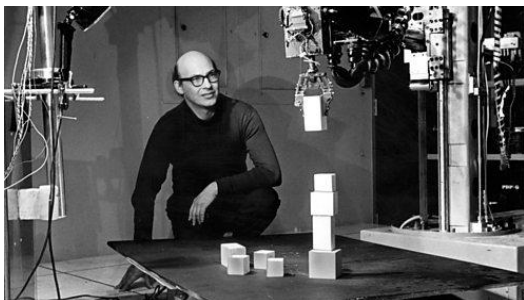
<https://ichef.bbci.co.uk/images/ic/496xn/p067jqmr.jpg>

Το 1950 ο Alan Turing δημοσίευσε τα υπολογιστικά μηχανήματα και τη νοημοσύνη, στην οποία ρώτησε: «Μπορούν οι μηχανές να σκέφτονται;» - μια ερώτηση που εξακολουθούμε να αγωνιζόμαστε. Με βάση την αυξανόμενη κατανόηση της ισχύος των υπολογιστών, το χαρτί ήταν μια από τις πρώτες προσπάθειες να περιγραφεί πώς θα μπορούσε να αναπτυχθεί η «τεχνητή» νοημοσύνη. Πρότεινε περίφημα το «παιχνίδι μίμησης», ένα τεστ για να διαπιστωθεί αν ένας υπολογιστής ήταν έξυπνος, ζητώντας από ένα άτομο να διακρίνει μεταξύ ενός ανθρώπου και ενός υπολογιστή όταν επικοινωνούσε μαζί τους τόσο μέσω δακτυλογραφημένων μηνυμάτων.

Το 1950 ο Alan Turing δημοσίευσε τα υπολογιστικά μηχανήματα και τη νοημοσύνη, στην οποία ρώτησε: «Μπορούν οι μηχανές να σκέφτονται;» - μια ερώτηση που εξακολουθούμε να αγωνιζόμαστε. Με βάση την αυξανόμενη

κατανόηση της ισχύος των υπολογιστών, το χαρτί ήταν μια από τις πρώτες προσπάθειες να περιγραφεί πώς θα μπορούσε να αναπτυχθεί η

1951



Εικόνα 2.5 Marvin Minsky at MIT in 1968

<https://ichef.bbci.co.uk/images/ic/496xn/p067jpdrg.jpg>

Ο Marvin Minsky και ο Dean Edmonds έχτισαν το πρώτο τεχνητό νευρωνικό δίκτυο - μια υπολογιστική προσομοίωση του τρόπου με τον οποίο λειτουργούν οι οργανικοί εγκέφαλοι. Ο Στοχαστικός Νευρωνικός Αναλογικός Υπολογιστής Ενίσχυσης (SNARC) έμαθε από την εμπειρία και χρησιμοποιήθηκε για να ψάξει σε ένα λαβύρινθο, όπως ένας αρουραίος σε ένα πείραμα ψυχολογίας. Κατασκευάστηκε σύμφωνα με τις αρχές του συνδέσμου, που αντιπροσωπεύουν ένα μυαλό σαν ένα δίκτυο απλών μονάδων μέσα στις οποίες μπορεί να προκύψει η νοημοσύνη. Ο Minsky συνέχισε να εργάζεται στο Εργαστήριο Τεχνητής Νοημοσύνης του MIT και έκανε πολλές άλλες σημαντικές παρεμβάσεις στη συζήτηση για τον AI. Ήταν σύμβουλος στην ταινία 2001: Μια διαστημική οδύσσεια.

Ο Marvin Minsky και ο Dean Edmonds έχτισαν το πρώτο τεχνητό νευρωνικό δίκτυο - μια υπολογιστική προσομοίωση του τρόπου με τον οποίο λειτουργούν οι οργανικοί εγκέφαλοι.

Ο Στοχαστικός Νευρωνικός Αναλογικός Υπολογιστής Ενίσχυσης (SNARC) έμαθε από την εμπειρία και χρησιμοποιήθηκε για να ψάξει σε ένα λαβύρινθο, όπως ένας αρουραίος σε ένα πείραμα

1974



Εικόνα 2.6 Sir James Lighthill, author of 'Artificial Intelligence: A General Survey'

<https://ichef.bbci.co.uk/images/ic/496xn/p031y3c4.jpg>

Το πρώτο AI 'χειμώνας'

Κατά τη διάρκεια της δεκαετίας του 1950 και της δεκαετίας του 1960 υπήρξε τεράστιος ενθουσιασμός για την έρευνα AI, αλλά οι άνθρωποι απογοητεύτηκαν όταν δεν έγιναν ανακαλύψεις. Η αποτυχία της μηχανικής μετάφρασης και της εξουδετέρωσης των δυνατοτήτων της AI οδήγησε σε μειωμένη χρηματοδότηση. Η Έκθεση Lighthill του 1973 προς το Κοινοβούλιο εξέτασε την κατάσταση της έρευνας και σημείωσε την αποτυχία να επιτύχει «μεγαλοπρεπούς στόχους». Τα πράγματα άλλαξαν στα τέλη της δεκαετίας του 1980 με νέες προσεγγίσεις, την εμφάνιση συστημάτων εμπειρογνομόνων και την επανάκτηση παλαιών ιδεών που θα μπορούσαν να εφαρμοστούν σε νέες ρυθμίσεις. Διάφορα ονόματα ήρθαν για να περιγράψουν την AI, όπως η πληροφορική, η μηχανική μάθηση και η υπολογιστική νοημοσύνη

1996



Εικόνα 2.7 Garry Kasparov lost his second match to a computer

<https://ichef.bbci.co.uk/images/ic/496xn/p0549xh4.jpg>

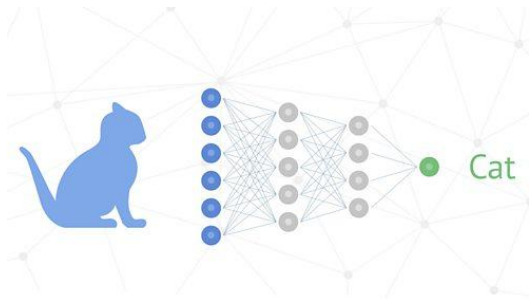
Το Deep Blue κερδίζει τον Garry Kasparov

Η ευαισθητοποίηση του κοινού για το AI αυξήθηκε σε μεγάλο βαθμό όταν ένας υπολογιστής IBM που ονομάζεται Deep Blue κέρδισε τον παγκόσμιο πρωταθλητή σκακιού Garry Kasparov στο πρώτο παιχνίδι ενός αγώνα.

Ο Kasparov κέρδισε το ματς του 1996, αλλά το 1997 ένα αναβαθμισμένο Deep Blue κέρδισε στη συνέχεια ένα δεύτερο αγώνα 3½ παιχνίδια σε 2½. Παρόλο που η Deep Blue έπαιξε ένα εντυπωσιακό παιχνίδι σκακιού βασίστηκε σε μεγάλο βαθμό στη βίαιη υπολογιστική δύναμη για να το πετύχει αυτό, συμπεριλαμβανομένων των 480 "σκακισμάτων" ειδικού σκοπού. Εργάστηκε με αναζήτηση από 6-20 κινήσεις μπροστά σε κάθε θέση, έχοντας μάθει με την αξιολόγηση χιλιάδων παλαιών παιχνιδιών σκακιού για να καθορίσει το μονοπάτι για το ματ.



2006



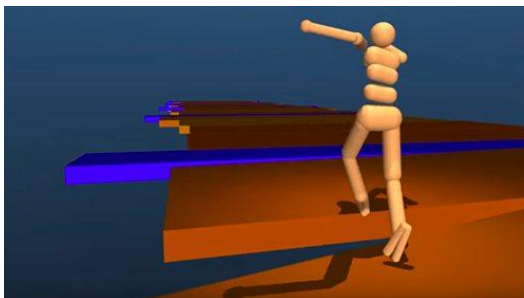
Εικόνα 2.8 Neural network image recognition via backpropagation  
<https://ichef.bbci.co.uk/images/ic/496xn/p067jsh4.jpg>

Η Επιστροφή!

Μία από τις βασικές τεχνικές που χρησιμοποιούνται στα συστήματα μηχανικής μάθησης είναι το backpropagation, που χρησιμοποιείται για την εκπαίδευση σε βαθιά νευρωνικά δίκτυα.

Πρώτα περιγράφεται στη δεκαετία του 1960 ως μέρος της θεωρίας ελέγχου και υιοθετήθηκε για νευρωνικά δίκτυα. Τα δίκτυα βαθιάς μάθησης είναι τώρα ένας βασικός άξονας της τρέχουσας εκμάθησης μηχανών. Το 2017, ο Hinton, ο οποίος τώρα εργάζεται για την Google, εξέφρασε ανησυχίες για το γεγονός ότι η backpropagation έχει φθάσει τα όριά της στην οικοδόμηση μηχανικών συστημάτων μάθησης και ότι απαιτούνται νέες ιδέες.

2014



Εικόνα 2.9 DeepMind artificial intelligence moving an animated figure  
<https://ichef.bbci.co.uk/images/ic/496xn/p058n9wr.jpg>

DeepMind

Το DeepMind απέκτησε μεγάλη σημασία όταν ανέπτυξε ένα νευρωνικό δίκτυο που θα μπορούσε να μάθει να παίζει βιντεοπαιχνίδια απλά αναλύοντας τη συμπεριφορά των εικονοστοιχείων σε μια οθόνη. Επίσης, δημιούργησε ένα νευρωνικό δίκτυο που μπορεί να έχει πρόσβαση στην εξωτερική μνήμη. Το 2016 το DeepMind

συμμετείχε σε μια διαμάχη σχετικά με τη χρήση δεδομένων ασθενών NHS από το Royal Free Hospital για την εκπαίδευση ενός ιατρικού συστήματος και το Γραφείο Επιτρόπων Πληροφοριών του Ηνωμένου Βασιλείου έκρινε ότι είχε παραβιάσει τον νόμο περί προστασίας προσωπικών δεδομένων.

2016



Εικόνα 2.10 Go was considered a difficult game  
<https://ichef.bbci.co.uk/images/ic/496xn/p067jsmj.jpg>

Το AlphaGo κερδίζει τον Lee Sedol

Το Go θεωρήθηκε ως ένα πιο δύσκολο παιχνίδι από το σκάκι για το AI να κυριαρχήσει

Αναπτύχθηκε από τους ερευνητές του DeepMind, ο AlphaGo κέρδισε τον πρώτο του αγώνα εναντίον ενός επαγγελματία το 2015, τον Μάρτιο του 2016 και τον αριθμό δύο παίκτη Lee Sedol τον Μάρτιο

του 2016 και τον αριθμό ένα παίκτη Ke Jie το 2017. Το νευρικό δίκτυο του AlphaGo χρησιμοποιεί έναν αλγόριθμο αναζήτησης του δέντρου Monte Carlo για να βρει κινήσεις. Η επιτυχία της είναι σημαντική καθώς οι ερευνητές του AI θεωρούν το παιχνίδι Go να είναι δύσκολο και δεν είχαν προβλέψει ότι οι άνθρωποι θα αρχίσουν να χάνουν από τους υπολογιστές τόσο σύντομα

2030s;



Εικόνα 2.11 BBC Click's Spencer Kelly and a humanoid robot  
<https://ichef.bbci.co.uk/images/ic/496xn/p061tw76.jpg>

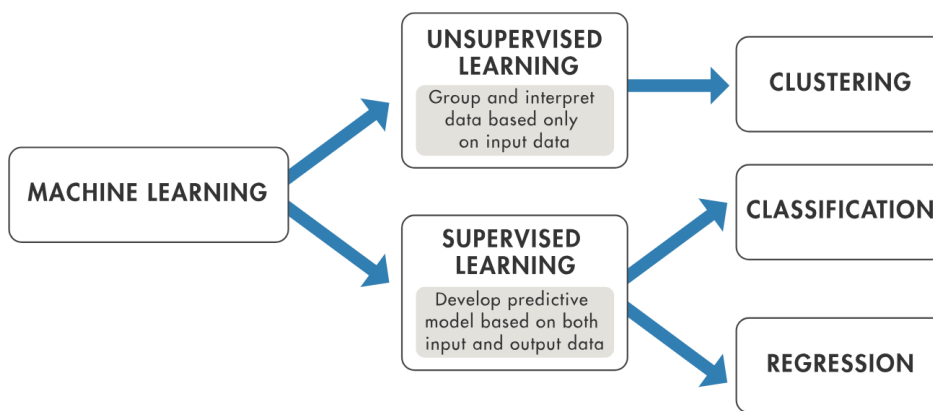
Μερικοί επιστήμονες υπολογιστών πιστεύουν ότι μόλις αναπτύξουμε ένα γενικευμένο AI σε επίπεδο ανθρώπινου επιπέδου ή πάνω από αυτό τότε θα αναπτύξει πιο προηγμένες εκδόσεις του.

Αυτή η διαδικασία ονομάζεται μοναδικότητα, ένας όρος που χρησιμοποιείται για πρώτη φορά κατ'αυτόν τον τρόπο από τον συγγραφέα SF Vernor Vinge. Εάν συμβεί αυτό, τότε η προκύπτουσα

εκθετική αύξηση της ικανότητας του AI θα ξεπεράσει γρήγορα την ανθρώπινη νοημοσύνη και θα βρεθούμε να είμαστε υπο των μηχανών. Η πιθανότητα της ιδιαιτερότητας μελετάται από οργανισμούς όπως το Κέντρο Κέμπριτζ για τη Μελέτη του Υπαρξιακού Κινδύνου, καθώς ακόμη και εάν θεωρείται πολύ απίθανο θα αποτελούσε σοβαρή απειλή για την ανθρώπινη επιβίωση.

### Διαδικασίες μάθησης:

Ακριβώς όπως υπάρχουν διαφορετικοί τρόποι με τους οποίους μαθαίνουν οι άνθρωποι από το περιβάλλον τους, το ίδιο ισχύει και σ αυτή την περίπτωση. Υπό μια ευρεία έννοια μπορούμε να κατηγοριοποιήσουμε τις διαδικασίες μάθησης μέσω των όποιων λειτουργιών τα ΝΔ ως εξής: μάθηση με εκπαιδευτή και μάθηση χωρίς εκπαιδευτή. Με το ίδιο κριτήριο η δεύτερη μορφή μπορεί να κατηγοριοποιηθεί περαιτέρω σε μη επιβλεπόμενη μάθηση και ενισχυτική μάθηση. Αυτές οι διαφορετικές μορφές μάθησης, όπως λαμβάνουν χώρα στα ΝΔ, προσομοιώνουν την ανθρώπινη μάθηση.



Εικόνα 2.12 Learning processes

[https://www.mathworks.com/help/stats/machinelearning\\_supervisedunsupervised.png](https://www.mathworks.com/help/stats/machinelearning_supervisedunsupervised.png)

### Μάθηση με Εκπαιδευτή

Η μάθηση με εκπαιδευτή αναφέρεται επίσης ως επιβλεπόμενη μάθηση. Με εννοιολογικούς όρους μπορούμε να θεωρήσουμε ότι ο εκπαιδευτής έχει γνώση του περιβάλλοντος και αυτή η γνώση αντιπροσωπεύεται από ένα σύνολο παραδειγμάτων εισόδου-εξόδου. Ωστόσο, το περιβάλλον είναι γνωστό στο νευρωνικό δίκτυο. Ας υποθέσουμε τώρα ότι ο εκπαιδευτής και το νευρωνικό δίκτυο εκτίθενται σ ένα διάνυσμα εκπαίδευσης (ένα παράδειγμα) αντλούμενο από το ίδιο περιβάλλον. Χάρη στην εγγενή του γνώση, ο εκπαιδευτής είναι σε θέση να παρέχει μια επιθυμητή απόκριση για το συγκεκριμένο διάνυσμα εκπαίδευσης.

Η επιθυμητή απόκριση αντιπροσωπεύει τη "βέλτιστη" ενέργεια που πρέπει να εκτελείται από το νευρωνικό δίκτυο. Οι παράμετροι δικτύου προσαρμόζονται υπό τη συνδυασμένη επιρροή του διανύσματος εκπαίδευσης και του σήματος σφάλματος. Το σήμα σφάλματος ορίζεται η διάφορα μεταξύ επιθυμητής και πραγματικής απόκρισης δικτύου. Αυτή η προσαρμογή

εκτελείται με επαναληπτικό τρόπο, βήμα προς βήμα, με στόχο να φέρει το ΝΔ σε μια κατάσταση όπου θα προσομοιώνει τη συμπεριφορά του εκπαιδευτή (η προσομοίωση κρίνεται ως βέλτιστη με κάποια στατιστική έννοια). Κατ' αυτό τον τρόπο, η γνώση του περιβάλλοντος που είναι διαθέσιμη στον εκπαιδευτή μεταφέρεται στο ΝΔ μέσω εκπαίδευσης και αποθηκεύεται με τη μορφή "σταθερών" συνοπτικών βαρών, τα όποια αντιπροσωπεύουν μακροπρόθεσμη μνήμη. Όταν επιτευχτεί αυτή η συνθήκη, μπορούμε να απαλλαγούμε από τον εκπαιδευτή και να αφήσουμε το νευρωνικό δίκτυο να αντιμετωπίσει το περιβάλλον εντελώς μόνο του.

### **Μάθηση χωρίς Εκπαιδευτή**

Στην επιβλεπόμενη μαθηση, η διαδικασία μαθησης λαμβάνει χώρα υπό την καθοδήγηση ενός εκπαιδευτή. Ωστόσο, στο παράδειγμα που είναι γνωστό ως μαθηση χωρίς εκπαιδευτή όπως δηλώνει το όνομα του, δεν υπάρχει εκπαιδευτής που να επιβλέπει την διαδικασία μαθησης. Δηλαδή, δεν υπάρχουν χαρακτηρισμένα παραδείγματα της λειτουργιάς που πρέπει να μάθει το δίκτυο. Στα πλαίσια αυτού του δευτέρου παραδείγματος, μπορούμε να ορίσουμε δυο υποκατηγορίες μαθησης:

### **Ενισχυτική Μάθηση:**

Στην ενισχυτική μαθηση, η εκμάθηση μιας αντιστοιχισής εισοδου-εξόδου εκτελείται μέσω συνεχούς αλληλεπίδρασης με το περιβάλλον, με στόχο την ελαχιστοποίηση ενός βαθμού δείκτη απόδοσης.

Ο στόχος της ενισχυτικής μαθησης είναι να ελαχιστοποιεί μια συνάρτηση τρέχοντος κόστους η όποια ορίζεται ως πρόβλεψη του αθροιστικού κόστους ενεργειών που εκτελούνται σε μια αλληλουχία βημάτων αντί απλώς του άμεσου κόστους μιας ενεργείας. Ενδεχομένως ορισμένες από τις ενέργειες που έχουν εκτελεστεί σαΐτα την αλληλουχία χρονικών βημάτων να είναι οι καλύτερες ορίζουσες της συνολικής συμπεριφοράς του συστήματος. Η λειτουργιά του συστήματος μαθησης είναι να ανακαλύψει αυτές τις ενέργειες και να τις τροφοδοτήσει πίσω στο περιβάλλον.

Η ενισχυτική μαθηση με καθυστέρηση είναι δύσκολο να εκτελεστεί για δυο βασικούς λόγους:

- Δεν υπάρχει εκπαιδευτής για να παρέχει μια επιθυμητή απόκριση σε κάθε βήμα της διαδικασίας μαθησης.
- Η καθυστέρηση με την όποια παράγεται το κύριο σήμα ενίσχυσης υποδηλώνει ότι η μανθάνουσα μηχανή πρέπει να λύσει το χρονικό πρόβλημα ανάθεσης εμπιστοσύνης. Μ' αυτό εννοούμε ότι η μηχανή πρέπει να είναι σε θέση να καθορίζει το βαθμό

επιτυχίας ατομικά για κάθε ενεργεία της χρονικής αλληλουχίας βημάτων που οδήγησαν στο τελικό αποτέλεσμα, ενώ ο κυρίως μηχανισμός ενίσχυσης μπορεί να αποτιμά μόνο το τελικό αποτέλεσμα.

### Μη Επιβλεπόμενη Μάθηση

Στη μη επιβλεπόμενη μαθηση δεν υπάρχει εξωτερικός εκπαιδευτής η κριτής που να επιβλέπει τη διαδικασία μαθησης. Αντ' αυτού, υπάρχει ένα ανεξάρτητο από την εργασία μετρό της ποιότητας της αναπαράστασης που καλείται να μάθει το δίκτυο και οι ελεύθερες παράμετροι του δικτύου βελτιστοποιούνται σε σχέση με αυτό το μετρό. Για ένα συγκεκριμένο ανεξάρτητο από την εργασία μετρό, αφού το δίκτυο "συντονιστεί" στις στατιστικές κανονικότητες των δεδομένων εισόδου αναπτύσσει τη δυνατότητα να σχηματίζει εσωτερικές αναπαραστάσεις για την κωδικοποίηση χαρακτηριστικών της εισόδου και, μέσω αυτών, να δημιουργεί νέες κλάσεις αυτόματα (Becker 1991).

Για την εκτέλεση μπορούμε να χρησιμοποιήσουμε έναν κανόνα ανταγωνιστικής μαθησης. Για παράδειγμα, μπορούμε να χρησιμοποιήσουμε ένα νευρωνικό δίκτυο το οποίο αποτελείται από δυο επίπεδα - ένα επίπεδο εισόδου και ένα ανταγωνιστικό επίπεδο. Το επίπεδο εισόδου λαμβάνει τα διαθέσιμα δεδομένα. Το ανταγωνιστικό επίπεδο αποτελείται από νευρώνες οι οποίοι ανταγωνίζονται ο ένας τον άλλο (σύμφωνα με έναν κανόνα μαθησης) για την "ευκαιρία" να αποκριθούν σε χαρακτηριστικά που περιέχονται στα δεδομένα εισόδου. Στην απλούστερη δυνατή μορφή του, το δίκτυο λειτουργεί σύμφωνα με μια στρατηγική "ο νικητής παίρνει τα πάντα". Βάσει μιας τέτοιας στρατηγικής, ο νευρώνας με την μεγαλύτερη συνολική είσοδο "νικά" στον ανταγωνισμό και ενεργοποιείται. Όλοι οι άλλοι νευρώνες του δικτύου απενεργοποιούνται.

Μια άλλη κατηγοριοποίηση των προβλημάτων μηχανικής μάθησης προκύπτει όταν κάποιος θεωρήσει το επιθυμητό αποτέλεσμα του συστήματος μηχανικής μάθησης.

- Στην ταξινόμηση, τα δεδομένα εισόδου χωρίζονται σε δύο ή περισσότερες κλάσεις, και η μηχανή πρέπει να κατασκευάσει ένα μοντέλο, το οποίο θα αντιστοιχίζει τα δεδομένα σε μία ή περισσότερες (multi-label ταξινόμηση) κλάσεις. Αυτό συνήθως εμπίπτει στην επιτηρούμενη μάθηση. Τα φίλτρα Spam είναι ένα παράδειγμα ταξινόμησης, όπου οι εισοδοί είναι τα emails ή άλλα μηνύματα και οι κλάσεις είναι "spam" και "όχι spam".

- Στην παλινδρόμηση, επίσης πρόβλημα επιτηρούμενης μάθησης, τα αποτελέσματα είναι συνεχή και όχι διακριτά.
- Στην συσταδοποίηση, ένα σύνολο εισόδων πρόκειται να χωριστεί σε ομάδες. Σε αντίθεση με την ταξινόμηση, οι ομάδες δεν είναι γνωστές εκ των προτέρων, καθιστώντας αυτόν τον διαχωρισμό τυπική εργασία μη επιτηρούμενης μάθησης.
- Στην εκτίμηση πυκνότητας βρίσκει την κατανομή των δεδομένων εισόδου σε κάποιο χώρο.
- Σε προβλήματα μείωσης διαστασιμότητας (dimensionality reduction), τα δεδομένα απλοποιούνται και αντιστοιχίζονται σε ένα χώρο λιγότερων διαστάσεων. Το στατιστικό μοντέλο θεμάτων (Topic modeling) είναι ένα σχετικό πρόβλημα, όπου η μηχανή καλείται να βρει έγγραφα που καλύπτουν παρόμοια θέματα από ένα σύνολο εγγράφων γραμμένων σε φυσική γλώσσα.

#### **Ορισμένες εφαρμογές της Μηχανικής Μάθησης:**

- Αναγνώριση ομιλίας και γραφικού χαρακτήρα
- Βιοπληροφορική
- Επεξεργασία φυσικής γλώσσας
- Ηλεκτρονικά παιχνίδια
- Ιατρική Διάγνωση
- Μετακίνηση Ρομπότ
- Μηχανές αναζήτησης
- Μηχανική αντίληψη
- Οικονομία
- Συναισθηματική υπολογιστική
- Υπολογιστική όραση- συμπεριλαμβανομένης της αναγνώρισης αντικειμένου
- Χρηματιστηριακή ανάλυση

## Αλγόριθμοι Μηχανικής Μάθησης

Στη μηχανική μάθηση, υπάρχει το θεώρημα ότι κανένας αλγόριθμος δεν λειτουργεί καλύτερα για κάθε πρόβλημα και είναι ιδιαίτερα σημαντικό για την εποπτευόμενη μάθηση (δηλαδή πρόβλεψη μοντελοποίησης). Για παράδειγμα, δεν μπορούμε να πούμε ότι τα νευρικά δίκτυα είναι πάντα καλύτερα από τα δέντρα απόφασης ή το αντίστροφο.

Υπάρχουν πολλοί παράγοντες που παίζουν ρόλο σ' αυτό, όπως το μέγεθος και η δομή του συνόλου των δεδομένων μας. Ως αποτέλεσμα, θα πρέπει να δοκιμάσουμε πολλούς διαφορετικούς αλγόριθμους για το πρόβλημά μας, ενώ χρησιμοποιούμε ένα "δοκιμαστικό σύνολο" δεδομένων για την αξιολόγηση της απόδοσης και την επιλογή του νικητή.

Φυσικά, οι αλγόριθμοι που δοκιμάζουμε πρέπει να είναι κατάλληλοι για το πρόβλημά μας, όπου έρχεται η επιλογή της σωστής εργασίας εκμάθησης μηχανής.

Ωστόσο, υπάρχει μια κοινή αρχή που βασίζεται σε όλους τους εποπτευόμενους αλγορίθμους μηχανικής μάθησης για πρόβλεψη μοντελοποίησης.

Οι αλγόριθμοι μάθησης μηχανών περιγράφονται ως εκμάθηση μιας συνάρτησης στόχου ( $f$ ) που καλεί καλύτερα τις μεταβλητές εισόδου ( $X$ ) σε μια μεταβλητή εξόδου ( $Y$ ):  $Y = f(X)$

Αυτό είναι ένα γενικό μάθημα εκμάθησης στο οποίο θα θέλαμε να κάνουμε προβλέψεις στο μέλλον ( $Y$ ) δίνοντας νέα παραδείγματα μεταβλητών εισόδου ( $X$ ). Δεν γνωρίζουμε ποια είναι η λειτουργία ( $f$ ) ή η μορφή της. Εάν το κάναμε, θα το χρησιμοποιήσαμε άμεσα και δεν θα χρειαζόταν να το μάθουμε από δεδομένα χρησιμοποιώντας αλγορίθμους μηχανικής μάθησης.

Ο πιο συνηθισμένος τύπος μηχανικής μάθησης είναι να μάθουμε τη χαρτογράφηση  $Y = f(X)$  για να κάνουμε προβλέψεις για το  $Y$  για το νέο  $X$ . Αυτό ονομάζεται πρόβλεψη μοντελοποίησης ή προγνωστικών αναλυτικών στοιχείων και ο στόχος μας είναι να κάνουμε τις πιο ακριβείς δυνατές προβλέψεις.

Η εποπτευόμενη μάθηση περιλαμβάνει δύο κατηγορίες αλγορίθμων:

**Ταξινόμηση:** για κατηγορίες τιμών απόκρισης, όπου τα δεδομένα μπορούν να διαχωριστούν σε συγκεκριμένες "κλάσεις"

**Παλινδρόμηση:** για τιμές συνεχούς απόκρισης



Οι κοινοί αλγόριθμοι **ταξινόμησης** περιλαμβάνουν

- Μηχανές διανυσμάτων υποστήριξης (Support vector machines-SVM)
- Νευρωνικά δίκτυα (Neural networks)
- Naïve Bayes ταξινομητής
- Δέντρα αποφάσεων(Decision trees)
- Πλησιέστεροι γείτονες (Nearest neighbors -kNN)

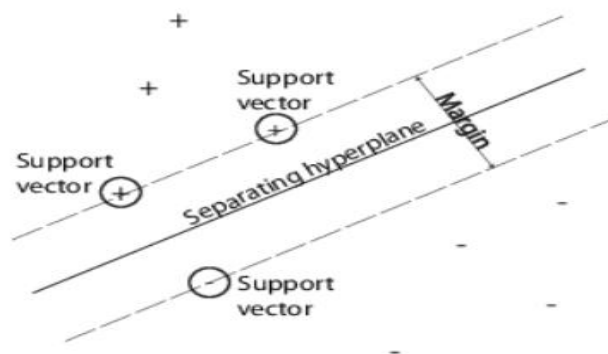
Οι συνήθεις αλγόριθμοι **παλινδρόμησης** περιλαμβάνουν:

- Γραμμικής παλινδρόμησης(Linear regression)
- Μη γραμμική παλινδρόμηση(Nonlinear regression)
- Δέντρα αποφάσεων(Decision trees)
- Νευρωνικά δίκτυα(Neural networks)

### **Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines)**

Μπορούμε να χρησιμοποιήσουμε μια μηχανή διανυσμάτων υποστήριξης (SVM) όταν τα δεδομένα μας έχουν ακριβώς δύο κατηγορίες. Ένα SVM ταξινομεί τα δεδομένα με την εύρεση του καλύτερου υπερεπιπέδου (hyperplane) που διαχωρίζει όλα τα σημεία δεδομένων μιας τάξης από εκείνα της άλλης τάξης. Το καλύτερο υπερεπιπέδο (hyperplane) για ένα SVM είναι αυτό με το μεγαλύτερο περιθώριο μεταξύ των δύο τάξεων. Το περιθώριο υπολογίζεται ως η κάθετη απόσταση από την γραμμή προς τα κοντινότερα σημεία.

Τα διανύσματα υποστήριξης είναι τα σημεία δεδομένων που βρίσκονται πλησιέστερα στο διαχωριστικό υπερεπιπέδο (hyperplane), αυτά τα σημεία βρίσκονται στο όριο της πλάκας. Το παρακάτω σχήμα απεικονίζει αυτούς τους ορισμούς, με "+" υποδεικνύοντας σημεία δεδομένων τύπου 1, και "-" υποδεικνύοντας σημεία δεδομένων τύπου -1.



Εικόνα 2.14 Διαχωρισμός σημείων ενδιαφέροντος  
<https://www.mathworks.com/help/stats/svmhyperplane.png>

Το υπερεπίπεδο (hyperplane) εκπαιδεύεται από τη Μηχανή Διανυσμάτων Υποστήριξης (SVM) μέσα από ένα σύνολο δεδομένων εισόδου (training set) χρησιμοποιώντας μια διαδικασία βελτιστοποίησης που μεγιστοποιεί το περιθώριο.

Στην πράξη όμως τα πραγματικά δεδομένα είναι ανακατεμένα και δεν μπορούν να χωριστούν τέλεια με ένα υπερεπίπεδο. Ο περιορισμός της μεγιστοποίησης του περιθωρίου της γραμμής που διαχωρίζει τις κλάσεις πρέπει να είναι λιγότερο αυστηρός. Αυτό αποκαλείται Soft Margin Classifier. Αυτή η αλλαγή επιτρέπει σε κάποια σημεία του συνόλου δεδομένων εκπαίδευσης (training set) να παραβιάσουν την διαχωριστική γραμμή.

### Παράμετροι συντονισμού:

Πυρήνας (Kernel): Η εκμάθηση του υπερεπιπέδου σε γραμμικό SVM γίνεται με το μετασχηματισμό του προβλήματος χρησιμοποιώντας κάποια γραμμική άλγεβρα. Εδώ είναι όπου ο πυρήνας παίζει ρόλο.

Για τον γραμμικό πυρήνα (linear kernel) η εξίσωση για την πρόβλεψη για μια νέα είσοδο χρησιμοποιώντας το dot προϊόν μεταξύ της εισόδου (x) και κάθε διάνυσμα υποστήριξης (xi) υπολογίζεται ως εξής:

$$f(x) = B(0) + \sum(a_i * (x, x_i))$$

Αυτή είναι μια εξίσωση που περιλαμβάνει τον υπολογισμό των εσωτερικών προϊόντων ενός νέου διανύσματος εισόδου (x) με όλους τους φορείς υποστήριξης στα δεδομένα εκπαίδευσης.

Οι συντελεστές B0 και ai (για κάθε είσοδο) πρέπει να εκτιμηθούν από τα δεδομένα εκπαίδευσης με τον αλγόριθμο μάθησης.

Ο πολυωνυμικός πυρήνας (polynomial kernel) μπορεί να γραφεί ως

$$K(x, x_i) = 1 + \text{αθροιστής}(x * x_i)^d$$

και εκθετικός ως: [Πηγή: <http://machinelearningmastery.com/>]

$$K(x, x_i) = \exp(-\gamma * \sum((x - x_i)^2))$$

Το πολυώνυμο και οι εκθετικοί πυρήνες υπολογίζουν τη γραμμή διαχωρισμού σε υψηλότερες διαστάσεις. Αυτό ονομάζεται κόλπο πυρήνα(kernel trick)

**Τακτοποίηση (Regularization):** Η παράμετρος Regularization (που συχνά ονομάζεται παράμετρος C) αναφέρει στη βελτιστοποίηση SVM πόσο θέλετε να αποφύγετε την εσφαλμένη ταξινόμηση κάθε παραδείγματος εκπαίδευσης.

Για τις μεγάλες τιμές του C, η βελτιστοποίηση θα επιλέξει ένα υπερεπίπεδο μικρότερου περιθωρίου αν αυτό το υπερεπίπεδο κάνει καλύτερη δουλειά παίρνοντας όλα τα σημεία κατάρτισης ταξινομημένα σωστά.

Αντίθετα, μια πολύ μικρή τιμή του C θα αναγκάσει το βελτιστοποιητή να αναζητήσει ένα μεγαλύτερο περιθώριο διαχωρισμού του υπερπληθυσμού, ακόμα κι αν αυτό το υπερεπίπεδο ταξινομεί εσφαλμένα περισσότερα σημεία.

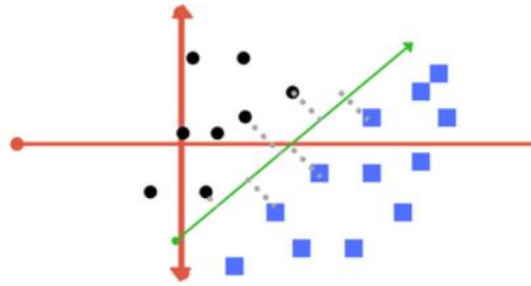
Οι παρακάτω εικόνες είναι παραδείγματα δύο διαφορετικών παραμέτρων κανονικοποίησης. Αριστερά υπάρχει κάποια εσφαλμένη ταξινόμηση λόγω της χαμηλότερης τιμής ρύθμισης. Η υψηλότερη τιμή οδηγεί σε αποτελέσματα όπως η δεξιά.



Εικόνα 2.15: Αριστερά: Χαμηλή τιμή κανονικοποίησης/ Δεξιά: Υψηλή τιμή κανονικοποίησης

[https://miro.medium.com/max/625/1\\*gt\\_dkcA5p0ZTHjIpq1qnLQ.png](https://miro.medium.com/max/625/1*gt_dkcA5p0ZTHjIpq1qnLQ.png)

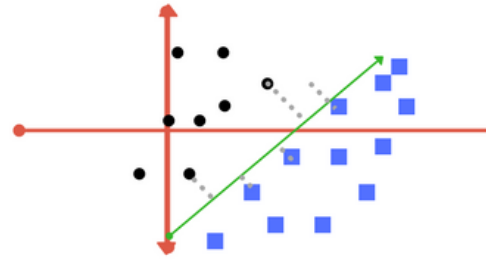
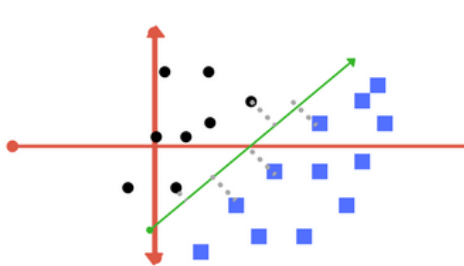
**Gamma:** Η παράμετρος gamma καθορίζει πόσο μακριά φτάνει η επιρροή ενός μόνο εκπαιδευτικού παραδείγματος, με χαμηλές τιμές που σημαίνουν «μακριά» και υψηλές τιμές που σημαίνουν «κλειστό». Με άλλα λόγια, με χαμηλή gamma, τα σημεία απομακρύνονται από την εύλογη γραμμή διαχωρισμού, περιλαμβάνονται στον υπολογισμό για τη γραμμή διαχωρισμού. Όπου "υψηλό" gamma σημαίνει ότι τα σημεία κοντά στην εύλογη γραμμή θεωρούνται στον υπολογισμό.



Εικόνα 2.16 Υψηλό Gamma

[https://miro.medium.com/max/750/1\\*dGDQxV8j83VB90skHsXktw.png](https://miro.medium.com/max/750/1*dGDQxV8j83VB90skHsXktw.png)

**Περιθώριο(Margin):** Ένα περιθώριο είναι ένας διαχωρισμός της γραμμής στα πλησιέστερα σημεία της κλάσης. Ένα καλό περιθώριο είναι εκείνο όπου ο διαχωρισμός αυτός είναι μεγαλύτερος και για τις δύο τάξεις. Οι εικόνες παρακάτω δίνουν σε οπτικό παράδειγμα καλό και κακό περιθώριο. Ένα καλό περιθώριο επιτρέπει στα σημεία να είναι στις αντίστοιχες τάξεις τους χωρίς να διασχίζουν σε άλλη κατηγορία.



Εικόνα 2.17 Κακό περιθώριο (πολύ κοντά στη μια περιοχή)

Καλό περιθώριο (ίσες αποστάσεις)

[https://miro.medium.com/max/750/1\\*Ftns0ebfUHJDdpWt3Wvp-Q.png](https://miro.medium.com/max/750/1*Ftns0ebfUHJDdpWt3Wvp-Q.png)

### Νευρωνικά Δίκτυα (Neural Networks - NN)

Το νευρωνικό δίκτυο είναι ένα δίκτυο από απλούς υπολογιστικούς κόμβους (νευρώνες, νευρώνια), διασυνδεδεμένους μεταξύ τους. Είναι εμπνευσμένο από το Κεντρικό Νευρικό Σύστημα (ΚΝΣ), το οποίο προσπαθεί να προσομοιώσει.

Οι *νευρώνες* είναι τα δομικά στοιχεία του δικτύου. Κάθε τέτοιος κόμβος δέχεται ένα σύνολο αριθμητικών εισόδων από διαφορετικές πηγές (είτε από άλλους νευρώνες, είτε από το περιβάλλον), επιτελεί έναν υπολογισμό με βάση αυτές τις εισόδους και παράγει μία έξοδο. Η

εν λόγω έξοδος είτε κατευθύνεται στο περιβάλλον, είτε τροφοδοτείται ως είσοδος σε άλλους νευρώνες του δικτύου. Υπάρχουν τρεις τύποι νευρώνων: οι *νευρώνες εισόδου*, οι *νευρώνες εξόδου* και οι *υπολογιστικοί νευρώνες* ή *κρυμμένοι νευρώνες*.

Οι νευρώνες εισόδου δεν επιτελούν κανέναν υπολογισμό, μεσολαβούν απλώς ανάμεσα στις περιβαλλοντικές εισόδους του δικτύου και στους υπολογιστικούς νευρώνες.

Οι νευρώνες εξόδου διοχετεύουν στο περιβάλλον τις τελικές αριθμητικές εξόδους του δικτύου.

Οι υπολογιστικοί νευρώνες πολλαπλασιάζουν κάθε είσοδό τους με το αντίστοιχο *συναπτικό βάρος* και υπολογίζουν το ολικό άθροισμα των γινομένων. Το άθροισμα αυτό τροφοδοτείται ως όρισμα στη *συνάρτηση ενεργοποίησης*, την οποία υλοποιεί εσωτερικά κάθε κόμβος. Η τιμή που λαμβάνει η συνάρτηση για το εν λόγω όρισμα είναι και η έξοδος του νευρώνα για τις τρέχουσες εισόδους και βάρη.

Εάν  $x_{ki}$  είναι η  $i$ -οστή είσοδος του  $K$  νευρώνα,  $w_{ki}$ : το  $i$ -οστό συναπτικό βάρος του  $K$  νευρώνα και  $\Phi(\cdot)$  η συνάρτηση ενεργοποίησης του νευρωνικού δικτύου, τότε η έξοδος  $Y_k$  του  $K$  νευρώνα δίνεται από την εξίσωση:

$$Y_k = \Phi(\sum_{i=0}^N x_{ki} * w_{ki})$$

Στον  $K$ -οστό νευρώνα υπάρχει ένα συναπτικό βάρος  $w_{k0}$  με ιδιαίτερη σημασία, το οποίο καλείται *πόλωση* ή *κατώφλι* (*bias*, *threshold*). Η τιμή της εισόδου του είναι πάντα η μονάδα,  $x_{k0} = 1$ . Εάν το συνολικό άθροισμα από τις υπόλοιπες εισόδους του νευρώνα είναι μεγαλύτερο από την τιμή αυτή, τότε ο νευρώνας ενεργοποιείται. Εάν είναι μικρότερο, τότε ο νευρώνας παραμένει ανενεργός. Η ιδέα προέκυψε από τα βιολογικά νευρικά κύτταρα.

Το κύριο χαρακτηριστικό των νευρωνικών δικτύων είναι η εγγενής ικανότητα μάθησης. Ως μάθηση μπορεί να οριστεί η σταδιακή βελτίωση της ικανότητας του δικτύου να επιλύει κάποιο πρόβλημα (π.χ. η σταδιακή προσέγγιση μίας συνάρτησης). Η μάθηση επιτυγχάνεται μέσω της εκπαίδευσης, μίας επαναληπτικής διαδικασίας σταδιακής προσαρμογής των παραμέτρων του δικτύου (συνήθως των βαρών και της πόλωσής του) σε τιμές κατάλληλες ώστε να επιλύεται με επαρκή επιτυχία το προς εξέταση πρόβλημα. Αφού ένα δίκτυο εκπαιδευτεί, οι παράμετροί του συνήθως «παγώνουν» στις κατάλληλες τιμές και από εκεί κι έπειτα είναι σε λειτουργική κατάσταση. Το ζητούμενο είναι το λειτουργικό δίκτυο να χαρακτηρίζεται από μία ικανότητα γενίκευσης: αυτό σημαίνει πως δίνει ορθές εξόδους για εισόδους καινοφανείς και διαφορετικές από αυτές με τις οποίες εκπαιδεύτηκε.

### Συναρτήσεις ενεργοποίησης:

Η συνάρτηση ενεργοποίησης μπορεί να είναι βηματική (step transfer function), γραμμική (linear transfer function), μη γραμμική (non-linear transfer function), στοχαστική (stochastic transfer function).

### **Βηματική συνάρτηση ενεργοποίησης:**

Η βηματική συνάρτηση ενεργοποίησης μπορεί να είναι:

$$\Phi(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

ή οποιαδήποτε άλλη βηματική συνάρτηση.

Η βηματική συνάρτηση δεν θεωρείται χρήσιμη ως συνάρτηση ενεργοποίησης στα ΤΝΔ (Τεχνητά Νευρωνικά Δίκτυα), καθώς σύμφωνα με τον απειροστικό λογισμό έχει το βασικό μειονέκτημα ότι η παράγωγός της απειρίζεται. Έτσι προέκυψε η ανάγκη συναρτήσεων ενεργοποίησης που η γραφική τους παράσταση τους να μοιάζει με τη βηματική, αλλά ταυτόχρονα να είναι συνεχείς και παραγωγίσιμες σε όλο το πεδίο ορισμού τους.

Τέτοια συνάρτηση είναι και η σιγμοειδής.

### **Γραμμική συνάρτηση ενεργοποίησης:**

Η γραμμική συνάρτηση ενεργοποίησης μπορεί να είναι:

$$\Phi(x) = x$$

ή οποιαδήποτε άλλη γραμμική συνάρτηση.

### **Μη γραμμική συνάρτηση ενεργοποίησης:**

Η μη γραμμική συνάρτηση ενεργοποίησης που χρησιμοποιείται συνήθως στα νευρωνικά δίκτυα καλείται σιγμοειδής συνάρτηση. Οι τυπικές σιγμοειδείς είναι δύο:

Λογιστική σιγμοειδής:

$$\Phi(x) = \frac{1}{1+e^{-x}}$$

Υπερβολική εφραπτομένη:

$$\Phi(x) = \tanh(x)$$

### Εφαρμογές

Τα νευρωνικά δίκτυα είναι εφαρμόσιμα σχεδόν σε κάθε κατάσταση στην οποία ισχύει μια σχέση μεταξύ μεταβλητών πρόβλεψης (ανεξάρτητες, εισροές) και προβλεπόμενες μεταβλητές (εξαρτημένες, εκροές), ακόμα και όταν αυτή η σχέση είναι πολύ περίπλοκη για να αποδοθεί με τους συνηθισμένους όρους της «συσχέτισης» ή των «διαφόρων ομάδων». Ενδεικτικά αντιπροσωπευτικά παραδείγματα προβλημάτων στα οποία η ανάλυση των νευρωνικών δικτύων έχει εφαρμοστεί με επιτυχία είναι τα εξής:

- Ιατρική διάγνωση
- Χρηματιστηριακές προβλέψεις
- Πιστωτική ανάθεση
- Παρακολούθηση της κατάστασης των μηχανημάτων
- Συστήματα διαχείρισης κινητήρα

### Naive Bayes

Ο Naive Bayesian ταξινομητής βασίζεται στο θεώρημα του Bayes με τις υποθέσεις ανεξαρτησίας μεταξύ των προγνωστικών. Ένα Naive Bayesian μοντέλο είναι εύκολο να οικοδομηθεί, χωρίς περίπλοκη επαναληπτική παραμετροποίηση που καθιστά ιδιαίτερα χρήσιμη για πολύ μεγάλα σύνολα δεδομένων. Χρησιμοποιείται ευρέως επειδή συχνά υπεραποδίδει πιο εξελιγμένες μεθόδους ταξινόμησης.

### Ο Αλγόριθμος:

Το θεώρημα Bayes παρέχει έναν τρόπο υπολογισμού της οπίσθιας πιθανότητας  $P(c | x)$ , από  $P(c)$ ,  $P(x)$  και  $P(x | c)$ . Ο ταξινομητής Naive Bayes υποθέτει ότι η επίδραση της τιμής ενός προγνώστη ( $x$ ) σε μια δεδομένη κλάση ( $c$ ) είναι ανεξάρτητη από τις τιμές άλλων προγνωστικών.

$$P(c | x) = \frac{P(x | c)P(c)}{P(x)}$$

Likelihood
Class Prior Probability  
↓
↓  
Posterior Probability
Predictor Prior Probability

$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$$

Εικόνα 2.18 Οπίσθια πιθανότητα

[https://www.saedsayad.com/images/Bayes\\_rule.png](https://www.saedsayad.com/images/Bayes_rule.png)



- $P(c | x)$  είναι η οπίσθια πιθανότητα της τάξης (στόχου) που δίνεται πρόβλεψη (χαρακτηριστικό).
- $P(c)$  είναι η προηγούμενη πιθανότητα της κλάσης.
- $P(x | c)$  είναι η πιθανότητα που είναι η πιθανότητα της προβλεπόμενης κατηγορίας προγνωστικού.
- $P(x)$  είναι η προηγούμενη πιθανότητα του προγνωστικού.

Στο μοντέλο ZeroR δεν υπάρχει κανένας προγνωστικός παράγοντας, στο μοντέλο OneR προσπαθούμε να βρούμε τον ενιαίο καλύτερο προγνωστικό παράγοντα, ο Naive Bayesian περιλαμβάνει όλους τους παράγοντες πρόβλεψης χρησιμοποιώντας τη διακυβέρνηση του Bayes και τις υποθέσεις ανεξαρτησίας μεταξύ των προγνωστικών.

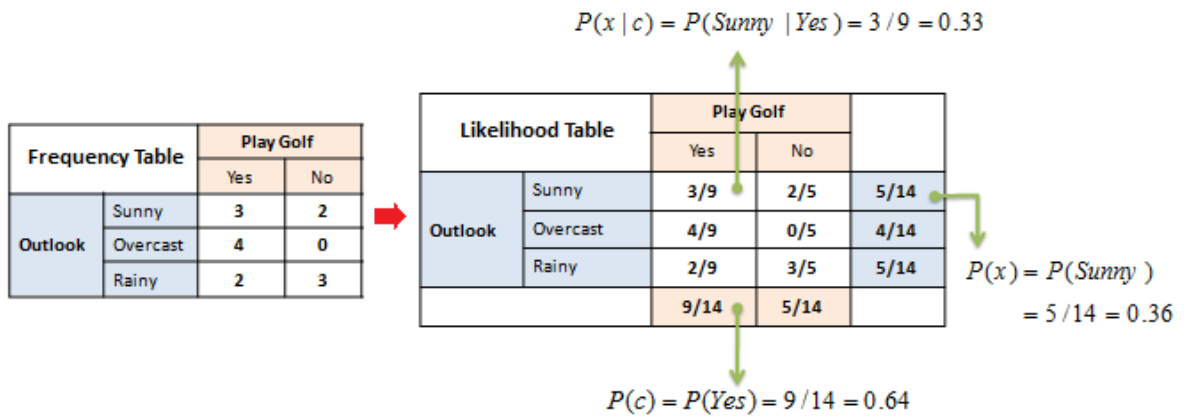
Παράδειγμα καιρού:

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

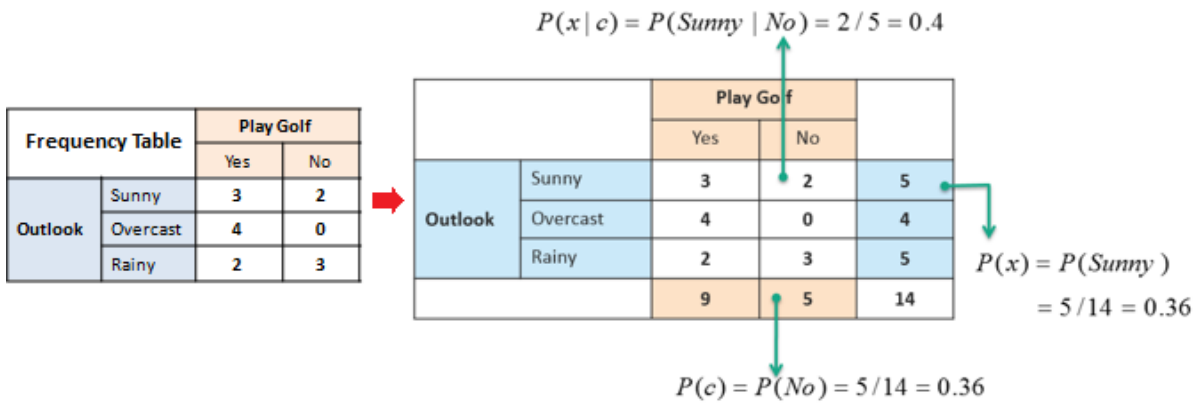
Εικόνα 2.19 Παράδειγμα πιθανότητας καιρού

[https://www.saedsayad.com/images/naive\\_bayes\\_data.png](https://www.saedsayad.com/images/naive_bayes_data.png)

Η οπίσθια πιθανότητα μπορεί να υπολογιστεί πρώτα, κατασκευάζοντας έναν πίνακα συχνότητας για κάθε χαρακτηριστικό έναντι του στόχου. Στη συνέχεια, μετασχηματίζοντας τους πίνακες συχνότητας σε πινάκες πιθανοτήτων και τέλος χρησιμοποιήστε την Naive Bayesian εξίσωση για να υπολογίσετε την οπίσθια πιθανότητα για κάθε κατηγορία. Η τάξη με την υψηλότερη posterior πιθανότητα είναι το αποτέλεσμα της πρόβλεψης.



Posterior Probability:  $P(c | x) = P(\text{Yes} | \text{Sunny}) = 0.33 \times 0.64 \div 0.36 = 0.60$



Posterior Probability:  $P(c | x) = P(\text{No} | \text{Sunny}) = 0.40 \times 0.36 \div 0.36 = 0.40$

Εικόνα 2.20 Αποτέλεσμα πιθανότητας

[https://www.saedsayad.com/images/Bayes\\_3.png](https://www.saedsayad.com/images/Bayes_3.png)

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf	
		Yes	No
Outlook	Sunny	3/9	2/5
	Overcast	4/9	0/5
	Rainy	2/9	3/5

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Humidity	High	3/9	4/5
	Normal	6/9	1/5

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Temp.	Hot	2/9	2/5
	Mild	4/9	2/5
	Cool	3/9	1/5

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

		Play Golf	
		Yes	No
Windy	False	6/9	2/5
	True	3/9	3/5

Εικόνα 2.21 Πίνακες συχνότητας - πιθανότητας

[https://www.saedsayad.com/images/naive\\_bayes\\_likelihood.png](https://www.saedsayad.com/images/naive_bayes_likelihood.png)

### Αριθμητικοί Προγνώστες (Numerical Predictors)

Οι αριθμητικές μεταβλητές πρέπει να μετατραπούν σε κατηγορίες (binning) πριν δημιουργήσουν τους πίνακες συχνότητάς τους. Η άλλη επιλογή που χρησιμοποιούμε είναι να χρησιμοποιήσουμε τη διανομή της αριθμητικής μεταβλητής για να έχουμε μια καλή εικόνα για τη συχνότητα. Για παράδειγμα, μια κοινή πρακτική είναι να υποθέσουμε κανονικές κατανομές για αριθμητικές μεταβλητές.

Η συνάρτηση πιθανότητας για την κανονική κατανομή καθορίζεται από δύο παραμέτρους (μέση και τυπική απόκλιση).

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad \text{Mean}$$

$$\sigma = \left[ \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)^2 \right]^{0.5} \quad \text{Standard deviation}$$

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad \text{Normal distribution}$$

Εικόνα 2.22 Παράμετροι της πιθανότητας

[https://www.saedsayad.com/images/Bayes\\_NormDist.png](https://www.saedsayad.com/images/Bayes_NormDist.png)

### Δέντρα αποφάσεων (Decision Trees)

Το δέντρο απόφασης δημιουργεί μοντέλα ταξινόμησης ή παλινδρόμησης με τη μορφή δομής δέντρου. Καταλύει ένα σύνολο δεδομένων σε μικρότερα και μικρότερα υποσύνολα, ενώ παράλληλα αναπτύσσεται προοδευτικά ένα συσχετισμένο δέντρο αποφάσεων. Το τελικό αποτέλεσμα είναι ένα δέντρο με κόμβους απόφασης και κόμβους φύλλων. Ένας κόμβος απόφασης (π.χ. Outlook) έχει δύο ή περισσότερους κλάδους (π.χ. Sunny, Clouds and Rainy). Ο κόμβος φύλλων (π.χ. Play) αντιπροσωπεύει μια ταξινόμηση ή μια απόφαση. Ο κορυφαίος κόμβος απόφασης σε ένα δέντρο που αντιστοιχεί στον καλύτερο προγνωστικό παράγοντα που ονομάζεται κόμβος ρίζας. Τα δέντρα αποφάσεων μπορούν να χειριστούν τόσο τα κατηγορηματικά όσο και τα αριθμητικά δεδομένα.



Εικόνα 2.23 Προβλέψεις του καιρού μέσω δέντρου απόφασης

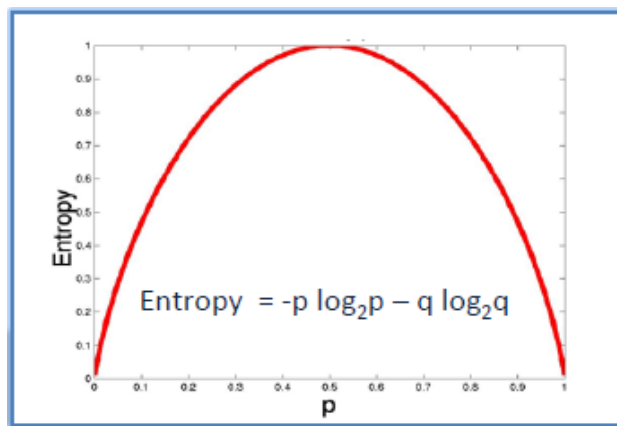
[https://www.saedsayad.com/images/Decision\\_Tree\\_1.png](https://www.saedsayad.com/images/Decision_Tree_1.png)

### Αλγόριθμος:

Ο πυρήνας του αλγόριθμου για τη δόμηση των δέντρων αποφάσεων ονομάζεται ID3 από τον J. R. Quinlan ο οποίος χρησιμοποιεί μια άκρη-προς τα κάτω, άπληστη αναζήτηση μέσα από το χώρο των πιθανών κλάδων χωρίς καμία ανασχεση. Το ID3 χρησιμοποιεί την Εντροπία και την Πληροφόρηση για την κατασκευή ενός δέντρου αποφάσεων. Στο μοντέλο ZeroR δεν υπάρχει κανένας προγνωστικός παράγοντας, στο μοντέλο OneR προσπαθούμε να βρούμε τον ενιαίο καλύτερο προγνωστικό παράγοντα, ο Naive Bayesian περιλαμβάνει όλους τους προγνωστικούς παράγοντες χρησιμοποιώντας τη διακυβέρνηση του Bayes και τις υποθέσεις ανεξαρτησίας μεταξύ των προγνωστικών, αλλά το δέντρο απόφασης περιλαμβάνει όλους τους προγνωστικούς δείκτες με τις υποθέσεις εξάρτησης μεταξύ των προγνωστικών.

### Εντροπία

Ένα δέντρο απόφασης είναι χτισμένο από την κορυφή προς τα κάτω από έναν κόμβο ρίζας και περιλαμβάνει τη διαίρεση των δεδομένων σε υποσύνολα που περιέχουν περιπτώσεις με παρόμοιες τιμές (ομοιογενείς). Ο αλγόριθμος ID3 χρησιμοποιεί την εντροπία για τον υπολογισμό της ομοιογένειας ενός δείγματος. Αν το δείγμα είναι εντελώς ομοιογενές, η εντροπία είναι μηδέν και αν το δείγμα είναι εξίσου διαιρεμένο έχει εντροπία ενός.



$$\text{Entropy} = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$$

Εικόνα 2.24 Εντροπία

<https://www.saedsayad.com/images/Entropy.png>

Για να χτίσουμε ένα δέντρο αποφάσεων, πρέπει να υπολογίσουμε δύο τύπους εντροπίας χρησιμοποιώντας τους πίνακες συχνότητας ως εξής:

α) Εντροπία χρησιμοποιώντας τον πίνακα συχνότητας ενός χαρακτηριστικού:

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)  
 = Entropy (0.36, 0.64)  
 = - (0.36 log<sub>2</sub> 0.36) - (0.64 log<sub>2</sub> 0.64)  
 = 0,94

Εικόνα 2.25 Εντροπία 2

[https://www.saedsayad.com/images/Entropy\\_3.png](https://www.saedsayad.com/images/Entropy_3.png)

β) Εντροπία χρησιμοποιώντας τον πίνακα συχνοτήτων δύο χαρακτηριστικών:

$$E(T, X) = \sum_{c \in X} P(c)E(c)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

$E(\text{PlayGolf, Outlook}) = P(\text{Sunny}) \cdot E(3,2) + P(\text{Overcast}) \cdot E(4,0) + P(\text{Rainy}) \cdot E(2,3)$   
 $= (5/14) \cdot 0.971 + (4/14) \cdot 0.0 + (5/14) \cdot 0.971$   
 $= 0.693$

Εικόνα 2.26 Εντροπία 3

[https://www.saedsayad.com/images/Entropy\\_2.png](https://www.saedsayad.com/images/Entropy_2.png)

### Κέρδος πληροφοριών

Το κέρδος πληροφορόρησης βασίζεται στη μείωση της εντροπίας μετά την κατάτμηση ενός συνόλου δεδομένων σε ένα χαρακτηριστικό. Η κατασκευή ενός δέντρου αποφάσεων αφορά το εύρημα ενός χαρακτηριστικού που επιστρέφει το υψηλότερο κέρδος πληροφορίας (δηλ. Τους πιο ομοιογενείς κλάδους).

- Βήμα 1: Υπολογίστε την εντροπία του στόχου.

**Entropy(PlayGolf) = Entropy (5,9)**  
 = Entropy (0.36, 0.64)  
 = - (0.36 log<sub>2</sub> 0.36) - (0.64 log<sub>2</sub> 0.64)  
 = 0.94

- Βήμα 2: Στη συνέχεια το σύνολο δεδομένων χωρίζεται σε διαφορετικά χαρακτηριστικά. Η εντροπία για κάθε κλάδο υπολογίζεται. Στη συνέχεια, προστίθεται αναλογικά, για να πάρει συνολική εντροπία για τη διάσπαση. Η προκύπτουσα εντροπία αφαιρείται από την εντροπία πριν από τη διάσπαση. Το αποτέλεσμα είναι η αύξηση της πληροφορίας ή η μείωση της εντροπίας.

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
		Gain = 0.247	

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
		Gain = 0.029	

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
		Gain = 0.152	

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
		Gain = 0.048	

Εικόνα 2.27 Περιπτώσεις κέρδους

[https://www.saedsayad.com/images/Entropy\\_attributes.png](https://www.saedsayad.com/images/Entropy_attributes.png)

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

$G(\text{PlayGolf}, \text{Outlook}) = E(\text{PlayGolf}) - E(\text{PlayGolf}, \text{Outlook})$ $= 0.940 - 0.693 = 0.247$
--

Εικόνα 2.28 Υπολογισμός κέρδους

[https://www.saedsayad.com/images/Entropy\\_gain.png](https://www.saedsayad.com/images/Entropy_gain.png)

- Βήμα 3: Επιλέξτε το χαρακτηριστικό με το μεγαλύτερο κέρδος πληροφοριών ως κόμβο απόφασης, διαιρέστε το σύνολο δεδομένων από τα υποκαταστήματά του και επαναλάβετε την ίδια διαδικασία σε κάθε κλάδο.



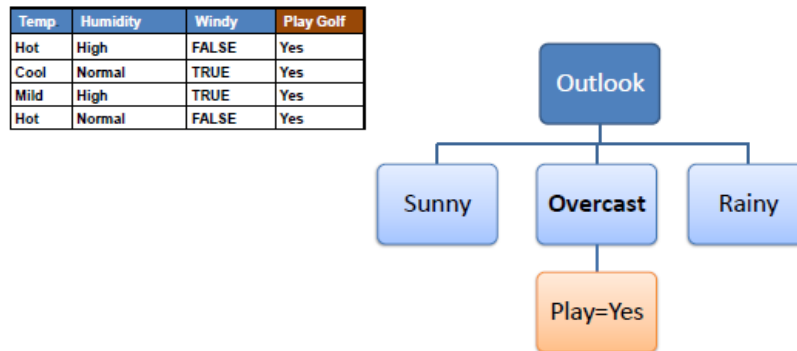
★		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

Εικόνα 2.29 Μεγαλύτερο κέρδος πληροφοριών

[https://www.saedsayad.com/images/Entropy\\_attribute\\_best.png](https://www.saedsayad.com/images/Entropy_attribute_best.png)

Outlook	Temp	Humidity	Windy	Play Golf	
Sunny	Sunny	Mild	High	FALSE	Yes
	Sunny	Cool	Normal	FALSE	Yes
	Sunny	Cool	Normal	TRUE	No
	Sunny	Mild	Normal	FALSE	Yes
	Sunny	Mild	High	TRUE	No
Overcast	Overcast	Hot	High	FALSE	Yes
	Overcast	Cool	Normal	TRUE	Yes
	Overcast	Mild	High	TRUE	Yes
	Overcast	Hot	Normal	FALSE	Yes
Rainy	Rainy	Hot	High	FALSE	No
	Rainy	Hot	High	TRUE	No
	Rainy	Mild	High	FALSE	No
	Rainy	Cool	Normal	FALSE	Yes
	Rainy	Mild	Normal	TRUE	Yes

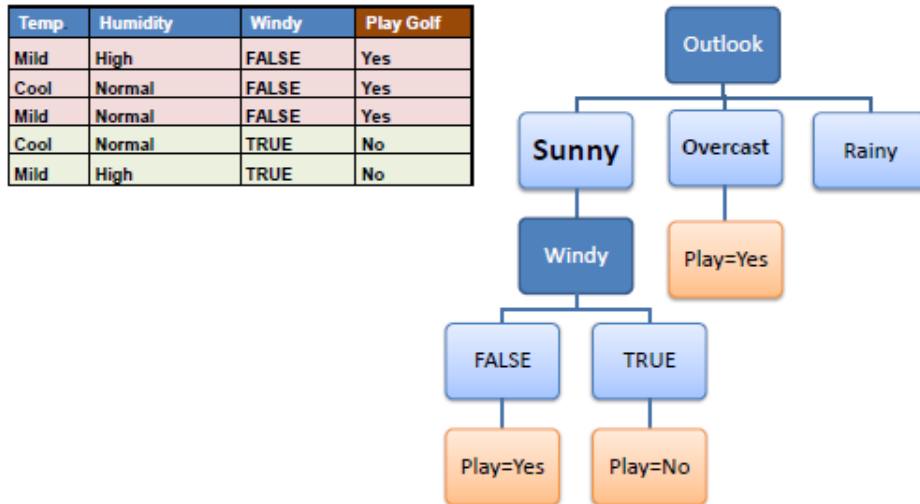
- Βήμα 4α: Ένα κλάσμα με εντροπία 0 είναι ένας κόμβος φύλλων.



Εικόνα 2.30 Γράφημα αποτελέσματος

[https://www.saedsayad.com/images/Entropy\\_overcast.png](https://www.saedsayad.com/images/Entropy_overcast.png)

- Βήμα 4β: Ένα κλάδο με εντροπία περισσότερο από 0 χρειάζεται περαιτέρω διάσπαση.

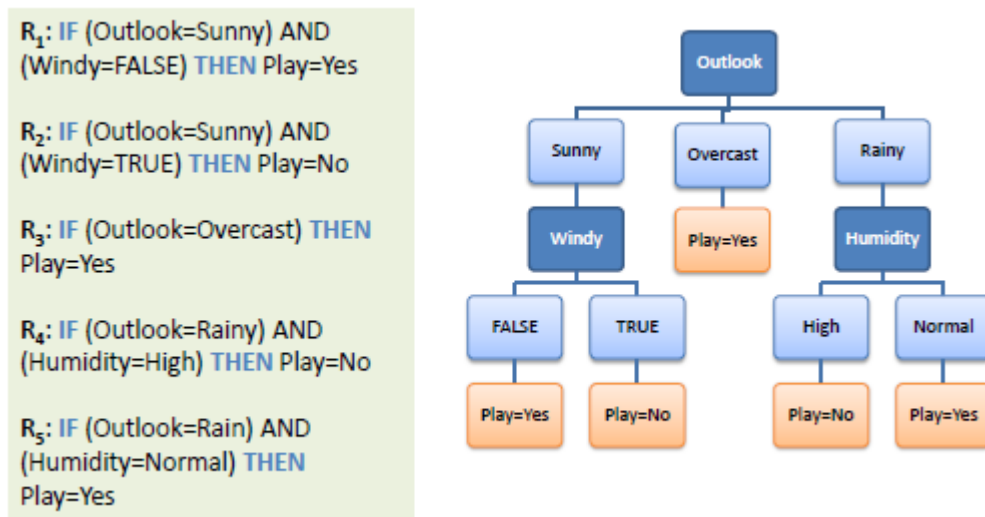


Εικόνα 2.31 Γράφημα αποτελέσματος 2

[https://www.saedsayad.com/images/Entropy\\_overcast.png](https://www.saedsayad.com/images/Entropy_overcast.png)

- Βήμα 5: Ο αλγόριθμος ID3 εκτελείται αναδρομικά στους μη-φύλλου κλάδους, μέχρι να ταξινομηθούν όλα τα δεδομένα.

Ένα δέντρο αποφάσεων μπορεί εύκολα να μετασχηματιστεί σε ένα σύνολο κανόνων με χαρτογράφηση από τον κόμβο ρίζας στους κόμβους των φύλλων ένα προς ένα.



Εικόνα 2.32 Δέντρο αποφάσεων

[https://www.saedsayad.com/images/Decision\\_rules.png](https://www.saedsayad.com/images/Decision_rules.png)

### *Πλησιέστερος γείτονας (K-NN Algorithm)*

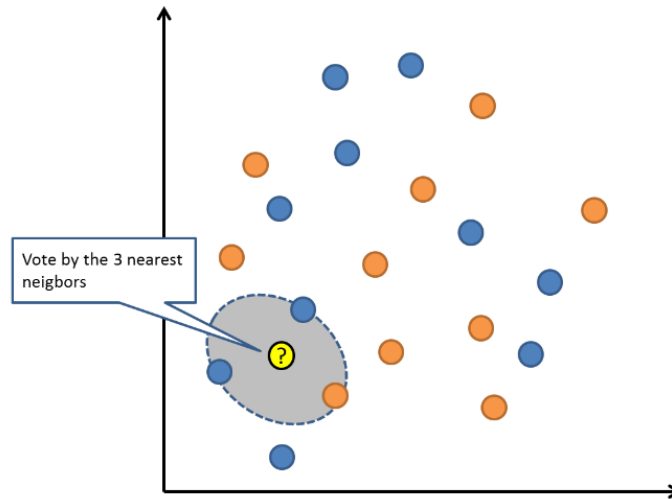
K πλησιέστεροι γείτονες είναι ένας απλός αλγόριθμος που αποθηκεύει όλες τις διαθέσιμες περιπτώσεις και ταξινομεί νέες περιπτώσεις με βάση ένα μέτρο ομοιότητας (π.χ. λειτουργίες απόστασης). Το KNN χρησιμοποιήθηκε στη στατιστική εκτίμηση και την αναγνώριση προτύπων ήδη στις αρχές της δεκαετίας του 1970 ως μη παραμετρική τεχνική.

Για την ορθή λειτουργία του KNN τα δεδομένα πρέπει να υποστούν μία προεπεξεργασία με τους παρακάτω τρόπους:

- Ρυθμίζεται η κλίμακα των δεδομένων (rescale): Ο KNN λειτουργεί καλύτερα όταν τα δεδομένα εισόδου είναι στην ίδια κλίμακα. Η ρύθμιση των δεδομένων μεταξύ 0 και 1 αποτελεί καλή πρακτική, αλλιώς αν τα δεδομένα έχουν κατανομή κατά Gauss (Gaussian) είναι δυνατό να κανονικοποιηθούν.
- Παράλειψη ελλιπών δεδομένων: Αν λείπουν δεδομένα αυτό σημαίνει ότι η απόσταση δεν είναι υπολογίσιμη και η μόνη λύση είναι η παράλειψη αυτών των δεδομένων.
- Μείωση των διαστάσεων: Ο KNN λειτουργεί καλύτερα όταν υπάρχουν λιγότερες διαστάσεις. Υπάρχει δυνατότητα λειτουργίας και με υψηλό αριθμό διαστάσεων όμως άλλες μέθοδοι είναι πιο κατάλληλες για εκείνες τις περιπτώσεις. Ο KNN θα λειτουργήσει καλύτερα αν υπάρχει τρόπος μείωσης των διαστάσεων.

### *Αλγόριθμος*

Μια υπόθεση κατατάσσεται με πλειοψηφία των γειτόνων της, με την περίπτωση να ανατίθεται στην τάξη που είναι πιο κοινή μεταξύ των πλησιέστερων γειτόνων της K που μετρούνται με μια συνάρτηση απόστασης. Εάν  $K = 1$ , τότε η περίπτωση απλά ανατίθεται στην κατηγορία του κοντινότερου γείτονα.



Εικόνα 2.33 Αλγόριθμος πλησιέστερων γειτόνων

<https://www.oreilly.com/library/view/statistics-for-machine/9781788295758/assets/c0c1aa94-6928-4267-b852-c8d21172fd34.png>

Euclidean	$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$
Manhattan	$\sum_{i=1}^k  x_i - y_i $
Minkowski	$\left( \sum_{i=1}^k ( x_i - y_i ^q) \right)^{1/q}$

Εικόνα 2.34 Συναρτήσεις αποστάσεων

[https://www.saedsayad.com/images/KNN\\_similarity.png](https://www.saedsayad.com/images/KNN_similarity.png)

Θα πρέπει επίσης να σημειωθεί ότι και τα τρία μέτρα απόστασης ισχύουν μόνο για συνεχείς μεταβλητές. Στην περίπτωση κατηγορικών μεταβλητών πρέπει να χρησιμοποιηθεί η απόσταση Hamming. Αναδεικνύει επίσης το ζήτημα της τυποποίησης των αριθμητικών μεταβλητών μεταξύ 0 και 1 όταν υπάρχει ένα μίγμα αριθμητικών και κατηγορικών μεταβλητών στο σύνολο δεδομένων.

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

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

X	Y	Distance
Male	Male	0
Male	Female	1

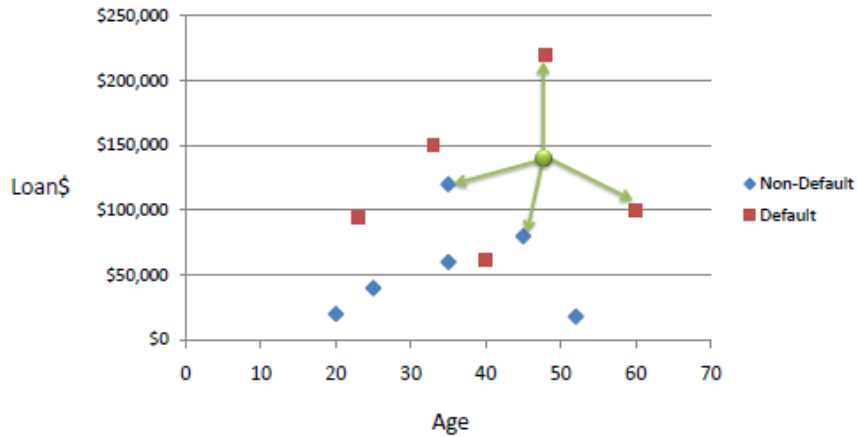
Εικόνα 2.35 Απόσταση Hamming

[https://www.saedsayad.com/images/KNN\\_hamming.png](https://www.saedsayad.com/images/KNN_hamming.png)

Η επιλογή της βέλτιστης τιμής για το K γίνεται καλύτερα με την πρώτη επιθεώρηση των δεδομένων. Γενικά, μια μεγάλη τιμή K είναι ακριβέστερη καθώς μειώνει το συνολικό θόρυβο αλλά δεν υπάρχει καμία εγγύηση. Η διασταυρούμενη επικύρωση είναι ένας άλλος τρόπος για τον εκ των υστέρων προσδιορισμό μιας καλής τιμής K χρησιμοποιώντας ένα ανεξάρτητο σύνολο δεδομένων για την επικύρωση της τιμής K. Ιστορικά, η βέλτιστη τιμή K για τα περισσότερα σύνολα δεδομένων ήταν μεταξύ 3-10. Αυτό παράγει πολύ καλύτερα αποτελέσματα από το KNN.

Παράδειγμα:

Εξετάστε τα ακόλουθα δεδομένα σχετικά με την πιστωτική προεπιλογή. Η ηλικία και το δάνειο είναι δύο αριθμητικές μεταβλητές (πρόβλεψη) και η προεπιλογή είναι ο στόχος.



Εικόνα 2.36 Γράφημα πρόβλεψης δανείου

[https://www.saedsayad.com/images/KNN\\_example\\_1.png](https://www.saedsayad.com/images/KNN_example_1.png)

Τώρα μπορούμε να χρησιμοποιήσουμε το σετ κατάρτισης για να ταξινομήσουμε μια άγνωστη περίπτωση (Age = 48 και Loan = \$ 142,000) χρησιμοποιώντας Euclidean απόσταση.

Αν  $K = 1$  τότε ο πλησιέστερος γείτονας είναι η τελευταία περίπτωση στο σύνολο εκπαίδευσης με Προεπιλογή = Y.

$$D = \text{Sqrt}[(48-33)^2 + (142000-150000)^2] = 8000.01 \gg \text{Default}=Y$$

Age	Loan	Default	Distance
25	\$40,000	N	102000
35	\$60,000	N	82000
45	\$80,000	N	62000
20	\$20,000	N	122000
35	\$120,000	N	22000
52	\$18,000	N	124000
23	\$95,000	Y	47000
40	\$62,000	Y	80000
60	\$100,000	Y	42000
48	\$220,000	Y	78000
33	\$150,000	Y	8000
<b>48</b>	<b>\$142,000</b>	<b>?</b>	

Euclidean Distance

$$D = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

Εικόνα 2.37 Euclidean απόσταση

[https://www.saedsayad.com/images/KNN\\_example\\_2.png](https://www.saedsayad.com/images/KNN_example_2.png)

Με το  $K = 3$ , υπάρχουν δύο Default = Y και ένα Default = N από τους τρεις πλησιέστερους γείτονες. Η πρόβλεψη για την άγνωστη περίπτωση είναι πάλι Προεπιλογή = Y.

Τυποποιημένη απόσταση

Ένα σημαντικό μειονέκτημα στον υπολογισμό των μέτρων απόστασης απευθείας από το σύνολο εκπαίδευσης είναι στην περίπτωση όπου οι μεταβλητές έχουν διαφορετικές κλίμακες μέτρησης ή υπάρχει ένα μίγμα αριθμητικών και κατηγορικών μεταβλητών. Για παράδειγμα, αν μια μεταβλητή βασίζεται σε ετήσιο εισόδημα σε δολάρια και το άλλο βασίζεται στην ηλικία των ετών τότε το εισόδημα θα έχει πολύ μεγαλύτερη επίδραση στην υπολογισμένη απόσταση. Μία λύση είναι η τυποποίηση του σετ εκπαίδευσης, όπως φαίνεται παρακάτω.

Age	Loan	Default	Distance
0.125	0.11	N	0.7652
0.375	0.21	N	0.5200
0.625	0.31	N	0.3160
0	0.01	N	0.9245
0.375	0.50	N	0.3428
0.8	0.00	N	0.6220
0.075	0.38	Y	0.6669
0.5	0.22	Y	0.4437
1	0.41	Y	0.3650
0.7	1.00	Y	0.3861
0.325	0.65	Y	0.3771
<b>0.7</b>	<b>0.61</b>	<b>?</b>	

Standardized Variable

$$X_s = \frac{X - Min}{Max - Min}$$

Εικόνα 2.38 Τυποποιημένη μεταβλητή

[https://www.saedsayad.com/images/KNN\\_example\\_3.png](https://www.saedsayad.com/images/KNN_example_3.png)

Χρησιμοποιώντας την τυποποιημένη απόσταση στο ίδιο σετ εκπαίδευσης, η άγνωστη περίπτωση επέστρεψε έναν διαφορετικό γείτονα που δεν είναι καλό σημάδι ευρωστίας.

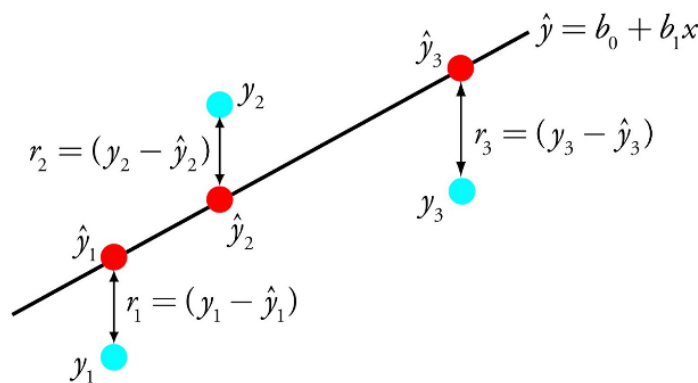


**Γραμμική παλινδρόμηση (Linear Regression)**

Η γραμμική παλινδρόμηση είναι ίσως ένας από τους πιο γνωστούς και κατανοητούς αλγορίθμους στις στατιστικές και στη μηχανική μάθηση.

Η προγνωστική μοντελοποίηση αφορά κατά κύριο λόγο την ελαχιστοποίηση του σφάλματος ενός μοντέλου ή την πραγματοποίηση των πιο ακριβών προβλέψεων, σε βάρος της εξήγησης.

Η παράσταση της γραμμικής παλινδρόμησης είναι μια εξίσωση που περιγράφει μια γραμμή που ταιριάζει καλύτερα στη σχέση μεταξύ των μεταβλητών εισόδου (x) και των μεταβλητών εξόδου (y), βρίσκοντας συγκεκριμένες σταθμίσεις για τις μεταβλητές εισόδου που ονομάζονται συντελεστές (B).



Εικόνα 2.39 Γραμμική παλινδρόμηση (Linear Regression)

[https://cdn-images-1.medium.com/max/2600/1\\*oXPGYqgTeIn0Ey3SWgkbsA.jpeg](https://cdn-images-1.medium.com/max/2600/1*oXPGYqgTeIn0Ey3SWgkbsA.jpeg)

Στην απλή γραμμική παλινδρόμηση υπάρχει η ανεξάρτητη μεταβλητή  $x_i$  και δύο παράμετροι  $\beta_0$  και  $\beta_1$ . Το μοντέλο έχει τη μορφή:

$$Y_i = \beta_0 + \beta_1 * x_i + e_i$$

$i=1, m$ , όπου  $e$  το σφάλμα της πρόβλεψης

Θα προβλέψουμε το  $y$  δεδομένης της εισόδου  $x$  και ο στόχος του αλγορίθμου μάθησης γραμμικής παλινδρόμησης είναι να βρούμε τις τιμές για τους συντελεστές  $B_0$  και  $B_1$ .

Διαφορετικές τεχνικές μπορούν να χρησιμοποιηθούν για να μάθουν το μοντέλο γραμμικής παλινδρόμησης από δεδομένα, όπως λύση γραμμικής άλγεβρας για βελτιστοποίηση συνήθους ελαχίστων τετραγώνων και κλίσης

Η γραμμική παλινδρόμηση υπήρξε εδώ και πάνω από 200 χρόνια και έχει μελετηθεί εκτενώς. Μερικοί ορθολογικοί κανόνες όταν χρησιμοποιεί κανείς αυτήν την τεχνική είναι να αφαιρέσει μεταβλητές που είναι παρόμοιες (συσχετισμένες) και να αφαιρέσει θόρυβο από τα δεδομένα μας, αν είναι δυνατόν.

## Παραδοχές για την εφαρμογή του απλού γραμμικού μοντέλου

### Γραμμικότητα

Η κατανομή της  $Y_i$  έχει, για τα διάφορα επίπεδα  $x_i$ ,  $i=1,2,\dots,v$  της  $X$ , μέση τιμή  $E(\frac{Y}{X})=a+\beta X$  όπου  $a$  και  $\beta$  παράμετροι που εκτιμώνται από το δείγμα  $(x_i, y_i)_{i=1,2,\dots,v}$ . Άρα, υποθέτουμε ότι οι μέσες τιμές της  $Y$ , για τα διάφορα επίπεδα της  $X$ , είναι γραμμικές συναρτήσεις της  $X$  (ότι βρίσκονται δηλαδή σε ευθεία γραμμή). Σημειώνουμε ότι στο μοντέλο  $Y = a + \beta X + e$ , τυχαίες μεταβλητές είναι μόνο οι  $Y$  και  $e$ .

### Σταθερότητα Διασποράς

Οι κατανομές της  $Y$  έχουν ίδια διασπορά για όλα τα επίπεδα της  $X$ , δηλαδή,  $\text{Var}(\frac{Y}{X} | x_i) = \sigma^2$ .

### Ανεξαρτησία

Οι τιμές της  $Y$  που αντιστοιχούν στα διάφορα επίπεδα της  $X$  είναι ανεξάρτητες μεταξύ τους. Εξαρτημένα  $Y$  εμφανίζονται συνήθως όταν παίρνουμε παρατηρήσεις από την ίδια πειραματική μονάδα σε διαφορετικές χρονικές στιγμές (π.χ. μετράμε την πίεση ή το βάρος του ίδιου ατόμου ανά εβδομάδα). Επίσης, σε περιπτώσεις όπου χρησιμοποιούνται μηχανές (π.χ. όργανα μέτρησης, κ.τλ) που αλλάζει η απόδοσή ανάλογα με τη χρήση αν ο χειριστής βελτιώνεται (ή χειροτερεύει) με την πάροδο του χρόνου. Είναι επομένως αναγκαίο, όταν έχουμε πειραματικά δεδομένα που παίρνονται με χρονική σειρά, να κάνουμε ένα διάγραμμα υπολοίπων ως προς το χρόνο έστω και αν ο χρόνος δεν χρησιμοποιείται ως μεταβλητή στο μοντέλο

### Κανονικότητα

Η κατανομή της  $Y$  για όλα τα επίπεδα της  $X$  είναι κανονική. Η κανονικότητα μπορεί να ελεγχθεί με διάφορους τρόπους όπως:

- Με ιστόγραμμα
- Με φυλλογράφημα (stem and leaf plot)
- Με θηκόγραμμα (box plot)

- Με διάγραμμα πιθανοτήτων (normal probability plot)

Όταν διαπιστώνεται παραβίαση της κανονικότητας μπορούμε, σε πολλές περιπτώσεις, να αντιμετωπίσουμε το πρόβλημα με κατάλληλους μετασχηματισμούς στις μεταβλητές

### *Μη-Γραμμική Παλινδρόμηση(NonLinear Regression)*

Η μη γραμμική παλινδρόμηση είναι μια στατιστική τεχνική που βοηθά στην περιγραφή μη γραμμικών σχέσεων σε πειραματικά δεδομένα. Τα μοντέλα μη γραμμικής παλινδρόμησης θεωρούνται γενικά παραμετρικά, όπου το μοντέλο περιγράφεται ως μη γραμμική εξίσωση. Συνήθως χρησιμοποιούνται μέθοδοι μηχανικής μάθησης για μη παραμετρική μη γραμμική παλινδρόμηση.

Η παραμετρική μη γραμμική παλινδρόμηση μοντελοποιεί τη εξαρτημένη μεταβλητή ως συνάρτηση ενός συνδυασμού μη γραμμικών παραμέτρων και μιας ή περισσοτέρων ανεξάρτητων μεταβλητών (αποκαλούμενων προγνωστικών). Το μοντέλο μπορεί να είναι μονοδιάστατο (μεταβλητή απλής ανταπόκρισης) ή πολλαπλών μεταβλητών (πολλαπλές μεταβλητές απόκρισης).

Οι παράμετροι μπορούν να λάβουν τη μορφή εκθετικής, τριγωνομετρικής, ισχύος ή οποιασδήποτε άλλης μη γραμμικής συνάρτησης. Για να προσδιοριστούν οι εκτιμήσεις μη γραμμικών παραμέτρων, τυπικά χρησιμοποιείται ένας επαναληπτικός αλγόριθμος.

$$Y = f(X, \beta) + \varepsilon$$

όπου  $\beta$  αντιπροσωπεύει τις μη γραμμικές εκτιμήσεις παραμέτρων που πρέπει να υπολογιστούν και  $\varepsilon$  αντιπροσωπεύει τους όρους σφάλματος.

Οι δημοφιλείς αλγόριθμοι για την τοποθέτηση μιας μη γραμμικής παλινδρόμησης περιλαμβάνουν:

- Gauss-Newton αλγόριθμος
- Αλγόριθμος καθόδου κλίσης
- Αλγόριθμος Levenberg-Marquardt

## Κεφάλαιο 3

### Εισαγωγή

Στο προηγούμενο κεφάλαιο παρουσιάσαμε αρκετές πληροφορίες ως αναφορά την μηχανική μάθηση σε αρκετούς τομείς καθώς οι εφαρμογές ποικίλουν πραγματικά και οι δυνατότητες είναι αμέτρητες. Εξηγήσαμε τις διαδικασίες μάθησης καθώς και τις έννοιες των αλγορίθμων μηχανικής μάθησης. Όλα αυτά έγιναν με σκοπό να "εισβάλουμε" για λίγο πίσω από ένα έτοιμο αλγόριθμο που διαθέτει το Matlab, αυτό διότι θα μας βοηθήσει πάρα πολύ στην κατανόηση των αλγορίθμων των εφαρμογών που ακολουθούν στο παρόν και τελευταίο κεφάλαιο. Επίσης θα γίνει λεπτομερή αναφορά και επεξήγηση γύρω από τα βήματα που ακολουθήσαμε για το αποτέλεσμα αυτό, έτσι ώστε να βοηθηθούν άτομα που είτε επιχειρούν κάτι παρόμοιο είτε προσπαθούν για κάτι καλύτερο.

### Τι είναι το Matlab;

Το MATLAB είναι ένα σύγχρονο ολοκληρωμένο μαθηματικό λογισμικό πακέτο που χρησιμοποιείται σε πανεπιστημιακά μαθήματα αλλά και ερευνητικές και άλλες εφαρμογές με επιστημονικούς υπολογισμούς (scientific computing). Το όνομά του προέρχεται από τα αρχικά γράμματα των λέξεων MATrix LABoratory (εργαστήριο πινάκων).

Το MATLAB είναι ένα διαδραστικό (interactive) πρόγραμμα για αριθμητικούς υπολογισμούς και οπτικοποίηση δεδομένων (data visualization) με δυνατότητες προγραμματισμού που το καθιστούν ένα ισχυρό και χρήσιμο εργαλείο στις μαθηματικές και φυσικές επιστήμες. Σε αντίθεση με τα λογισμικά Maple και Mathematica, το MATLAB στις αρχικές του εκδοχές δεν έκανε συμβολικούς υπολογισμούς. Στις νεότερες εκδοχές του, το πακέτο περιλαμβάνει εργαλείοι που επιτρέπουν συμβολικούς υπολογισμούς.

Όπως υποδηλώνεται και από το όνομά του, το MATLAB είναι ειδικά σχεδιασμένο για υπολογισμούς με πίνακες, όπως η επίλυση γραμμικών συστημάτων, η εύρεση ιδιοτιμών και ιδιοδιανυσμάτων, η αντιστροφή τετραγωνικών πινάκων κλπ. Επιπλέον το πακέτο αυτό είναι εφοδιασμένο με πολλές επιλογές για γραφικά (δηλ. την κατασκευή γραφικών παραστάσεων) και προγράμματα γραμμένα στη δική του γλώσσα προγραμματισμού για την επίλυση άλλων

προβλημάτων όπως η εύρεση των ριζών μη γραμμικής εξίσωσης, η επίλυση μη γραμμικών συστημάτων, η επίλυση προβλημάτων αρχικών τιμών με συνήθεις διαφορικές εξισώσεις κα.

Η γλώσσα προγραμματισμού του MATLAB δίνει την ευχέρεια στον χρήστη να το επεκτείνει με δικά του προγράμματα. Επιπλέον είναι σχεδιασμένο για την αριθμητική επίλυση προβλημάτων σε αριθμητική πεπερασμένης ακρίβειας (finite-precision arithmetic), δηλαδή δεν βρίσκει την ακριβή αλλά μια προσεγγιστική λύση ενός προβλήματος. Αυτή είναι και η βασική του διαφορά από τα συστήματα συμβολικών υπολογισμών όπως η Maple και το Mathematica. Τέλος, στην επίσημη ιστοσελίδα της MATLAB: <http://www.mathworks.com> μπορεί κάποιος να βρει μια πληθώρα πληροφοριών τόσο για αρχάριους όσο και προχωρημένους.

### Πλεονεκτήματα του MATLAB

- Ευκολότερη εκμάθηση από μια γλώσσα προγραμματισμού.
- Βελτιστοποιημένος κώδικας για διεξαγωγή υπολογισμών με πίνακες.
- Γλώσσα προγραμματισμού για ανάπτυξη εφαρμογών και ταυτόχρονα λογισμικού υλοποίησης επιστημονικών υπολογισμών.
- Εύκολος εντοπισμός και διόρθωση λαθών.
- Φιλικό περιβάλλον επικοινωνίας με το χρήστη.

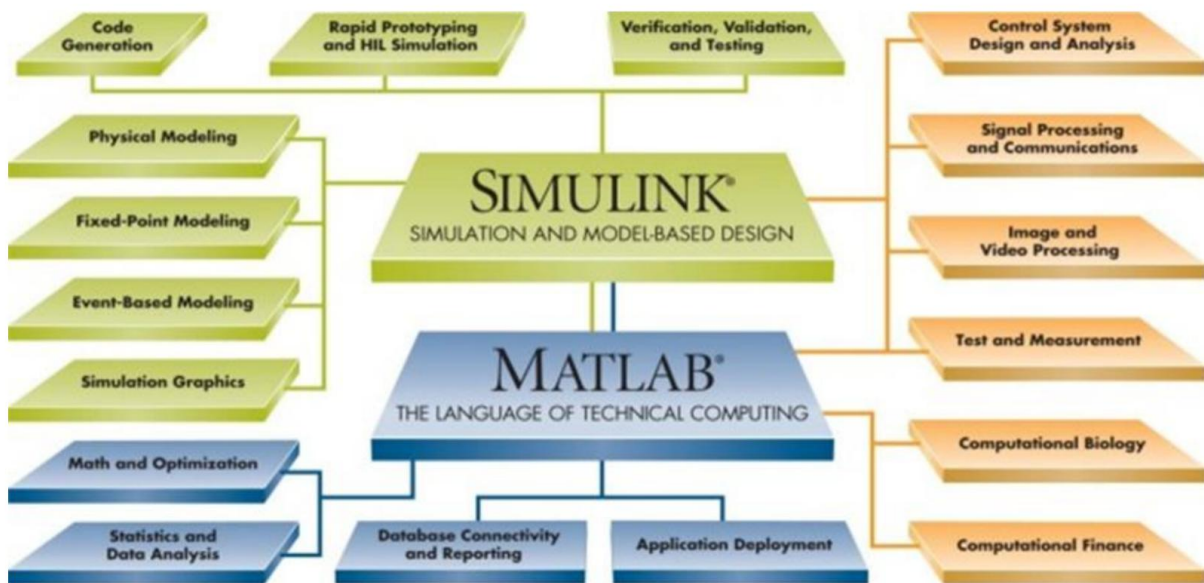
### Μειονεκτήματα του MATLAB

- Εξειδικευμένη γλώσσα προγραμματισμού.
- Το λογισμικό MATLAB αναπτύχθηκε μόνο για διεξαγωγή επιστημονικών υπολογισμών κι έτσι δεν ενδείκνυται ή υποστηρίζει την ανάπτυξη άλλου είδους εφαρμογών, π.χ. επεξεργασία κειμένου.
- Οι αναπτυσσόμενες εφαρμογές υστερούν σε απόδοση από την άποψη χρόνου εκτέλεσης σε σχέση με αντίστοιχες που αναπτύσσονται με τις κλασικές γλώσσες προγραμματισμού (C, C++, Fortran).

### Εφαρμογές:

- Μαθηματική ανάλυση & Επεξεργασία
- Ψηφιακή Επεξεργασία Σήματος
- Επεξεργασία Εικόνας και Ήχου
- Σχεδιασμός Συστημάτων Αυτομάτου Ελέγχου
- Οικονομικές εφαρμογές
- Βιομηχανικές Εφαρμογές
- Περισσότερα από 60 toolboxes!

### Εργαλειοθήκες - Toolboxes



Εικόνα 3.1 Βιβλιοθήκες του Matlab

## Αριθμητική Ταξινόμηση (Numerical Classification)

### Breast Cancer Wisconsin

Η αριθμητική ταξινόμηση αποτελεί ένα απλο παραδειγμα ταξινόμησης δεδομένων μεσω του Matlab. Έχουμε εκατονταδες αριθμητικά δεδομένα οπου συνδιαστικά μας πληροφορούν για καποιο αποτέλεσμα. Αυτό το αποτέλεσμα είναι ο σταθερος παράγοντας ταξινόμησης των δεδομένων. Παρακάτω ακολουθούν παραδειγματα και επεξηγήσεις για του λόγου το αληθές.

Το παρακάτω παραδειγμα αντλήθηκε από :

(<https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/>)

Τα χαρακτηριστικά υπολογίζονται από μια ψηφιοποιημένη εικόνα μιας μάζας στήθους. Περιγράφουν τα χαρακτηριστικά των κυτταρικών πυρήνων που υπάρχουν στην εικόνα. Τα δεδομένα που χρησιμοποιούμε έχουν την παρακάτω μορφή:

```
842302,M,17.99,10.38,122.8,1001,0.1184,0.2776,0.3001,0.1471,0.2419,0.07871,1.095,0.9053,8.589,1
842517,M,20.57,17.77,132.9,1326,0.08474,0.07864,0.0869,0.07017,0.1812,0.05667,0.5435,0.7339,3.3
84300903,M,19.69,21.25,130,1203,0.1096,0.1599,0.1974,0.1279,0.2069,0.05999,0.7456,0.7869,4.585,
```

Εικόνα 3.2 Χαρακτηριστικές τιμές γνωρισμάτων του καρκίνου

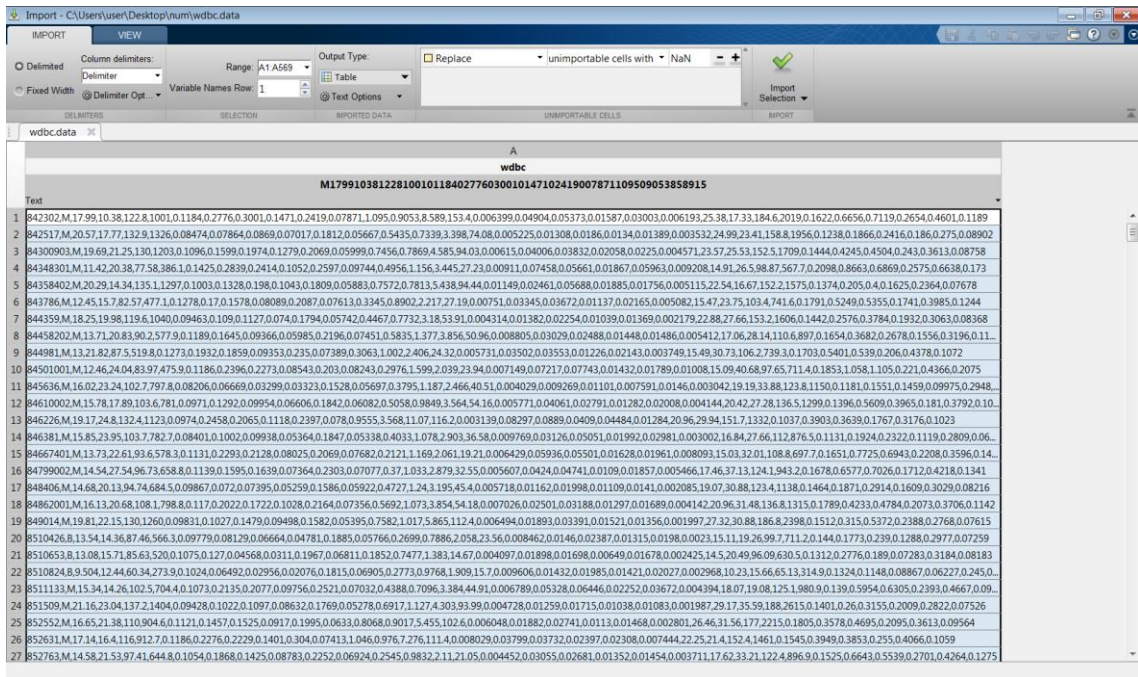
1) ID number

2) Diagnosis (M= malignant, B= benign)

3-32) χαρακτηριστικές τιμές όπου ξεχωρίζουν εάν τα γνωρίσματα της τυχαίας εικόνας 842302 τείνουν να ορίσουν την εικόνα ως malignant.

Ξεκινώντας λοιπόν φορτώνουμε τα δεδομένα μας στο χώρο εργασίας του Matlab μέσω της καρτέλας **Home** επιλέγουμε το **Import Data**, βρίσκουμε το αρχείο και μόλις το ανοίξουμε βλέπουμε το παρακάτω





Εικόνα 3.3 Πίνακας εισαγωγής των τιμών

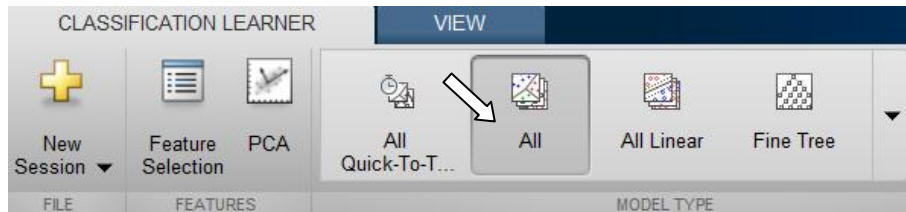
έπειτα ακολουθούμε τις εξής ενέργειες ώστε να το φέρουμε στη μορφή που θέλουμε



Εικόνα 3.4 Διαχωρισμός σε στήλες και εισαγωγή σε διάταξη πίνακα

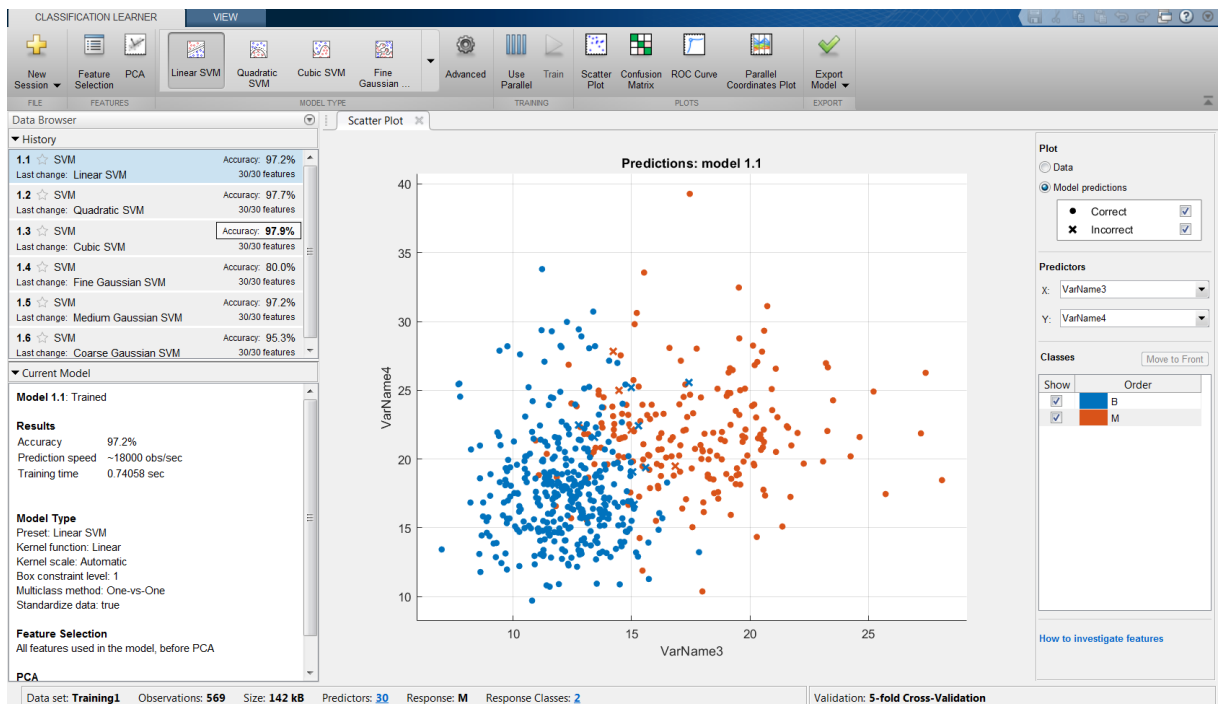
αφού κάνουμε τις παραπάνω ενέργειες και δημιουργηθεί ο πίνακας στο workspace παμε στην καρτέλα **Apps** και βρίσκουμε το **Classification Learner**. Ξεκινάμε **New Session**, επιλέγουμε τον πίνακα που έχουμε δημιουργήσει και στο κελί **Response** διαλέγουμε τον τύπο **categorical** δηλαδή τον σταθερό όρο με τον οποίο θα γίνει η ταξινόμηση όπως αναφέραμε παραπάνω.

Στην καρτέλα Classification Learner, στην ενότητα Type Model, επιλέγουμε να γίνει ταξινόμηση με όλους τους ταξινομητές και στη συνέχεια πατάμε το **Train**



Εικόνα 3.5 Καρτέλα ταξινομητών

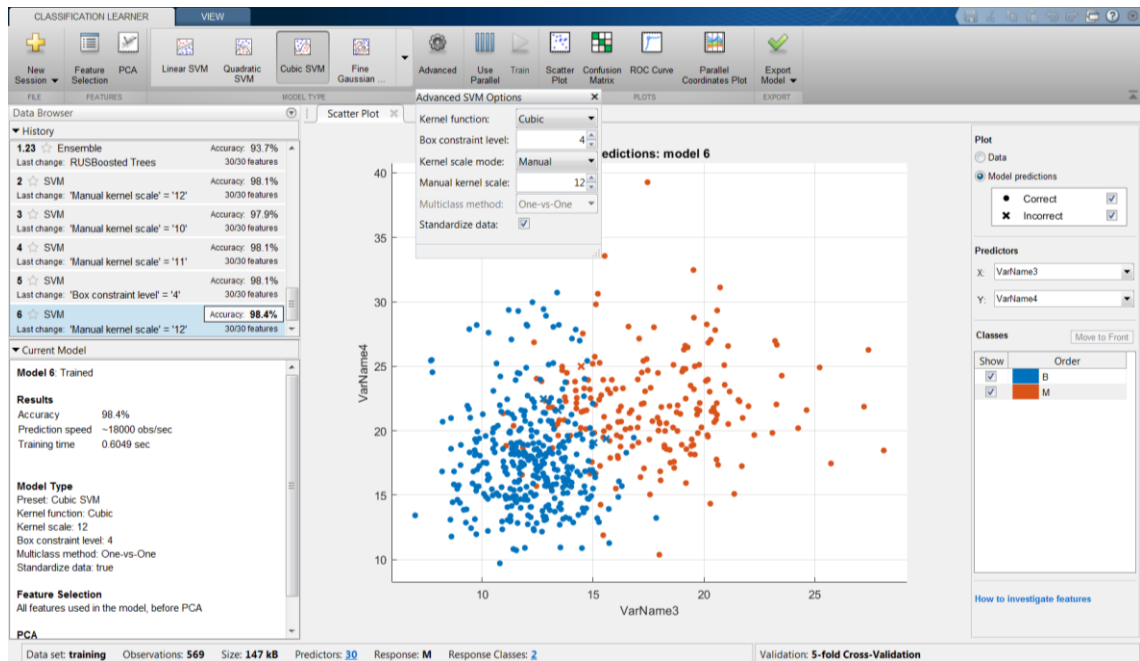
Η επιλεγόμενοι ταξινομητές εμφανίζονται στη λίστα Ιστορικό. Όταν τελειώσουν την εκπαίδευση, το καλύτερο ποσοστό Ακρίβειας επισημαίνεται σε ένα κουτί.



Εικόνα 3.6 Predictions model

Όπως βλέπουμε έχουμε δύο ταξινομητές με το υψηλότερο ποσοστό, είναι οι Quadratic & Cubic SVM με 98,2% ακρίβεια.

Επιπλέον μπορούμε να αυξήσουμε τα ποσοστα όπως φαίνεται παρακάτω χρησιμοποιώντας την **Advance** επιλογή. Οι τυχαίες τιμες που επιλεξαμε είναι 4 και 12 αντιστοιχα και ειχαμε αυξηση της ταξης του 0,2%.

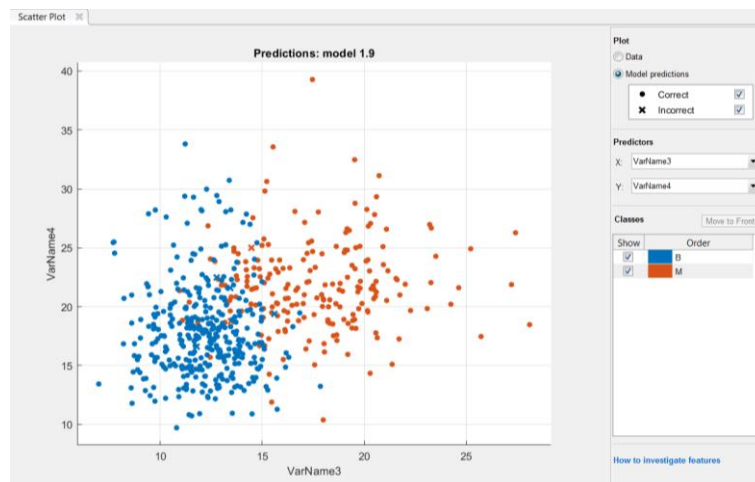


Εικόνα 3.7 Predictions model

Παρακάτω ακολουθούν γραφήματα απεικόνισης του ταξινομητή.

Scatter plot:

είναι ένας τύπος γραφήματος που χρησιμοποιεί καρτεσιανές συντεταγμένες για την εμφάνιση τιμών για ένα σύνολο δεδομένων. Τα δεδομένα παρουσιάζονται ως μια συλλογή σημείων, η κάθε μία από τις οποίες έχει την τιμή μιας μεταβλητής που καθορίζει τη θέση στον οριζόντιο άξονα και την τιμή της άλλης μεταβλητής που καθορίζει τη θέση στον κατακόρυφο άξονα .

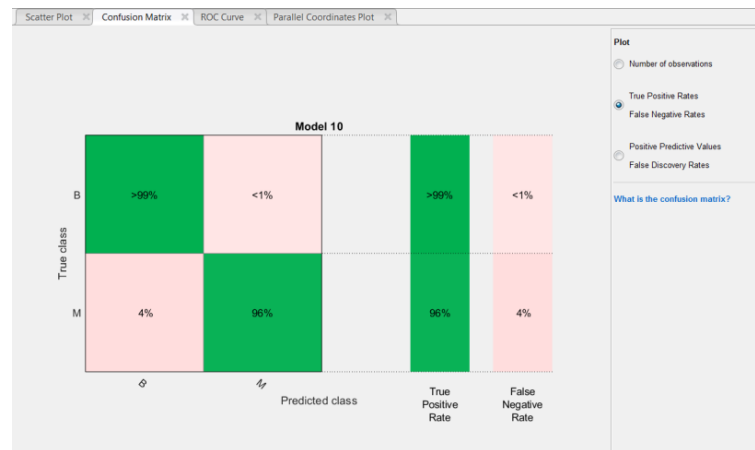


Εικόνα 3.8 Predictions model

Confusion matrix:

Κάθε σειρά του πίνακα αντιπροσωπεύει τις περιπτώσεις σε μια προβλεπόμενη κλάση ενώ κάθε στήλη αντιπροσωπεύει τις περιπτώσεις σε μια πραγματική.

Πρόκειται για ένα ειδικό είδος πίνακα έκτακτης ανάγκης, με δύο διαστάσεις ("πραγματικό" και "προβλεπόμενο"), και πανομοιότυπα σύνολα "κλάσεων" και στις δύο διαστάσεις (κάθε συνδυασμός διαστάσεων και κλάσης είναι μια μεταβλητή στον πίνακα ενδεχόμενων γεγονότων).

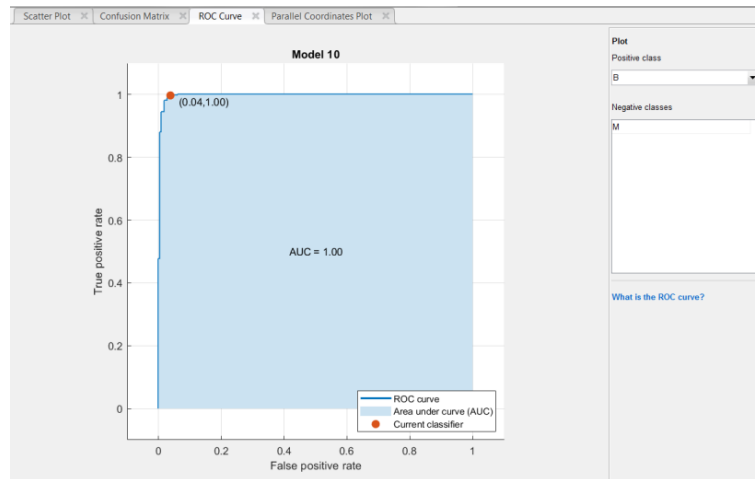


Εικόνα 3.9 Confusion matrix

ROC curve:

Η καμπύλη ROC δείχνει πραγματικό θετικό ρυθμό έναντι ψευδώς θετικού ρυθμού για τον τρέχοντα επιλεγμένο εκπαιδευμένο ταξινομητή.

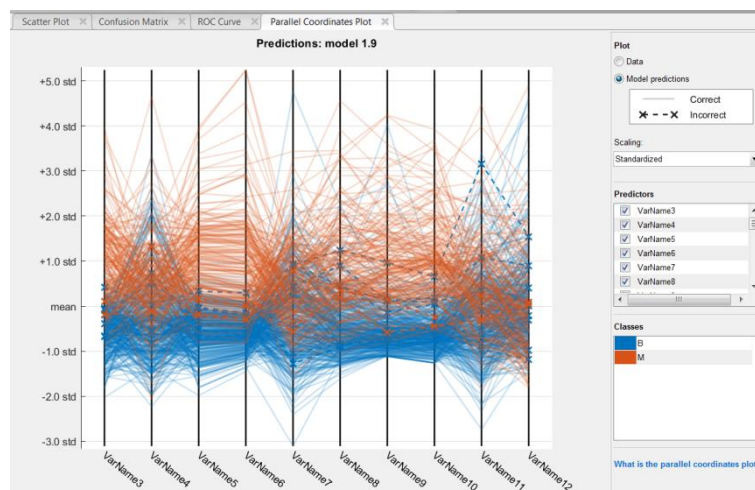
Ο δείκτης στο γράφημα δείχνει την απόδοση του τρέχοντος επιλεγμένου ταξινομητή. Ο δείκτης εμφανίζει τις τιμές της ψευδώς θετικής συχνότητας (FPR) και την πραγματική θετική συχνότητα (TPR) για τον τρέχοντα επιλεγμένο ταξινομητή.



Εικόνα 3.10 ROC curve

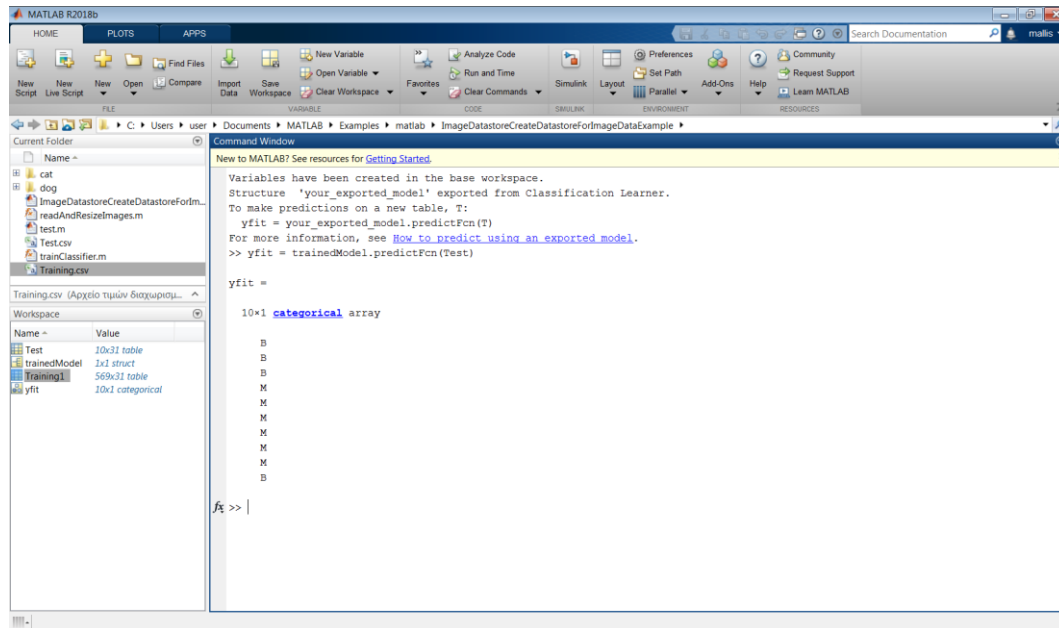
Parallel Coordinates Plot:

Τα Σχήματα είναι ιδανικά για τη σύγκριση πολλών μεταβλητών μεταξύ τους και για την προβολή των σχέσεων μεταξύ τους. Σε ένα Parallel Coordinates Plot, κάθε μεταβλητή δίνεται στον δικό της άξονα και όλοι οι άξονες τοποθετούνται παράλληλα ο ένας στον άλλο. Κάθε άξονας μπορεί να έχει διαφορετική κλίμακα, καθώς κάθε μεταβλητή λειτουργεί από διαφορετική μονάδα μέτρησης ή όλοι οι άξονες μπορούν να ομαλοποιηθούν για να διατηρήσουν όλες τις κλίμακες ομοιόμορφες. Οι τιμές απεικονίζονται ως μια σειρά γραμμών που συνδέονται σε όλους τους άξονες



Εικόνα 3.11 Parallel Coordinates Plot

Μόλις τελειώσουμε και μ αυτό το κομμάτι της ταξινόμησης πρέπει να εξάγουμε το μοντέλο του ταξινομητή για επερχόμενη προβλεψη του αρχείου Test. Στο αρχείο Test έχουμε επιλέξει λιγα αρχεία έτσι ώστε να να γίνει η πρόβλεψη.



Εικόνα 3.12 Αποτελέσματα πρόβλεψης

```
function [trainedClassifier, validationAccuracy] =
trainClassifier(trainingData)
% [trainedClassifier, validationAccuracy] =
trainClassifier(trainingData)
% returns a trained classifier and its accuracy. This code
recreates the
% classification model trained in Classification Learner
app. Use the
% generated code to automate training the same model with
new data, or to
% learn how to programmatically train models.
%
% Input:
%     trainingData: a table containing the same predictor
and response
%     columns as imported into the app.
%
% Output:
%     trainedClassifier: a struct containing the trained
classifier. The
%     struct contains various fields with information
about the trained
%     classifier.
%     trainedClassifier.predictFcn: a function to make
predictions on new
%     data.
%
```



```

%     validationAccuracy: a double containing the
accuracy in percent. In
%     the app, the History list displays this overall
accuracy score for
%     each model.
%
% Use the code to train the model with new data. To
retrain your
% classifier, call the function from the command line with
your original
% data or new data as the input argument trainingData.
%
% For example, to retrain a classifier trained with the
original data set
% T, enter:
% [trainedClassifier, validationAccuracy] =
trainClassifier(T)
%
% To make predictions with the returned
'trainedClassifier' on new data T2,
% use
% yfit = trainedClassifier.predictFcn(T2)
%
% T2 must be a table containing at least the same
predictor columns as used
% during training. For details, enter:
%     trainedClassifier.HowToPredict

% Auto-generated by MATLAB on 06-May-2019 15:40:10

% Extract predictors and response
% This code processes the data into the right shape for
training the
% model.
inputTable = trainingData;
predictorNames = {'VarName3', 'VarName4', 'VarName5',
'VarName6', 'VarName7', 'VarName8', 'VarName9',
'VarName10', 'VarName11', 'VarName12', 'VarName13',
'VarName14', 'VarName15', 'VarName16', 'VarName17',
'VarName18', 'VarName19', 'VarName20', 'VarName21',
'VarName22', 'VarName23', 'VarName24', 'VarName25',
'VarName26', 'VarName27', 'VarName28', 'VarName29',
'VarName30', 'VarName31', 'VarName32'};
predictors = inputTable(:, predictorNames);
response = inputTable.M;

```



```

isCategoricalPredictor = [false, false, false, false,
false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false, false,
false, false];

% Train a classifier
% This code specifies all the classifier options and
trains the classifier.
classificationSVM = fitcsvm(...
    predictors, ...
    response, ...
    'KernelFunction', 'polynomial', ...
    'PolynomialOrder', 3, ...
    'KernelScale', 'auto', ...
    'BoxConstraint', 1, ...
    'Standardize', true, ...
    'ClassNames', categorical({'B'; 'M'}));

% Create the result struct with predict function
predictorExtractionFcn = @(t) t(:, predictorNames);
svmPredictFcn = @(x) predict(classificationSVM, x);
trainedClassifier.predictFcn = @(x)
svmPredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedClassifier.RequiredVariables = {'VarName3',
'VarName4', 'VarName5', 'VarName6', 'VarName7',
'VarName8', 'VarName9', 'VarName10', 'VarName11',
'VarName12', 'VarName13', 'VarName14', 'VarName15',
'VarName16', 'VarName17', 'VarName18', 'VarName19',
'VarName20', 'VarName21', 'VarName22', 'VarName23',
'VarName24', 'VarName25', 'VarName26', 'VarName27',
'VarName28', 'VarName29', 'VarName30', 'VarName31',
'VarName32'};
trainedClassifier.ClassificationSVM = classificationSVM;
trainedClassifier.About = 'This struct is a trained model
exported from Classification Learner R2018b.';
trainedClassifier.HowToPredict = sprintf('To make
predictions on a new table, T, use: \n yfit =
c.predictFcn(T) \nreplacing 'c' with the name of the
variable that is this struct, e.g. 'trainedModel'. \n
\nThe table, T, must contain the variables returned by: \n
c.RequiredVariables \nVariable formats (e.g.
matrix/vector, datatype) must match the original training
data. \nAdditional variables are ignored. \n \nFor more
information, see <a

```

```

href="matlab:helpview(fullfile(docroot, 'stats',
'stats.map'),
'appclassification_exportmodeltoworkspace')">How to
predict using an exported model</a>.);

% Extract predictors and response
% This code processes the data into the right shape for
training the
% model.
inputTable = trainingData;
predictorNames = {'VarName3', 'VarName4', 'VarName5',
'VarName6', 'VarName7', 'VarName8', 'VarName9',
'VarName10', 'VarName11', 'VarName12', 'VarName13',
'VarName14', 'VarName15', 'VarName16', 'VarName17',
'VarName18', 'VarName19', 'VarName20', 'VarName21',
'VarName22', 'VarName23', 'VarName24', 'VarName25',
'VarName26', 'VarName27', 'VarName28', 'VarName29',
'VarName30', 'VarName31', 'VarName32'};
predictors = inputTable(:, predictorNames);
response = inputTable.M;
isCategoricalPredictor = [false, false, false, false,
false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false,
false, false, false, false, false, false, false, false,
false, false];

% Perform cross-validation
partitionedModel =
crossval(trainedClassifier.ClassificationSVM, 'KFold', 5);

% Compute validation predictions
[validationPredictions, validationScores] =
kfoldPredict(partitionedModel);

% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel,
'LossFun', 'ClassifError');
*
*
*

% To make prediction
yfit = trainedModel.predictFcn(Test)
yfit =

```

1×10 categorical array

B / B / B / M / M / M / M / M / M / B → (B=Benign, M=Malignant)

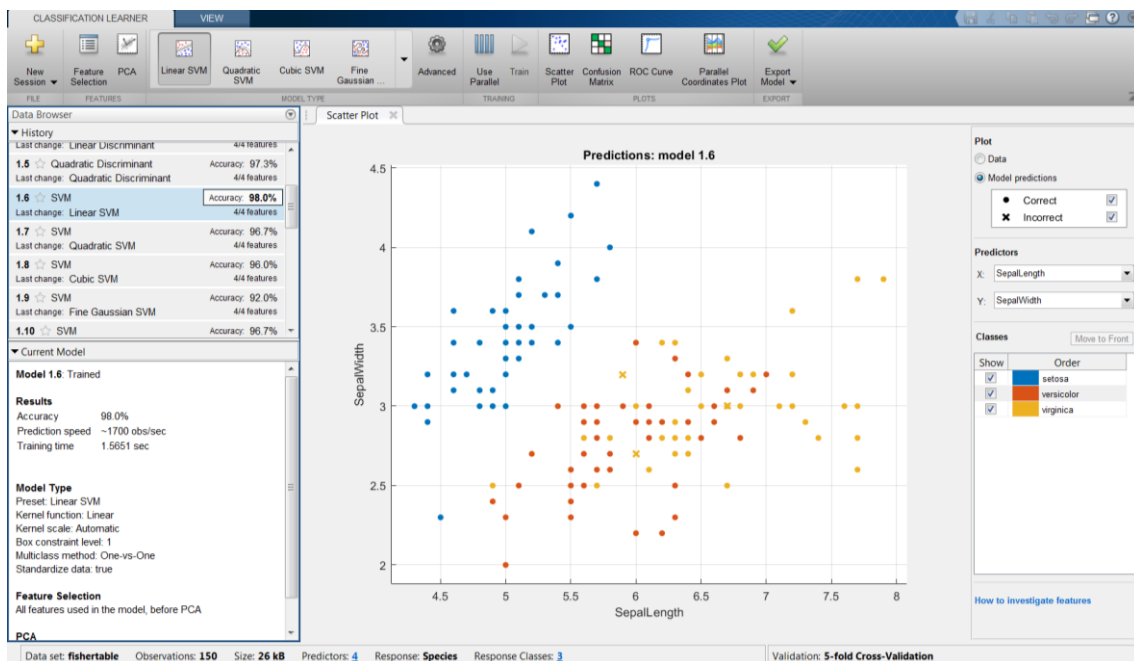
Η ίδια ακριβώς τεχνική υλοποιείται και στο επόμενο παραδείγμα που έχει σχέση με αριθμητική ταξινόμηση, με τη μόνη διαφορά πως για να μην επαναλαμβανόμαστε θα εμφανήσουμε μόνο τις εικόνες από τα αποτελέσματα που πήραμε.

## Fisher Iris

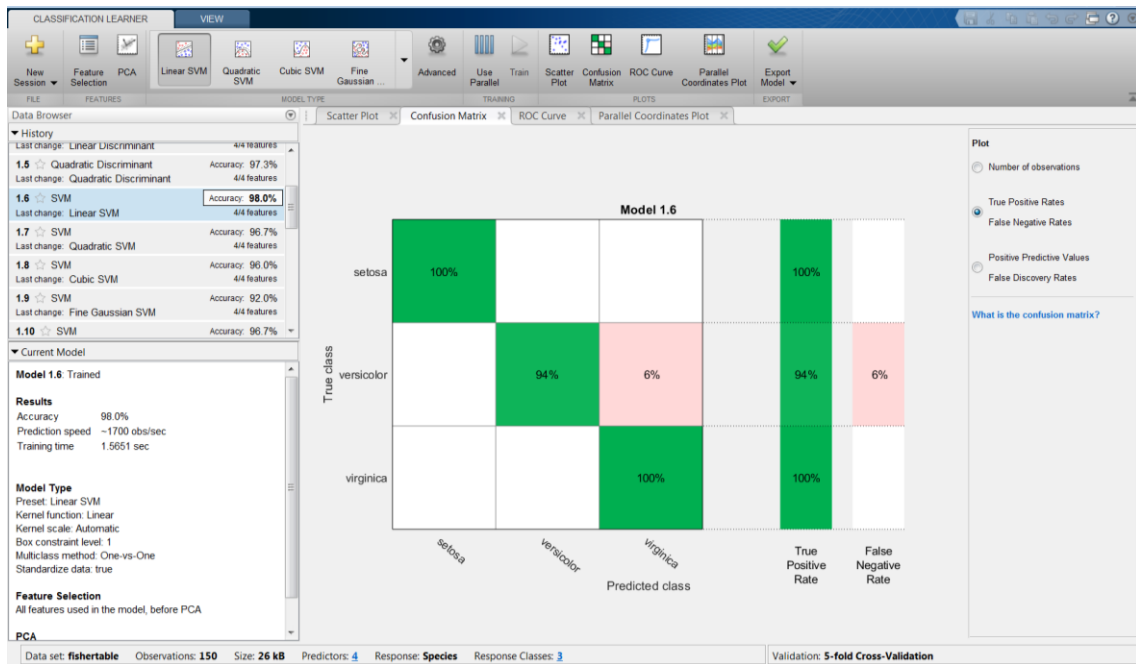
Εκμεταλλεύοντας λοιπόν το γεγονός ότι το Matlab διαθέτει μια μεγάλη γκάμα από παραδείγματα και δεδομένα έτοιμα προς χρήση θα επιλέξουμε να ασχοληθούμε με τα Fisher Iris και πληκτρολογώντας στο command window:

```
fishertable = readtable('fisheriris.csv');
```

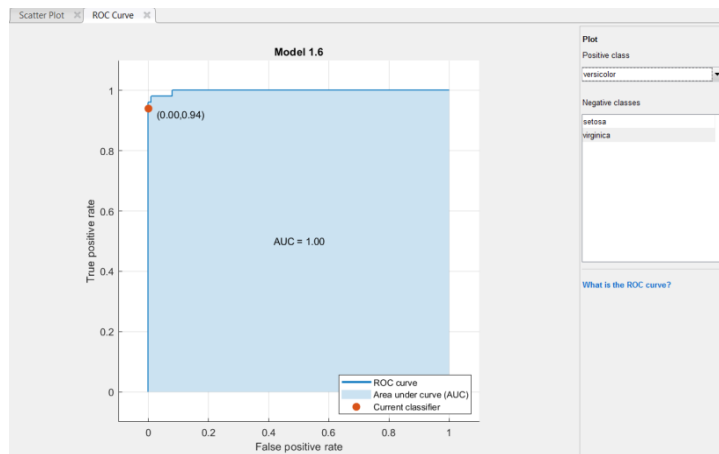
δημιουργουμε έτοιμο πίνακα δεδομένων στο Workspace.



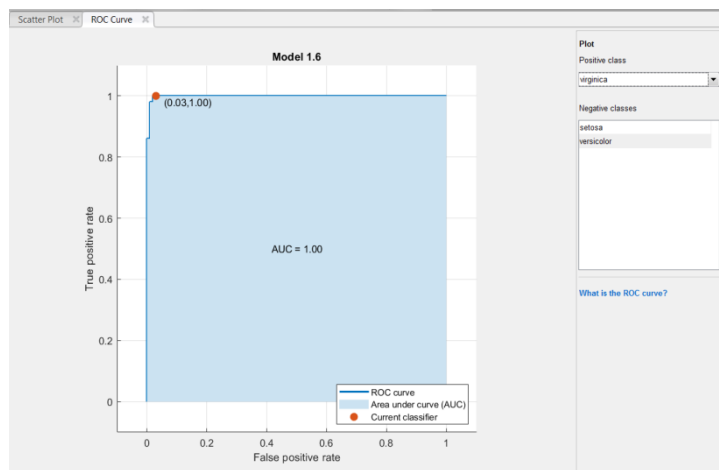
Εικόνα 3.13 Predictions Model



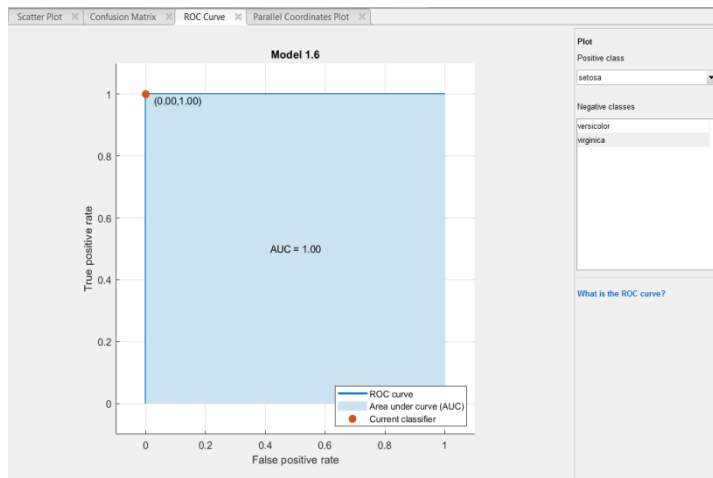
Εικόνα 3.14 Confusion matrix



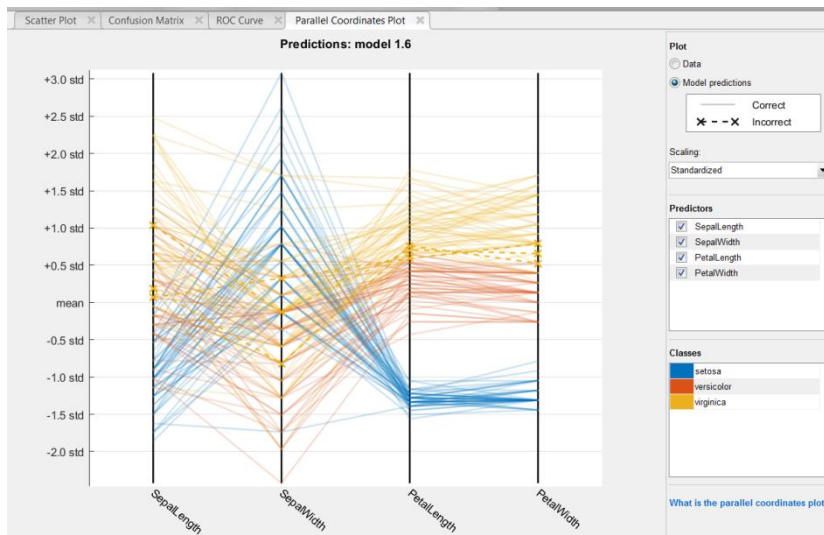
Εικόνα 3.15 ROC Curve



Εικόνα 3.16 ROC Curve



Εικόνα 3.17 ROC Curves



Εικόνα 3.18 Parallel Coordinates Plot

## Ταξινόμηση εικόνων (Image Classification)

Η Ψηφιακή Επεξεργασία Εικόνας είναι η επιστήμη που ασχολείται με την επεξεργασία ψηφιακών εικόνων, δηλαδή διδιάστατων συναρτήσεων  $f(x, y)$  με πεπερασμένο και διακριτό πλήθος στοιχείων. Κάθε ζεύγος  $(x, y)$  της συνάρτησης αντιστοιχεί σε ένα εικονοστοιχείο (pixel) και ορίζεται μία τιμή έντασης. Η επεξεργασία εικόνας ορίζει το σύνολο μεθόδων και αλγορίθμων που επιτρέπουν την επεξεργασία. Το βασικό εργαλείο κατανόησης του περιβάλλοντος για τον άνθρωπο αποτελεί η όραση, η οποία στηρίζεται κατά κύριο λόγο στην επεξεργασία των οπτικών ερεθισμάτων που καθημερινώς λαμβάνει. Η υπολογιστική όραση ή όραση υπολογιστών είναι η επιστήμη που περιλαμβάνει μεθόδους για την ανάλυση και κατανόηση των εικόνων μέσω της χρήσης ηλεκτρονικού υπολογιστή. Η όραση υπολογιστών σε συνεργασία με τη τεχνητή νοημοσύνη (A.I.) έχουν σκοπό την εκπαίδευση μοντέλων μάθησης στηριζόμενες σε οπτικό υλικό από εικόνες. Ένα κυρίαρχο πρόβλημα που η επιστήμη της όρασης υπολογιστών καλείται να απαντήσει, αποτελεί το κομμάτι της αναγνώρισης προτύπων σε ψηφιακές εικόνες. Ως αναγνώριση εικόνας θεωρούμε τη διαδικασία κατά την οποία δεδομένης μιας εικόνας ο υπολογιστής προσπαθεί να εντοπίσει τι αντικείμενα ή έννοιες που περιέχονται σε αυτήν.

Πιο συγκεκριμένα στην αναγνώριση εικόνας εμπίπτουν τα παρακάτω θεμελιώδη προβλήματα:

- Ο εντοπισμός και η αναγνώριση αντικειμένου σε εικόνα (Object Detection and Object Recognition)
- Η ανάκτηση εικόνων με βάση το περιεχόμενο (Content Based Image Retrieval - CBIR)
- Η ταξινόμηση εικόνων με βάση τις έννοιες που περιέχονται. (Image Classification)

Η πρώτη περίπτωση έχει ως σκοπό την εύρεση ενός ή περισσότερων αντικειμένων σε μια εικόνα, καθώς βασίζεται κυρίως σε μεθόδους που αφορούν την γεωμετρία του αντικειμένου

Η δεύτερη περίπτωση έχει ως στόχο την εύρεση και την ανάκτηση από μία βάση εικόνων που περιγράφουν το ίδιο αντικείμενο. Η δυσκολία της συγκεκριμένης κατηγορίας έγκειται στο γεγονός ότι στη βάση υπάρχει πληθώρα εικόνων, ενώ πρέπει να ανιχνευτούν σωστά μόνο όσες έχουν ενδιαφέρον προς το ερώτημα. Ένα παράδειγμα ανάκτησης εικόνων παρουσιάζεται σαν μια εικόνα, η οποία περιέχει τέσσερα διαφορετικά είδη εικόνων, "αεροπλάνο", "πλοίο", "αυτοκίνητο" και "ζώα". Από αυτό το σύνολο επιθυμούμε να ανακτήσουμε μόνο τις εικόνες με "αεροπλάνο", ενώ οι υπόλοιπες να μην επιστραφούν.

Η τρίτη περίπτωση που αποτελεί το κύριο θέμα της παρούσας εργασίας είναι η ταξινόμηση. Σκοπός της είναι η κατανόηση του περιεχομένου των φωτογραφιών, καθώς και η κατηγοριοποίησή τους.

### **Το πρόβλημα της ταξινόμησης εικόνων.**

Το πρόβλημα της ταξινόμησης εικόνων ορίζεται ως η διαδικασία σύμφωνα με την οποία υπάρχοντος ενός δεδομένου αριθμού εικόνων και ενός δεδομένου αριθμού διαφορετικών κατηγοριών, επιδιώκεται το σωστό ταίριασμα των εικόνων στις κατηγορίες που ανήκουν.

Προτού συνεχίσουμε στην διαδικασία της ταξινόμησης, ίσως ήταν χρήσιμο για την καλύτερη κατανόηση να αναρωτηθούμε το εξής: Ποιές κατηγορίες και με ποιό τρόπο μπορεί να αναγνωρίσει ο υπολογιστής; Η απάντηση στην ερώτηση αυτή έρχεται, όταν αναλογιστούμε τον τρόπο με το οποίο ο ανθρώπινος νους οργανώνει την πληροφορία σε διάφορα επίπεδα. Ένα άμεσο παράδειγμα φαίνεται στον τρόπο που ο άνθρωπος αναφέρεται σε ένα αντικείμενο όπως το αυτοκίνητο. Αρχικά μπαίνει στην διαδικασία της αναγνώρισής του λαμβάνοντας υπόψη ολόκληρο το σχήμα του, κάποιο μεμονωμένο χαρακτηριστικό του όπως οι πόρτες ή οι ρόδες ή και τη χρήση του για μεταφορά ανθρώπων. Έπειτα θα αναφερθεί σε αυτό είτε χρησιμοποιώντας την μάρκα, πχ «Honda», είτε απλά τον όρο «αυτοκίνητο». Παρόλες τις υπάρχουσες ορολογίες που μπορεί να χρησιμοποιήσει, η λέξη «αυτοκίνητο» έρχεται στο νου διότι είναι πιο απλή σε σχέση με το να θυμάται τη μάρκα. Συνεπώς ο καλύτερος τρόπος για οργάνωση της πληροφορίας, είναι να παραμείνει σε ένα "απλό" επίπεδο περιγραφής των χαρακτηριστικών της. Επιστρέφοντας στο πρόβλημα της ταξινόμησης πρέπει να βρούμε ποιά απλά χαρακτηριστικά χρειάζεται ένας υπολογιστής, ώστε να εκτελέσει με επιτυχία την ταξινόμηση. Σε αντίθεση με το πρόβλημα του εντοπισμού των αντικειμένων, που στηρίζεται στη μοναδική περιγραφή ενός αντικειμένου, στην ταξινόμηση θα χρειαστούμε περισσότερη πληροφορία από τα χαρακτηριστικά που προσφέρει ένα αντικείμενο. Άρα κρίνεται απαραίτητη η εξαγωγή χαρακτηριστικών από πολλές εικόνες. Όσο όμως αυξάνεται ο αριθμός των εικόνων, τόσο αυξάνεται και το πλήθος των διανυσμάτων που χρειάζονται στην περιγραφή. Αυτό συμβαίνει καθώς εικόνες που ενδέχεται να απεικονίζουν το ίδιο αντικείμενο, το απεικονίζουν με διαφορετικούς τρόπους, δηλαδή με διαφορές στο φωτισμό, την οπτική γωνία, χρώματα ή ακόμα το παρουσιάζουν σε ένα σκηνικό μαζί με άλλα αντικείμενα. Συνεπώς όσο πιο πολλά αντικείμενα περιγράφουμε, τόσο μεγαλώνει ο όγκος της πληροφορίας, με αποτέλεσμα να ξεφεύγουμε απ' το απλό επίπεδο περιγραφής. Έτσι είναι αναγκαία η δημιουργία κάποιου μοντέλου, που βασίζεται στα χαρακτηριστικά,



αποσκοπώντας αφενός στη μείωση του όγκου των δεδομένων και αφετέρου στην πληρέστερη περιγραφή του συνόλου. Με το σύνολο των διανυσματικών τιμών που προκύπτουν από την επεξεργασία των εικόνων, μας δίνεται η δυνατότητα εκπαίδευσης ενός στατιστικού μοντέλου, στο οποίο βασιζόμαστε ώστε να προβλέψουμε την κατηγορία κάθε εικόνας. Ένας τρόπος είναι να κατασκευαστεί το μοντέλο αυτό μέσω μίας διαδικασίας επιβλεπόμενης μάθησης, (κατηγορία της μηχανικής μάθησης) κατά την οποία χρησιμοποιώντας ένα σύνολο δεδομένων ως είσοδο για την εκπαίδευση ενός μοντέλου μπορούμε να προβλέψουμε την έξοδο.

### Σύγκριση και ταξινόμηση

Είναι ενδιαφέρον πώς μπορούμε να αναγνωρίσουμε ανθρώπους, ζώα. Είναι ακόμα πιο ενδιαφέρουσα η ανάπτυξη ενός μαθηματικού αλγορίθμου που μιμείται τον μηχανισμό αναγνώρισης εικόνας. Αρχικά, ας δούμε πώς το κάνουμε αυτό βιολογικά. Στην καθημερινότητά μας, παίρνουμε μια τεράστια ποσότητα πληροφοριών από πολλές πηγές γύρω μας. Στη συνέχεια, διαμορφώνουμε μια προκαταρκτική αντίληψη. Για να αναγνωρίσουμε οπτικά τις εικόνες, συγκρίνουμε τις εικόνες με τα γνωρίσματα που έχουμε συγκεντρώσει μέσα από την εμπειρία μας.

Ας δούμε τις παρακάτω εικόνες για παράδειγμα. Μπορούμε εύκολα να πούμε ότι είναι εικόνες σκύλου και γάτας, διότι ήδη έχουμε ένα σύνολο εικόνων από αυτά τα δύο ζώα στο μυαλό μας.



Οι ακμές των εικόνων παίρνουν ένα σημαντικό μέρος της δικής μας βιολογικής διαδικασίας αναγνώρισης εικόνων. Σε αυτό το κεφάλαιο, αναπτύσσουμε έναν μαθηματικό αλγόριθμο για να αναγνωρίσουμε τις εικόνες των γατών και των σκύλων, ακριβώς όπως το μυαλό μας κάνει να αναγνωρίζει τις εικόνες. Μας δίνονται με ένα εκπαιδευτικό σύνολο 6000 εικόνων γατών

και 6000 εικόνων σκυλιών ώστε να εκπαιδύσουμε τον υπολογιστή για να μάθουμε πως θα πρέπει να μοιάζει μια γάτα και ένας σκύλος.

Εν κατακλείδι, ο στόχος μας είναι να καταστήσουμε τον υπολογιστή σε διαδικασία αναγνώρισης των επιλεγμένων τυχαίων εικόνων ως γάτα ή σκύλο, συγκρίνοντας αυτές τις άγνωστες εικόνες με το σύνολο εκπαίδευσης και εξάγοντας και ποσοστό επιτυχίας.

Το μαθηματικό μοντέλο που αντλεί δεδομένα από το σύνολο εκπαίδευσης είναι το Bag of Features.

### **Αναπαράσταση του Bag of Features:**

Οι αναπαραστάσεις αντικειμένων που βασίζονται σε συλλογές αναλλοίωτων περιγραφικών στοιχείων που εξάγονται από τοπικές ετικέτες εικόνας έχουν γίνει πολύ δημοφιλείς. Στην υπολογιστική όραση, τα τοπικά χαρακτηριστικά της εικόνας παίζουν σημαντικό ρόλο σε εφαρμογές όπως η ανάλυση υφής, η ανάκτηση εικόνων με βάση το περιεχόμενο και η αναγνώριση αντικειμένων. Οι τοπικές λειτουργίες εξάγονται για να δημιουργήσουν ένα διάλυμα χαρακτηριστικών που παρέχει μια αναπαράσταση που επιτρέπει την κατάλληλη αντιστοίχιση των τοπικών δομών μεταξύ των εικόνων. Ένας καλός περιγραφέας έχει μερικά σημαντικά χαρακτηριστικά, συμπεριλαμβανομένης της αναλλοίωσής του στο φωτισμό, την περιστροφή και την κλίμακα. Επιπλέον, θα πρέπει να είναι ανθεκτικό σε σφάλματα ανίχνευσης, γεωμετρικές παραμορφώσεις από διαφορετικές οπτικές γωνίες και φωτομετρικές παραμορφώσεις που προκαλούνται από διαφορετικές συνθήκες φωτισμού.

Για την αναγνώριση αντικειμένων, η μέθοδος Bag of Features έχει κερδίσει την προσοχή από τους ερευνητές στην όραση στον υπολογιστή.

Μια μέθοδος Bag of Features είναι αυτή που αντιπροσωπεύει τις εικόνες ως συλλογές τοπικών χαρακτηριστικών. Το όνομα προέρχεται από την αναπαράσταση Bag of Words που χρησιμοποιείται στην ανάκτηση πληροφοριών κειμένου. Υπάρχουν δύο κοινές προοπτικές για να εξηγηθεί η αναπαράσταση εικόνας του BoF.

Η αναπαράσταση της εικόνας του Bag of Features είναι ανάλογη. Ένα οπτικό λεξιλόγιο κατασκευάστηκε για να αντιπροσωπεύει το λεξικό συγκεντρώνοντας χαρακτηριστικά που εξάγονται από ένα σύνολο εικόνων. Τα χαρακτηριστικά εικόνας αντιπροσωπεύουν τοπικές περιοχές της εικόνας, ακριβώς όπως οι λέξεις είναι τοπικά χαρακτηριστικά ενός εγγράφου.

Απαιτείται ομαδοποίηση, ώστε να μπορεί να δημιουργηθεί ένα διακριτό λεξιλόγιο από εκατομμύρια (ή δισεκατομμύρια) τοπικών χαρακτηριστικών από τα δεδομένα εκπαίδευσης. Κάθε σύμπλεγμα χαρακτηριστικών είναι μια οπτική λέξη. Με μια νέα εικόνα, τα χαρακτηριστικά ανιχνεύονται και αποδίδονται στους πλησιέστερους όρους ταύτισής τους (κέντρα συμπλέγματος) από το οπτικό λεξιλόγιο.

Η ανίχνευση χαρακτηριστικών είναι η διαδικασία λήψης αποφάσεων για το πού και σε ποια κλίμακα θα γίνει δειγματοληψία μιας εικόνας. Η έξοδος ανίχνευσης χαρακτηριστικών είναι ένα σύνολο σημείων-κλειδιών που καθορίζουν θέσεις στην εικόνα με αντίστοιχες κλίμακες και προσανατολισμούς. Αυτά τα σημεία-κλειδιά είναι διαφορετικά από τους περιγραφείς χαρακτηριστικών, τα οποία κωδικοποιούν πληροφορίες από τα εικονοστοιχεία που βρίσκονται κοντά στα σημεία-κλειδιά.

### *Ανιχνευτές χαρακτηριστικών*

Υπάρχουν διάφοροι αλγόριθμοι για την επιλογή χαρακτηριστικών από εικόνες. Οι υποστηριζόμενοι ανιχνευτές χαρακτηριστικών είναι SIFT, SURF, GFTT, FAST, ORB, BRISK, STAR και MSER.

- Δείγματα κατάρτισης:

Πριν μπορεί να χρησιμοποιηθεί ένας ταξινομητής, πρέπει να εκπαιδευτεί με αρκετά δείγματα των αντικειμένων στόχων. Ως εκ τούτου, δημιουργήθηκε βάση δεδομένων κατάρτισης χρησιμοποιώντας το λεξιλόγιο των οπτικών λέξεων που υπολογίστηκαν νωρίτερα.

Σε αυτό το στάδιο, κάθε εικόνα της λίστας εκπαιδευτικών είναι προεπεξεργασμένη, τα σημεία χαρακτηριστικών της υπολογίζονται και διαχωρίζονται στις αντίστοιχες τάξεις και στη συνέχεια οι περιγραφείς κατασκευάζονται χρησιμοποιώντας το οπτικό λεξιλόγιο.

Τα αποτελέσματα είναι ένα σύνολο ομαλοποιημένων ιστογραμμάτων των οπτικών λέξεων που υπάρχουν σε κάθε εικόνα προπόνησης, που σχετίζονται με τις αντίστοιχες ετικέτες, που θα ενημερώνουν τον ταξινομητή στον οποίο ταξινομούνται τα δείγματα εκπαίδευσης.

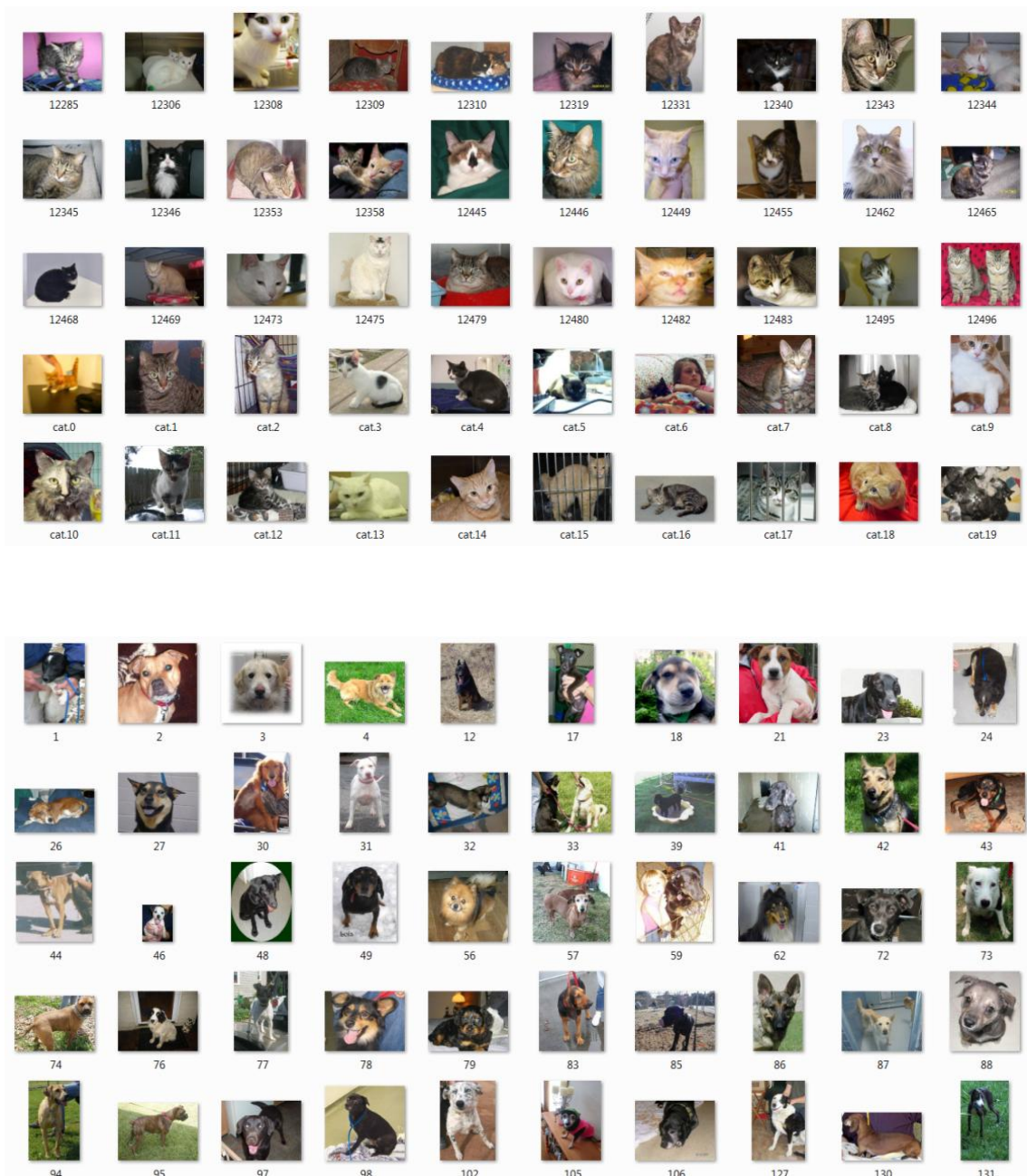
- Εκπαίδευση ταξινομητή:

Μετά τη λήψη των δειγμάτων κατάρτισης, ένας εκπαιδευτής ταξινομείται για να οικοδομήσει ένα μοντέλο της κατανομής των περιγραφέων οπτικών λέξεων αντικειμένων στόχου. Αυτό το μοντέλο μπορεί στη συνέχεια να χρησιμοποιηθεί για την πρόβλεψη με αποδεκτή ακρίβεια αν τα αντικείμενα-στόχοι είναι σε μια εικόνα ή όχι.

Υπάρχουν διάφοροι ταξινομητές εκμάθησης μηχανών για την εκτέλεση αναγνώρισης αντικειμένων. Οι συμπεριλαμβανόμενοι ταξινομητές είναι οι Support Vector Machines, Artificial Neural Networks, Normal Bayes Classifier, Decision Trees, Gradient Boosting Trees, Random Trees and Extremely Randomized Trees.

### Παράδειγμα Dog & cat

Τα δεδομένα που χρησιμοποιήθηκαν έχουν ληφθεί από διαφορετικές πηγές οι οποίες αναφέρονται αναλυτικά στη βιβλιογραφία. Βρήκαμε λίγο παραπάνω από 6000 εικόνες για το κάθε είδος, οι οποίες χωρίστηκαν σε διαφορετικούς φακέλους με τις αντίστοιχες ονομασίες.



```

%% Load image data
clear all
close all
clc
imds =
imageDatastore('C:\Users\user\Desktop\cat&dog_6000+6000\',...
'IncludeSubfolders',true,'LabelSource','foldernames')
imds.ReadFcn =
@readAndResizeImages;

```

```

%% Display Class Names and Counts
tbl = countEachLabel(imds)
categories = tbl.Label;

```

```

%% Display Sampling of Image Data
visImds =
splitEachLabel(imds,1,'randomize')
;

for ii = 1:2 % this assumes 2
categories of pet
    subplot(2,1,ii);
    imshow(visImds.readimage(ii));

title(char(visImds.Labels(ii)));
end

```

Χρησιμοποιούμε τις 3 πρώτες γραμμές για να καθαρίσουμε όλες τις προηγούμενες εντολές και το ιατρικό

Έπειτα μέσω αυτής της εντολής μπορούμε και διαβάζουμε όλες τις εικόνες της διεύθυνσης αυτόματα και μας δίνεται η δυνατότητα να εξάγουμε τους υποφακέλους και τις τους αντίστοιχα.

Καλούμε την συνάρτηση που έχουμε δημιουργήσει πιο κάτω για να εφαρμόσουμε σε όλες τις εικόνες το ίδιο μέγεθος.

---

Εδώ εμφανίζουμε σε ένα πίνακα τις ετικέτες και την αντίστοιχη ποσότητα εικόνων.

---

Δημιουργούμε δύο νέες βάσεις δεδομένων από τα αρχεία τραβώντας τυχαία από κάθε ετικέτα. Το πρώτο figure περιέχει ένα τυχαίο αρχείο με την ετικέτα cat και το figure2 ένα τυχαίο αρχείο με την ετικέτα dog.



```

%% Pre-process Training Data:
*Feature Extraction using Bag Of
Words*
%% Partition 5000 images for
training and 1000 for testing
[training_set, test_set] =
imds.splitEachLabel(5000,1000,'ran
domize',true);

```

```

%% Create Visual Vocabulary
tic
bag =
bagOfFeatures(training_set,...

'VocabularySize',250,'PointSelecti
on','Detector');
scenedata = double(encode(bag,
training_set));
toc

```

```

%% Visualize Feature Vectors
imgs =
training_set.splitEachLabel(1,'ran
domize',true);

```

```

img = readimage(imgs,1);
featureVector = encode(bag, img);
subplot(2,2,1); imshow(img);
subplot(2,2,2);
bar(featureVector);title('Visual
Word Occurrences');xlabel('Visual
Word Index');ylabel('Frequency');

```

```

img = readimage(imgs,2);
featureVector = encode(bag, img);
subplot(2,2,3); imshow(img);
subplot(2,2,4);
bar(featureVector);title('Visual
Word Occurrences');xlabel('Visual
Word Index');ylabel('Frequency');

```

Εκχωρεί τυχαία το καθορισμένο ποσό αρχείων από κάθε ετικέτα στις νέες βάσεις δεδομένων training\_set & test\_set. Ουσιαστικά έχουμε επιλέξει 5000 εικόνες από κάθε κατηγορία για εκπαίδευση του προγράμματος και 1000 εικόνες για να ελέγξουμε την εγκυρότητα, φυσικά γίνεται τυχαία επιλογή.

Επιλογή χαρακτηριστικών σημείων λειτουργίας χρησιμοποιώντας τη μέθοδο ανίχνευσης, καθώς και εξαγωγή χαρακτηριστικών SURF από τις επιλεγμένες θέσεις σημείων λειτουργίας (μόλις τελειώσει αυτή η εντολή θα έχουμε και την διάρκεια της επεξεργασίας ως αποτέλεσμα). Τέλος φτιάχνουμε ένα πίνακα με τα χαρακτηριστικά τω εικόνων.

Στο σημείο αυτό εξάγουμε από τυχαία επιλογή τα 250 χαρακτηριστικά που έχει ξεχωρίσει το bagOfFeatures για μία εικόνα από κάθε κατηγορία ώστε να μπορούμε να δούμε τις φανερές διαφορές που υπάρχουν μεταξύ τους, μέσω των γραφημάτων τους.

```

%% Create a Table using the
encoded features
SceneImageData =
array2table(scenedata);
SceneImageData.sceneType =
training_set.Labels;

%% Use the new features to train a
model and assess its performance
using
classificationLearner

%% Test out accuracy on test set!

testSceneData = double(encode(bag,
test_set));
testSceneData =
array2table(testSceneData, 'VariableNames', trainedModel.RequiredVariables);
actualSceneType = test_set.Labels;
predictedOutcome =
trainedModel.predictFcn(testSceneData);
correctPredictions =
(predictedOutcome ==
actualSceneType);
validationAccuracy =
sum(correctPredictions)/length(predictedOutcome) %#ok

```

Μετατροπή της μορφής των δεδομένων σε πίνακα και εμφανίζουμε σαν τελευταία τιμή τις ετικέτες των εικόνων (cat - dog), δηλαδή έχουμε τις 250 τιμές και σαν 251<sup>η</sup> τιμή εμφανίζουμε την αντίστοιχη ονομασία. Αυτή η σταθερά είναι απαραίτητη για την ταξινόμηση και την αναγνώριση σαν τελικό στάδιο.

Ακολούθως ανοίγουμε το classification learner και ακολουθούμε τα βήματα που υπάρχουν στη συνέχεια του κεφαλαίου.

---

Έχουμε φτάσει στο σημείο να ελέγξουμε την ακρίβεια του τεστ που πραγματοποιούμε στις εικόνες για την πρόβλεψη του αποτελέσματος.

Πιο συγκεκριμένα φέρνουμε τα χαρακτηριστικά στη μορφή που επιθυμούμε και έπειτα με βάση το μοντέλο που έχουμε εξάγει παίρνουμε το αποτέλεσμα της πρόβλεψης ένα-ένα ξεχωριστά, αλλά και σαν ποσοστό επιτυχίας στο τέλος (ValidationAccuracy)

```

%% Visualize how the classifier
works
figure(2);
random_num =
randi(length(test_set.Labels));

img =
test_set.readimage(random_num);

imshow(img)
imagefeatures = double(encode(bag,
img));

% Find two closest matches for
each feature
[bestGuess, score] =
predict(trainedModel.Classificatio
nSVM,imagefeatures);

% Display the string label for img
if bestGuess ==
test_set.Labels(random_num)
    titleColor = [0 0.8 0];
else
    titleColor = 'r';
end
title(sprintf('Best Guess: %s;
Actual: %s',...

char(bestGuess),test_set.Labels(ra
ndom_num)),...
    'color',titleColor)

function img =
readAndResizeImages(filename)

% read image
im = imread(filename);

% resize image
img = imresize(im,[200 200]);
end

```

Τέλος εμφανίζουμε το γραφικό περιβάλλον της πρόβλεψης, εκεί όπου συγκρίνουμε το αποτέλεσμα του τεστ με το πραγματικό label της εικόνας.

Εμφανίζουμε την τυχαία εικόνα και επιλέγουμε να εμφανίσουμε με πράσινους χαρακτήρες εάν είναι σωστό το αποτέλεσμα της πρόβλεψης και με κόκκινους εάν είναι λάθος η πρόβλεψη.


Στον τίτλο εμφανίζονται οι πραγματικές ετικέτες που εξάγονται μέσω των labels και δίπλα είναι η ετικέτα κατά προσέγγιση.

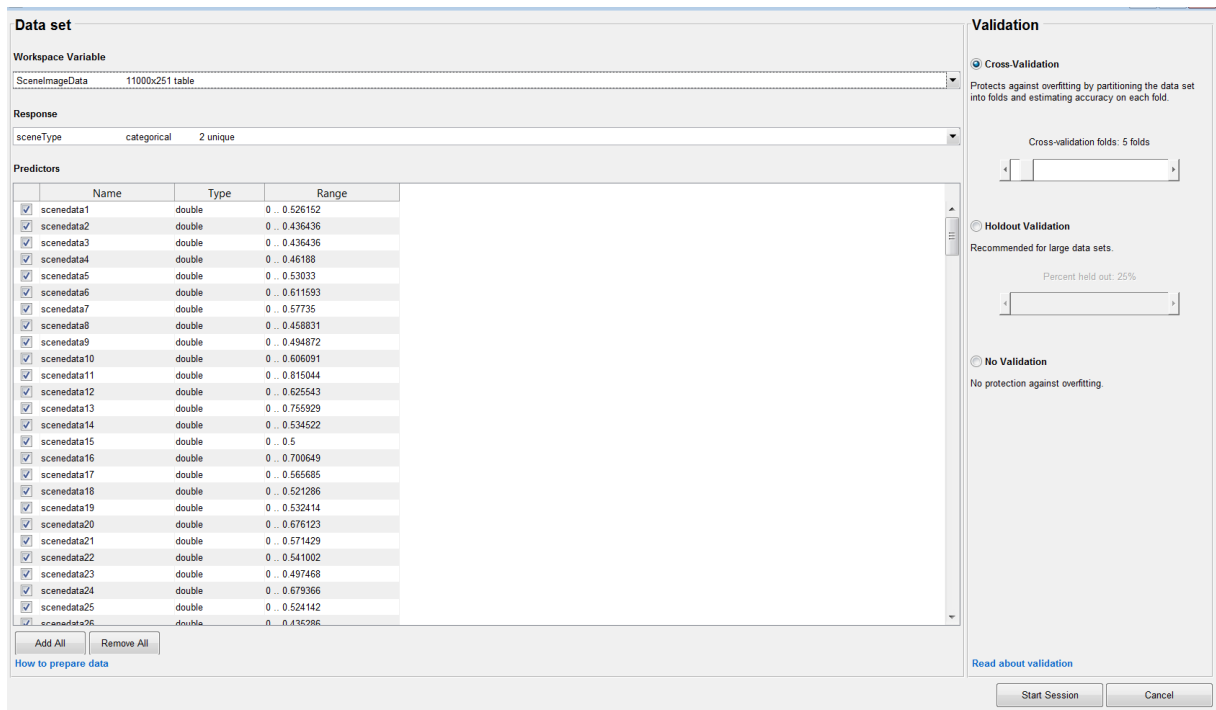
---

Συνάρτηση, η οποία καλείται μέσω του κύριου προγράμματος και εκτελεί αλλαγή στο μέγεθος όλων των εικόνων. Αποτελεί το κύριο βήμα για την επεξεργασία και άντληση των χαρακτηριστικών.



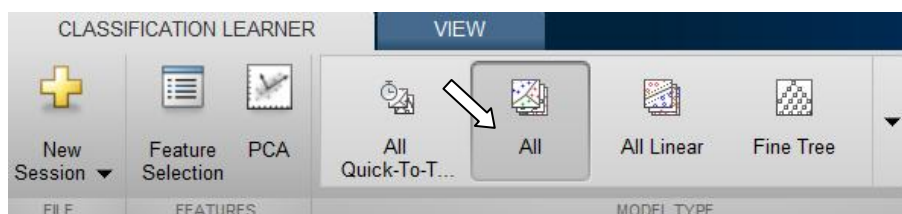
## Γραφικό περιβάλλον Classification Learner

Επιλέγουμε  και σαν δεδομένα για την ταξινόμηση επιλέγουμε το SceneImageData.

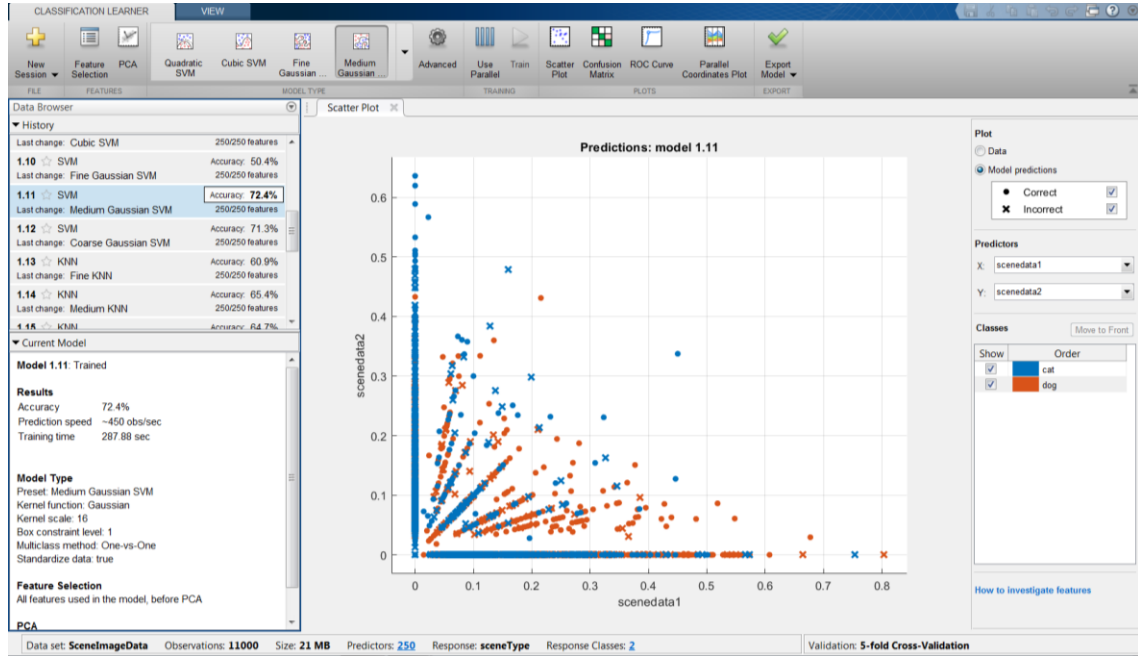


Εικόνα 3.19 Πίνακας Dataset

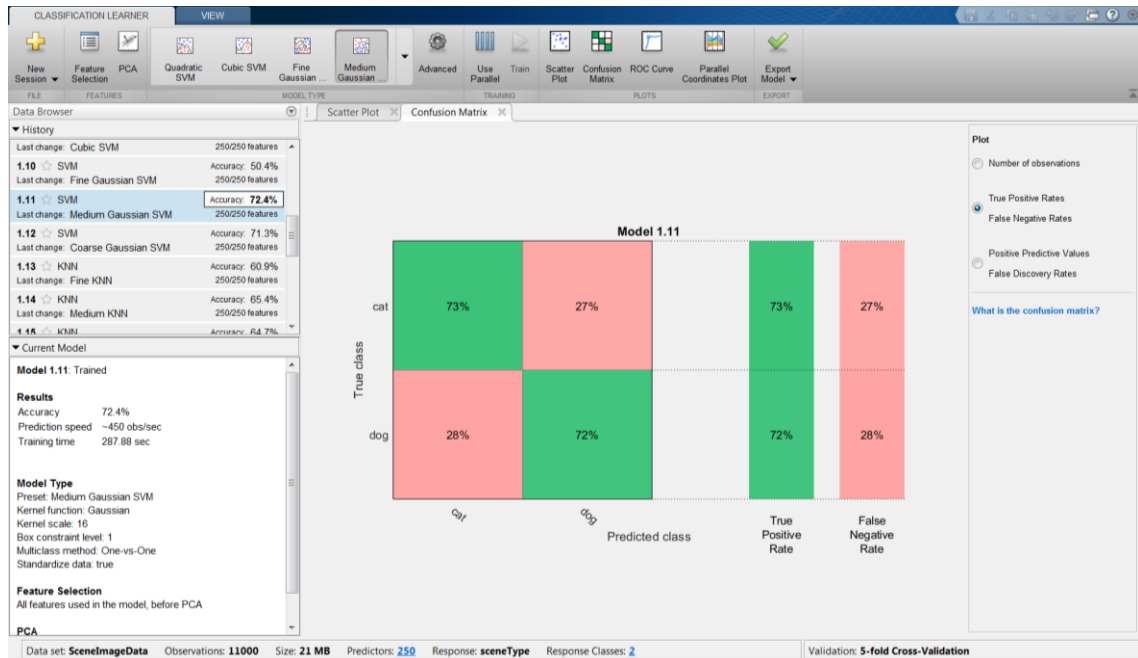
Στην καρτέλα Classification Learner, στην ενότητα Type Model, επιλέγουμε να γίνει ταξινόμηση με όλους τους ταξινομητές και στη συνέχεια πατάμε το **Train**



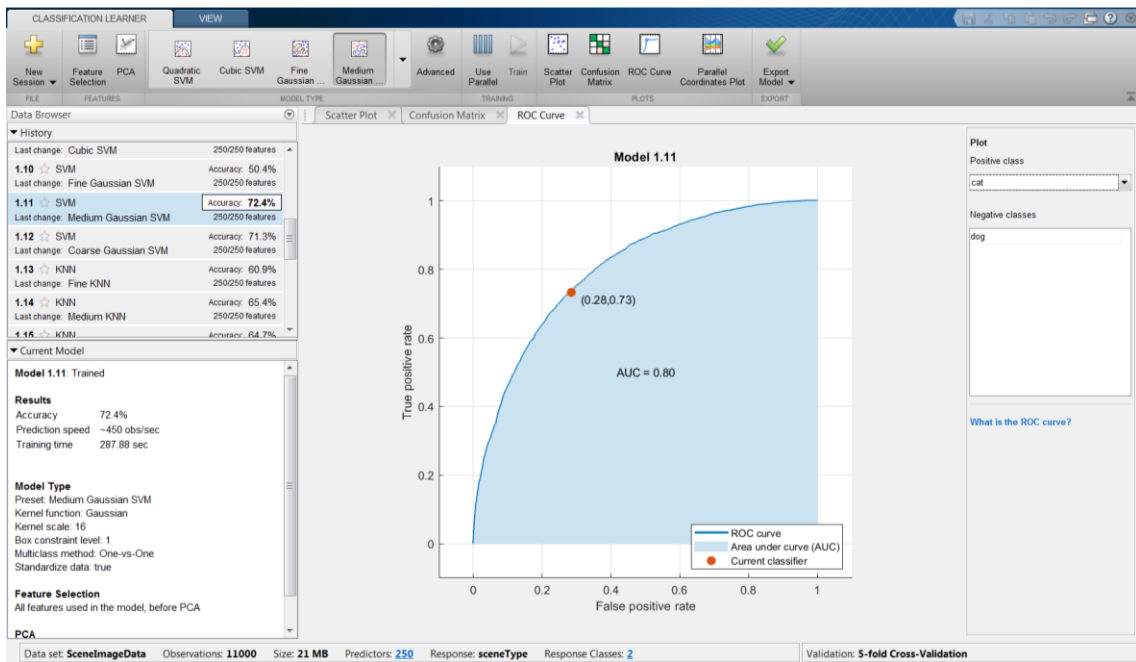
Η επιλεγόμενοι ταξινομητές εμφανίζονται στη λίστα Ιστορικό. Όταν τελειώσουν την εκπαίδευση, το καλύτερο ποσοστό Ακρίβειας επισημαίνεται σε ένα κουτί.



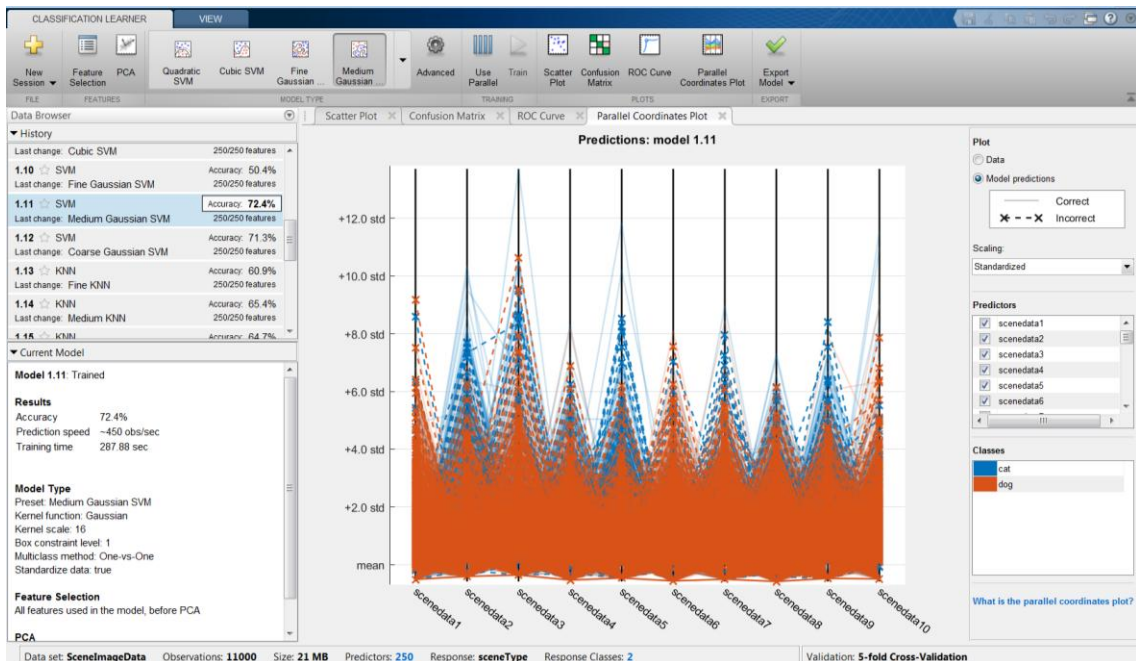
Εικόνα 3.20 Scatter plot



Εικόνα 3.21 Confusion Matrix

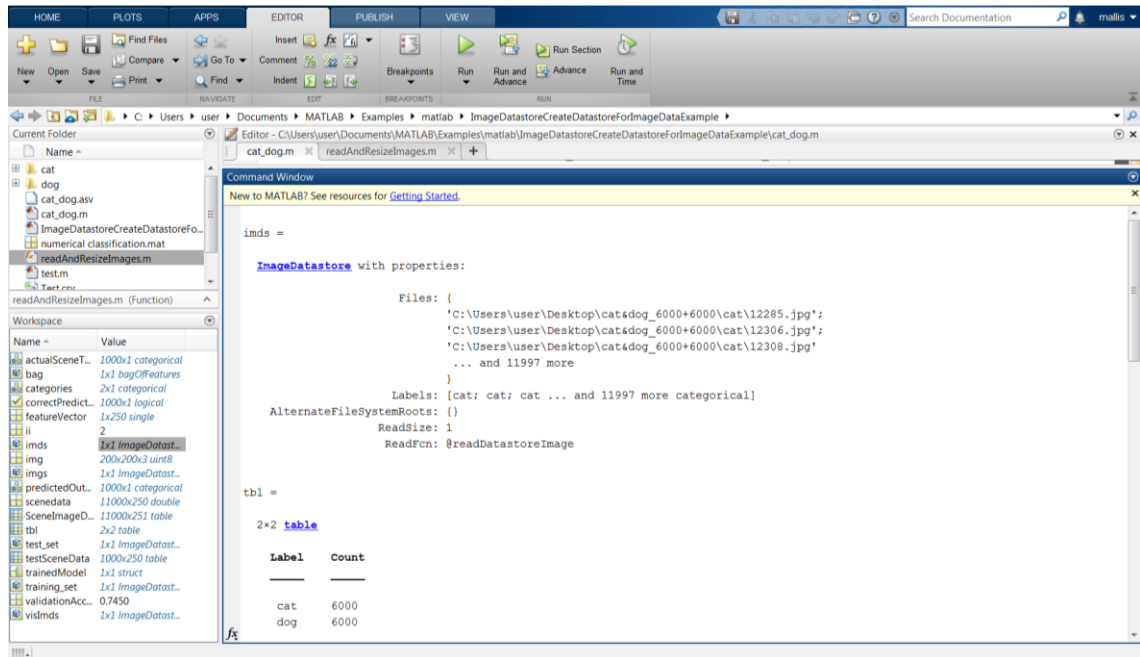


Εικόνα 3.22 ROC curve

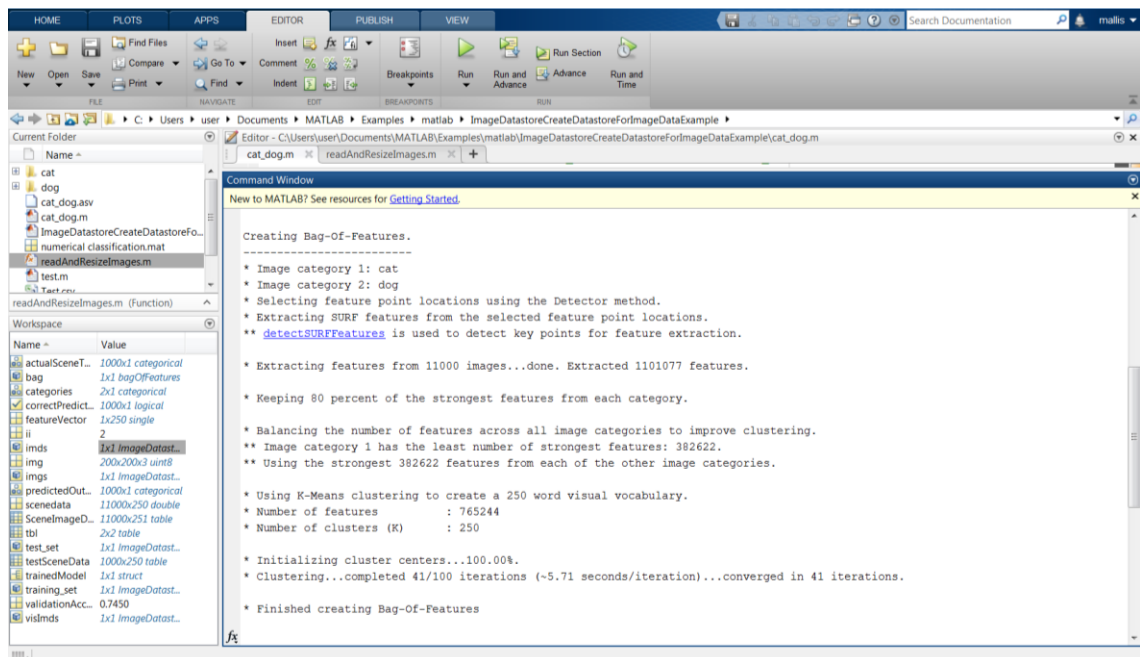


Εικόνα 3.23 Parallel Coordinates plot

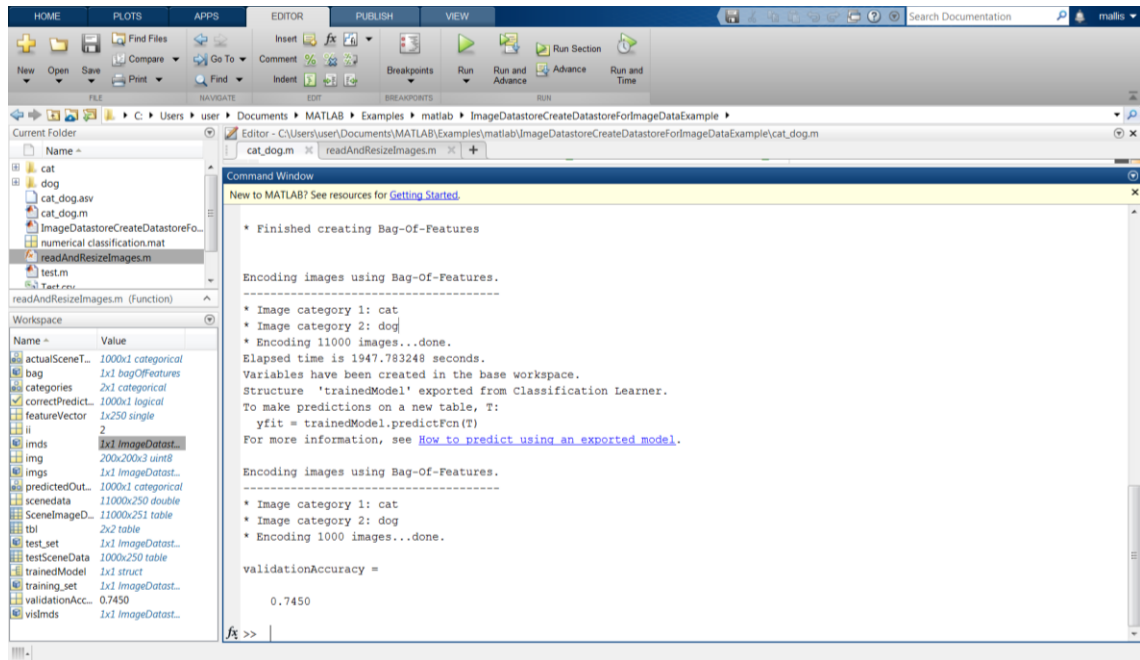
Ακολουθούν εικόνες από το command window κατά την διάρκεια εκτέλεσης του κώδικα.



Εικόνα 3.24 Ροή εκτέλεσης εντολών 1



Εικόνα 3.25 Ροή εκτέλεσης εντολών 2

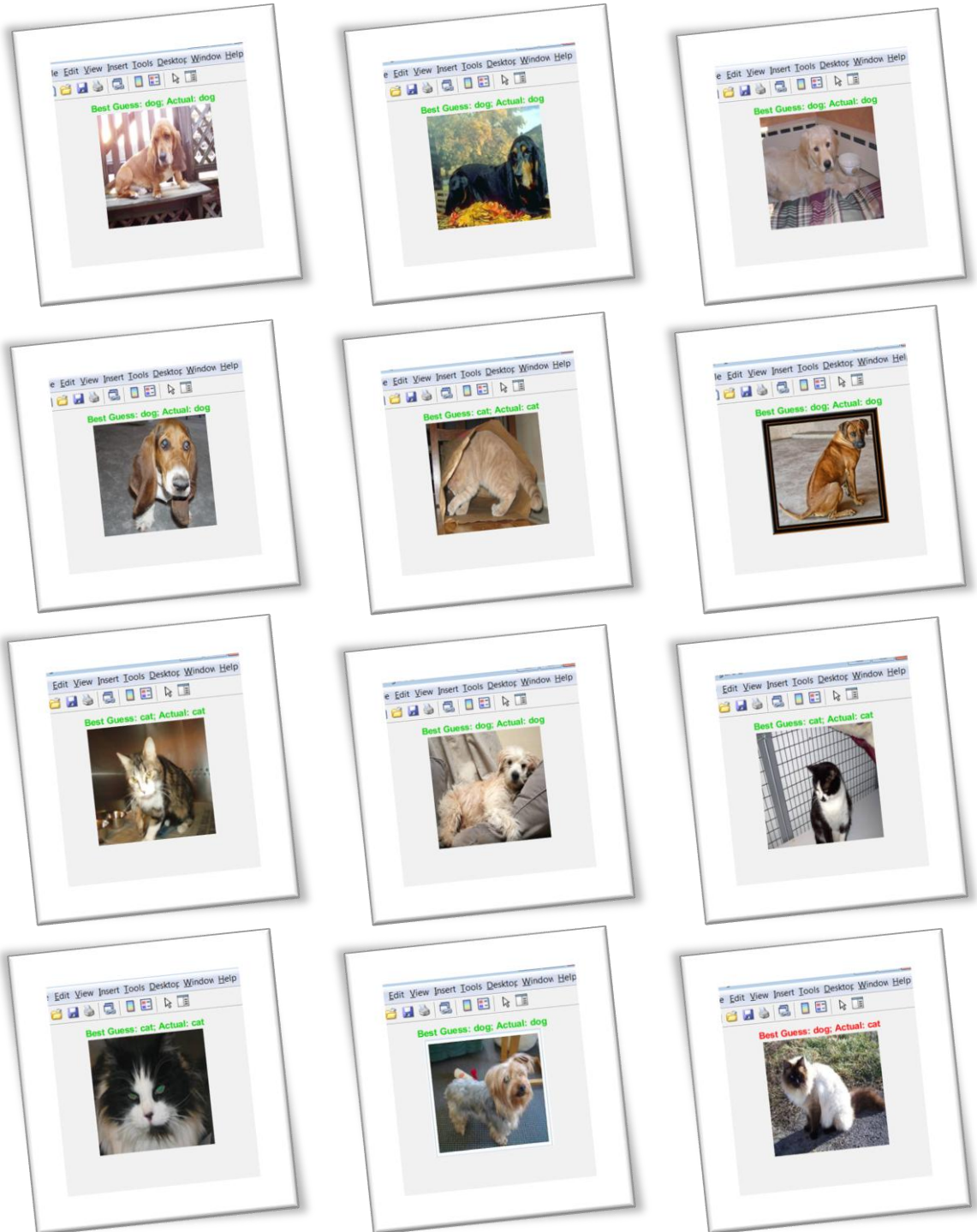


Εικόνα 3.26 Ροή εκτέλεσης εντολών 3

Το αποτέλεσμα εγκυρότητας της αναγνώρισης είναι σχετικά καλό ποσοστό, βεβαίως έχει περιθώρια βελτίωσης. Ένας τρόπος είναι να επιλέξουμε καλύτερες εικόνες με όσο γίνεται δυνατό χαμηλότερο θόρυβο κατά την άντληση δεδομένων, καθώς σε αρκετές εικόνες παρατηρείται το φαινόμενο να υπάρχουν μπροστά από μια γάτα η σκύλο ένας άνθρωπος να ένα άλλο αντικείμενο. Επίσης μπορούμε μέσω του Advance settings να αυξήσουμε λίγες ποσοστιαίες μονάδες το αποτέλεσμα.

## Αποτελέσματα

Στην συνέχεια έχουμε εμφανίσει εικόνες στις οποίες έγινε αναγνώριση μεσω του κώδικα για το δίλλημα αν είναι γάτα η σκύλος το εικονιζόμενο ζώο.



Όπως παρατηρούμε έχουμε μία αστοχία στις 12 φωτογραφίες που δώσαμε για αναγνώριση.

## Συμπεράσματα

Σε αυτή τη διπλωματική εργασία υλοποιήσαμε ένα σύστημα αναγνώρισης προτύπων, για την ταξινόμηση διαφόρων εικόνων σε κατηγορίες. Πειραματιστήκαμε με διάφορες κατηγορίες και με διαφορετικό πλήθος εικόνων κάθε φορά, μέχρι να φτάσουμε στα αποτελέσματα που παρουσιάσαμε στο κεφάλαιο 3.

Επίσης, ο θόρυβος που περιελάμβαναν οι εικόνες ή ακόμα και το διαφορετικό φόντο που είχε η κάθε μια, έδωσαν έναν ρεαλιστικό τόνο στο πείραμά μας.

Η εκτέλεση του πειράματός μας, μας οδήγησε σε ποσοστά μέσης ακρίβειας βάση των οποίων μπορούμε να επιλέξουμε ποιοι από τους αλγορίθμους μας τα πήγαν καλύτερα.

Σίγουρα η βελτίωση των ποσοστών ακρίβειας καθώς και η προσπάθεια ταξινόμησης όλο και περισσότερων εικόνων μπορεί να αποτελέσει προϊόν μελλοντικής έρευνας και μελέτης.



## Βιβλιογραφία

- [1] [http://laonikos13galanis.blogspot.com/2013/12/blog-post\\_6756.html](http://laonikos13galanis.blogspot.com/2013/12/blog-post_6756.html), προσπέλαση στις 5/4/2018
- [2] <http://users.sch.gr/jenyk/index.php/artificialintelligence/ai-historicalreview/5-historicalroute>, προσπέλαση στις 1/4/2018
- [3] [https://el.wikipedia.org/wiki/%CE%A4%CE%B5%CF%87%CE%BD%CE%B7%CF%84%CE%AE\\_%CE%BD%CE%BF%CE%B7%CE%BC%CE%BF%CF%83%CF%8D%CE%BD%CE%B7](https://el.wikipedia.org/wiki/%CE%A4%CE%B5%CF%87%CE%BD%CE%B7%CF%84%CE%AE_%CE%BD%CE%BF%CE%B7%CE%BC%CE%BF%CF%83%CF%8D%CE%BD%CE%B7), προσπέλαση στις 7/4/2018
- [4] [https://el.wikipedia.org/wiki/%CE%9C%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CE%BA%CE%AE\\_%CE%BC%CE%AC%CE%B8%CE%B7%CF%83%CE%B7](https://el.wikipedia.org/wiki/%CE%9C%CE%B7%CF%87%CE%B1%CE%BD%CE%B9%CE%BA%CE%AE_%CE%BC%CE%AC%CE%B8%CE%B7%CF%83%CE%B7), προσπέλαση στις 2/2/2018
- [5] <https://translate.google.gr/translate?hl=el&sl=en&u=https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72&prev=search>, προσπέλαση στις 2/2/2018
- [6] <https://www.saedsayad.com/classification.htm>, προσπέλαση στις 5/2/2018
- [7] <https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29>, προσπέλαση στις 18/2/2018
- [8] <https://www.mathworks.com/help/stats/select-data-and-validation-for-classification-problem.html>, προσπέλαση στις 22/4/2018
- [9] <https://www.mathworks.com/help/stats/train-classification-models-in-classification-learner-app.html>, προσπέλαση στις 4/4/2018
- [10] <https://www.mathworks.com/help/stats/export-classification-model-for-use-with-new-data.html>, προσπέλαση στις 25/5/2018
- [11] [http://web.csulb.edu/~jchang9/m521/m695\\_sp10\\_FinalReport\\_Minh\\_Angela.pdf](http://web.csulb.edu/~jchang9/m521/m695_sp10_FinalReport_Minh_Angela.pdf), προσπέλαση στις 15/5/2018
- [12] <https://arxiv.org/pdf/1101.3354.pdf>, προσπέλαση στις 15/5/2018
- [13] <https://www.robots.ox.ac.uk/~vedaldi/assets/pubs/parkhi12cat.pdf>, προσπέλαση στις 4/4/2018
- [14] <http://www.cs.toronto.edu/~kriz/cifar.html>, προσπέλαση στις 5/4/2018
- [15] <https://www.kaggle.com/deadskull7/cats-vs-dogs-images/data>, προσπέλαση στις 5/4/2018
- [16] <https://www.kaggle.com/jessicali9530/stanford-dogs-dataset>, προσπέλαση στις 5/4/2018



- [17] [https://uk.mathworks.com/help/stats/classreg.learning.classif.compactclassificationsvm.pr edict.html#responsive\\_offcanvas](https://uk.mathworks.com/help/stats/classreg.learning.classif.compactclassificationsvm.pr edict.html#responsive_offcanvas), προσπέλαση στις 4/4/2018
- [18] <https://www.vision.ee.ethz.ch/~surf/eccv06.pdf>, προσπέλαση στις 12/4/2018
- [19] <https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>, προσπέλαση στις 12/4/2018
- [20] <http://www.ai.mit.edu/courses/6.891/handouts/shi94good.pdf>, προσπέλαση στις 15/4/2018
- [21] [https://www.edwardrosten.com/work/rosten\\_2006\\_machine.pdf](https://www.edwardrosten.com/work/rosten_2006_machine.pdf), προσπέλαση στις 15/4/2018
- [22] [https://www.willowgarage.com/sites/default/files/orb\\_final.pdf](https://www.willowgarage.com/sites/default/files/orb_final.pdf), προσπέλαση στις 5/4/2018
- [23] <http://www.margaritachli.com/papers/ICCV2011paper.pdf>, προσπέλαση στις 9/3/2018
- [24] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.465.1117&rep=rep1&type=pdf> προσπέλαση στις 13/3/2018
- [25] <https://pdfs.semanticscholar.org/e045/aafb44693c9de943c750a6b60f2153ac3ccb.pdf>, προσπέλαση στις 13/3/2018
- [26] [https://www.researchgate.net/publication/258848394\\_FREAK\\_Fast\\_retina\\_keypoint](https://www.researchgate.net/publication/258848394_FREAK_Fast_retina_keypoint), προσπέλαση στις 13/3/2018
- [27] [https://www.cs.ubc.ca/~lowe/525/papers/calonder\\_eccv10.pdf](https://www.cs.ubc.ca/~lowe/525/papers/calonder_eccv10.pdf), προσπέλαση στις 1/2/2019
- [28] <https://www.bbc.com/timelines/zypd97h>, προσπέλαση στις 25/8/2019