



ΑΝΩΤΑΤΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΕΙΡΑΙΑ
ΤΕΧΝΟΛΟΓΙΚΟΥ ΤΟΜΕΑ

ΙΑΤΡΙΚΕΣ ΕΦΑΡΜΟΓΕΣ ΤΗΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ

ΦΑΜΕΛΙΑΡΗ ΒΑΣΙΛΙΚΗ
ΒΑΣΙΛΑΚΟΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΝΙΚΟΛΑΟΥ ΓΡΗΓΟΡΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΑΥΤΟΜΑΤΙΣΜΟΥ

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/Η κάτωθι υπογεγραμμένος/η Ραμνελίση Βασιλεί
του Ιωάννη με αριθμό μητρώου 40208 φοιτητής/τρια του
Τμήματος Μηχανικών Αυτοματισμού Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. 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 απόφασής της, μετά από αίτηση του ενδιαφερόμενου, του
αναθέτει εκ νέου την εκπόνηση της 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. Κατά τα λοιπά εφαρμόζονται τα
προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

Ο Δηλών


Ημερομηνία
28/7/2016

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο/Η κάτωθι υπογεγραμμένος/η Βασιλάκος Κωνσταντίνος,
του Κυριάκου, με αριθμό μητρώου 39290 φοιτητής/τρια του
Τμήματος Μηχανικών Αυτοματισμού Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. 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 απόφασής της, μετά από αίτηση του ενδιαφερόμενου, του
αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα
καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός
ημερολογιακού 6μήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα
προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»

Ο Δηλών

Μουσαν

Ημερομηνία

28/7/2016

Περίληψη

Η παρούσα εργασία πραγματεύεται το θέμα των εφαρμογών Τεχνητής Νοημοσύνης και Μηχανικής Μάθησης στον τομέα της Ιατρικής. Ξεκινάει με μία αποσαφήνιση των όρων της Τεχνητής Νοημοσύνης και της Μηχανικής Μάθησης και μετά παρέχει μία ιστορική αναδρομή που ακολουθεί την εξέλιξη των επιστημών πίσω από την Τεχνητή Νοημοσύνη από την αρχαιότητα μέχρι σήμερα καθώς και τις σύγχρονες εφαρμογές της σε ποικίλους τομείς. Έπειτα επισημαίνεται η σημασία του κλάδου της Τεχνητής Νοημοσύνης στην Ιατρική εστιάζοντας ιδιαίτερα στην πρόβλεψη της πιθανότητας εμφάνισης Σακχαρώδους Διαβήτη. Στη συνέχεια αναλύονται οι δημοφιλέστεροι σύγχρονοι τρόποι μηχανικής μάθησης και αναπτύσσονται διαφορετικά μοντέλα πρόβλεψης της πιθανότητας εμφάνισης Σακχαρώδους Διαβήτη βασισμένα σε εκείνους. Τέλος γίνεται σύγκριση των διαφορετικών μοντέλων που αναπτύχθηκαν μέσω διαφορετικών κριτηρίων αξιολόγησης της απόδοσης.

Abstract

This Thesis focuses on the Medical Applications of Artificial Intelligence and Machine Learning. The concepts of Artificial Intelligence and Machine Learning are first clarified and then a brief history of the major scientific advances that contributed to the development of Artificial Intelligence is provided, as well as a description of its various modern applications in different fields. The importance of Artificial Intelligence in the field of medicine is then emphasized with a focus on its importance in Diabetes Mellitus Prognosis. The most popular machine learning methods are then discussed in detail and different prediction models for Diabetes Mellitus are presented. Finally there is a comparison between the different prediction models using the most popular performance measures.

Περιεχόμενα

Κεφάλαιο 1: Τεχνητή Νοημοσύνη και Μηχανική Μάθηση

1.1 Εισαγωγή	6
1.2 Ιστορική Αναδρομή	8
1.3 Σύγχρονες Εφαρμογές της Τεχνητής Νοημοσύνης	13

Κεφάλαιο 2: Εφαρμογές της Τεχνητής Νοημοσύνης στην Ιατρική

2.1 Η Τεχνητή Νοημοσύνη στην Ιατρική.....	17
2.2 Διαβήτης και Τεχνητή Νοημοσύνη	20

Κεφάλαιο 3: Παραδείγματα Σύγχρονων Μεθόδων Μηχανικής Μάθησης

3.1 Εισαγωγή στις έννοιες που θα χρησιμοποιηθούν	24
3.2 Λογιστική Παλινδρόμηση (Logistic Regression).....	31
3.3 Αλγόριθμος K-NN (K-Nearest Neighbours Algorithm)	34
3.4 Μηχανές Διανυσμάτων Υποστήριξης (SVMs)	37
3.5 Νευρωνικά Δίκτυα (Neural Networks)	42

Κεφάλαιο 4: Μοντέλα Πρόβλεψης της Πιθανότητας Εμφάνισης Διαβήτη

4.1 Εισαγωγή – Χρήση Dataset	46
4.2 Μοντέλο με χρήση Νευρωνικού Δικτύου	48
4.3 Μοντέλο με προσέγγιση K-NN	55
4.4 Μοντέλο με τη μέθοδο Logistic Regression.....	60
4.5 Μοντέλο με SVM	62

Κεφάλαιο 5: Σύγκριση Διαφορετικών Μοντέλων Πρόβλεψης

5.1 Σύγκριση της Απόδοσης Διαφορετικών Συναρτήσεων Εκπαίδευσης Νευρωνικών Δικτύων	65
---	----

5.2 Σύγκριση της Απόδοσης Διαφορετικών Μοντέλων Πρόβλεψης	78
---	----

Βιβλιογραφία	80
---------------------------	----

Παράρτημα

Κατάλογος Εικόνων	84
Κατάλογος Πινάκων.....	86

Κεφάλαιο 1:

Τεχνητή Νοημοσύνη και Μηχανική Μάθηση

1.1 Εισαγωγή

Ο Marvin Minsky, θεωρούμενος από πολλούς ως ο πατέρας της Τεχνητής Νοημοσύνης, γράφει στο βιβλίο του “Society of Mind” πως μία έννοια όπως η Νοημοσύνη δεν θα έπρεπε να οριστεί με απόλυτο τρόπο και αντίθετα δίνει τον “ορισμό” της ως εξής: “Ο νους περιέχει διαδικασίες που μας δίνουν τη δυνατότητα να λύνουμε προβλήματα που θεωρούμε δύσκολα. “Νοημοσύνη” είναι το όνομα που δίνουμε για οποιαδήποτε από αυτές τις διαδικασίες δεν καταλαβαίνουμε ακόμα”

(Minsky, M. 1986 p.71)

Παρά το γεγονός ότι ο ορισμός αυτός στην προσπάθειά του να είναι καθολικός σημαίνει πως υπόκινται σε αλλαγές με την πάροδο της επιστήμης, είναι εντούτοις δυνατό να ορίσουμε μία λίστα από θεμελιώδεις δυνατότητες της νοημοσύνης ως εξής (Hofstadter, D.R. 1979 p.26)

- Ανταπόκριση σε καταστάσεις με ελαστικότητα, δηλαδή μη μονότονη αντίδραση σε ταυτόσημα προβλήματα (αποφυγή μηχανικής συμπεριφοράς)
- Η αντίληψη και κατανόηση ασαφών ή αντιφατικών μηνυμάτων από τα συμφραζόμενα
- Η αναγνώριση και ιεράρχηση των καταστάσεων με βάση τη σπουδαιότητά τους
- Η εύρεση ομοιοτήτων σε φαινομενικά ανόμοιες καταστάσεις
- Η εύρεση διαφορών σε καταστάσεις που εκ πρώτης όψεως μοιάζουν παρόμοιες
- Η σύνθεση νέων εννοιών συνδυάζοντας παλαιότερες έννοιες με νέους τρόπους

Όλες αυτές οι δυνατότητες είναι έμφυτες στον άνθρωπο και η προσπάθεια αναπαραγωγής τους σε μηχανές οι οποίες χαρακτηρίζονται από το γεγονός ότι είναι άκαμπτες, ακολουθούν μόνο κανόνες επακριβώς ορισμένους και δεν παρουσιάζουν καμία τάση προς δημιουργικότητα, κατηγοριοποίηση ή ιεράρχηση, μοιάζει με ένα αδύνατο εγχείρημα. Όμως είναι και το παράδοξο με το οποίο ασχολείται η έρευνα πάνω στην Τεχνητή Νοημοσύνη – η οποία αποτελεί μία προσπάθεια των ανθρώπων να συνδυάσουν μεγάλα σύνολα κανόνων που να δείχνουν σε άκαμπτες μηχανές πώς να γίνονται πιο ευέλικτες.

Η **Τεχνητή Νοημοσύνη** (Artificial Intelligence) είναι ο τομέας της Επιστήμης των Υπολογιστών που ασχολείται με τη σχεδίαση και την υλοποίηση προγραμμάτων τα οποία είναι ικανά να μιμηθούν τις ανθρώπινες γνωστικές ικανότητες, εμφανίζοντας έτσι χαρακτηριστικά που αποδίδουμε συνήθως σε ανθρώπινη συμπεριφορά, όπως για παράδειγμα η επίλυση προβλημάτων, η αντίληψη μέσω της όρασης, η μάθηση, η εξαγωγή συμπερασμάτων, η κατανόηση φυσικής γλώσσας, κλπ
(Βλαχάβας, I. et al. 2006)

Σε αυτό το σημείο θα πρέπει να κάνουμε μία διευκρίνηση. Αν και ο όρος Τεχνητή Νοημοσύνη ετυμολογικά μας παραπέμπει σε μία μηχανή που δημιουργήθηκε κατα τέτοιο τρόπο ώστε να έχει η ίδια τη δική της Νοημοσύνη και κατά συνέπεια να ακολουθεί τις αρχές λειτουργίας ενός ανθρώπου, οι μέχρι τώρα εξελίξεις στον τομέα δεν έχουν φτάσει σε αυτό το επίπεδο. Άρα θα έλεγε κανείς ότι μπορούμε να διακρίνουμε δύο τύπους Τεχνητής Νοημοσύνης (Searle, J. 1980), την **Ισχυρή Τεχνητή Νοημοσύνη** (Strong Artificial Intelligence) και την **Αδύναμη Τεχνητή Νοημοσύνη** (Weak Artificial Intelligence).

Στην Αδύναμη Τεχνητή Νοημοσύνη η μηχανή είναι απλώς ένα καλά σχεδιασμένο εργαλείο το οποίο λειτουργεί υπό τις εντολές του προγραμματιστή δίχως να έχει δυνατότητα αντίληψης. Η λειτουργία της μπορεί να θεωρηθεί ότι ίσως να προσομοιώνει ως ένα σημείο την ανθρώπινη νοημοσύνη όμως η μηχανή δεν διαθέτει νοημοσύνη ίδια ή παρόμοια με του ανθρώπου.

Αντίθετα στην Ισχυρή Τεχνητή Νοημοσύνη η μηχανή έχει νοημοσύνη με την έννοια ότι, έχοντας τα κατάλληλα προγράμματα, μπορεί κυριολεκτικά να καταλαβαίνει και να λαμβάνει άλλα γνωσιακά επίπεδα και άρα δεν αποτελεί απλά προσομοίωση αλλά αναπαραγωγή της ανθρώπινης νοημοσύνης.

Καθώς οι έως τώρα εφαρμογές της Τεχνητής Νοημοσύνης ανήκουν στην κατηγορία της Αδύναμης Τεχνητής Νοημοσύνης οι δύο αυτοί όροι συχνά χρησιμοποιούνται αδιακρίτως.

Κύριο χαρακτηριστικό των συστημάτων που εφαρμόζουν τις αρχές της Τεχνητής Νοημοσύνης είναι η δυνατότητα τους να λαμβάνουν κάποια σήματα ως εισόδους, να τα επεξεργάζονται με τρόπο παρόμοιο με εκείνο της ανθρώπινης σκέψης και να εξάγουν κάποια συμπεράσματα ως έξοδο. Αντίστοιχα στους ανθρώπους, το άτομο επιλέγει και προσλαμβάνει τα ερεθίσματα (τα οποία οποία είναι οι φορείς των πληροφοριών) με τους αισθητηριακούς υποδοχείς, τα μετασχηματίζει σε μία μορφή μηνυμάτων που μπορούν να χρησιμοποιηθούν από το νοητικό (γνωστικό) σύστημα, τα επεξεργάζεται, τα συγκρατεί για λίγο ή για πολύ χρόνο και

τα χρησιμοποιεί όταν τα χρειάζεται. (Πόρποδας, Κ.Δ., 1996). Αυτή η διαδικασία της απόκτησης, συγκράτησης και χρησιμοποίησης των πληροφοριών συγκροτεί το **φαινόμενο της μάθησης** (Estes 1975)

Η **Μηχανική Μάθηση** είναι ο τομέας της Επιστήμης των Υπολογιστών που ασχολείται με την ανάπτυξη συστημάτων ικανά να μαθαίνουν από εμπειρία χωρίς να έχουν προγραμματιστεί ρητά από πριν. Αν και είναι ένας τομέας στενά συνδεδεμένος και μάλιστα αποτελεί κλάδο του ευρύτερου τομέα της Τεχνητής Νοημοσύνης, η Μηχανική Μάθηση ασχολείται πλέον κυρίως με την αντιμετώπιση επιλύσιμων πρακτικών προβλημάτων και όχι με την προσπάθεια για την επίτευξη των στόχων της Τεχνητής Νοημοσύνης (Langley, P. 2011). Για να πετύχει τους σκοπούς της η Μηχανική Μάθηση κάνει χρήση περισσότερο στατιστικών και στοχαστικών μοντέλων της θεωρίας πιθανοτήτων.

1.2 Ιστορική Αναδρομή

Η Τεχνητή Νοημοσύνη μπορεί να μοιάζει με μία σύγχρονη έννοια όμως τα θεμέλια για τις προσπάθειες επίτευξής της ξεκίνησαν πολλά χρόνια πριν.

Ήδη από την αρχαιότητα υπήρχε το όραμα για τη δημιουργία κατασκευών που να έχουν νοημοσύνη και αυτό φαίνεται στη μυθολογία. Παραδείγματα αποτελούν οι Χρυσές Θεραπαινίδες, δημιουργήματα του Ηφαίστου που εμφανίζονται στην Ιλιάδα και έμοιαζαν με κοπέλες με δική τους φωνή και λογική και ο Τάλως, ένας χάλκινος γίγαντας που προστάτευε το νησί της Κρήτης, αναφορές για τον οποίο έχουμε σε κείμενα από τον Απολλώνιο τον Ρόδιο και τον Απολλόδωρο. (Καλλιγερόπουλος Δ. & Βασιλειάδου Σ. 2005)

Στα θεμέλια της πρακτικής προσέγγισης της Τεχνητής Νοημοσύνης αξίζει να αναφέρουμε τη συνεισφορά του Αριστοτέλη (384π.Χ.) ο οποίος πρώτος περιέγραψε το Συλλογισμό, έναν τρόπο κωδικοποίησης της ορθής σκέψης μέσω κανόνων που περιέγραφαν τη διαδικασία της σκέψης και αποτέλεσαν τη βάση του πεδίου της λογικής.

Ο Πορφύριος ο Τύριος (280 μ.Χ.) κάνει κατηγοριοποιήσεις στο βιβλίο του “Εισαγωγή” μεταξύ άλλων και της γνώσης και της λογικής.

Ο René Descartes στις αρχές του 17^{ου} αιώνα κάνει αναφορές στις διαφορές των νοητικών φαινομένων και του σώματος των ζώων που δεν είναι παρά περίπλοκες μηχανές.

Το 1642 ο Blaise Pascal δημιούργησε την πρώτη αριθμομηχανή.

Το 1672 ο Gottfried Leibniz μεταξύ άλλων ανακάλυψε το επίσημο δυαδικό σύστημα όπως χρησιμοποιείται στη σημερινή του μορφή.

Το 1833 περίπου ο Charles Babbage και η Ada Lovelace δουλεύουν για τη δημιουργία προγραμματιζόμενων μηχανικών αριθμομηχανών και τη δημιουργία της Αναλυτικής Μηχανής.

Το 1842 η Ada Lovelace έγραψε τα πρώτα προγράμματα για την Αναλυτική Μηχανή

Το 1854 ο George Boole δημιούργησε την άλγεβρα Boole σε μία προσπάθεια να δώσει αλγεβρική έκφραση των νοητικών λειτουργιών.

Το 1889 ο Herman Hollerith εφηύρε τις διάτρητες κάρτες.

Το 1936 ο Alan Turing πρότεινε τη Μηχανή Turing (Turing Machine), μία μηχανή περιγραφόμενη από μία πεπερασμένη ακολουθία εντολών με τη δυνατότητα να εκτελεί οποιαδήποτε λειτουργία.

Το 1938 ο Konrad Zuse έφτιαξε τον πρώτο ηλεκτρομηχανικό προγραμματιζόμενο υπολογιστή, τον Z1.

Το 1943 οι Warren Sturgis McCulloch και Walter Pitts έθεσαν τα θεμέλια για τα Τεχνητά Νευρωνικά Δίκτυα με το άρθρο τους “A logical calculus of the ideas immanent in nervous activity”. Στην εργασία τους αναφέρονται σε ένα τεχνητό μοντέλο νευρώνων σύμφωνα με αυτό του εγκεφάλου και ικανό να υπολογίζει μαθηματικές συναρτήσεις.

Το 1944 οι John von Neumann και Oskar Morgenstern με τα έργα τους θεμελίωσαν τη Σημασία της Θεωρίας Παιγνίων (Game Theory) ως ξεχωριστό κλάδο που αργότερα αποδείχθηκε σημαντικός στην πρόοδο της Τεχνητής Νοημοσύνης.

Το 1946 κατασκευάστηκε ο ENIAC, ο πρώτος ηλεκτρονικός υπολογιστής

Το 1948 ο Claude Shannon θεμελίωσε τη θεωρία πληροφοριών που βοήθησε στις ψηφιακές επικοινωνίες.

Το 1949 ο Donald Hebb έκανε αναφορά σε έναν κανόνα που αφορούσε τη μεταβολή της δύναμης σύνδεσης μεταξύ των νευρώνων έτσι ώστε να επιτυγχάνεται η μάθηση.

Το 1950 ο Alan Turing εισήγαγε την έννοια του Turing Test, ενός τρόπου μέτρησης της νοημοσύνης των μηχανών.

Το 1951 ο Marvin Minsky και ο Dean Edmonds έφτιαξαν το πρώτο νευρωνικό δίκτυο με την ονομασία SNARC – Stochastic Neural Analog Reinforcement Calculator. Είχε 40 νευρώνες και λειτουργούσε με συνάψεις Hebb που ρύθμιζαν τα βάρη (weights) σύμφωνα με την επιτυχία στην εκτέλεση μίας συγκεκριμένης λειτουργίας.

Το 1956 λαμβάνει χώρα το Dartmouth Summer Conference διοργανωμένο από τους John McCarthy, Marvin Minsky, Nathaniel Rochester και Claude Shannon και πρώτη φορά αναφέρεται και γίνεται αποδεκτός ο όρος Τεχνητή Νοημοσύνη.

Το 1957 παρουσιάζεται από τους Allen Newell, Herbert Simon και John Shaw ο Γενικός Επιλυτής Προβλημάτων (General Problem Solver), ένα πρόγραμμα που ακολουθούσε τον ανθρώπινο τρόπο σκέψης για την επίλυση προβλημάτων.

Το 1958 ο John McCarthy δημιούργησε τη συναρτησιακή γλώσσα προγραμματισμού LISP η οποία χρησιμοποιείται ακόμα κ σήμερα.

Το 1958 παρουσιάζονται στο Teddington Conference του Ηνωμένου Βασιλείου οι εργασίες “Some Methods of Heuristic Programming and Artificial Intelligence” του Marvin Minsky, “Pandemonium” του Oliver Selfridge και “Programs with Common Sense” του John McCarthy, ο οποίος περιέγραφε ένα υποθετικό σύστημα Τεχνητής Νοημοσύνης που χρησιμοποιούσε γνώση για την επίλυση προβλημάτων και μπορεί να θεωρηθεί ως το πρώτο πλήρες σύστημα Τεχνητής Νοημοσύνης.

Το 1958 επίσης προτάθηκε από τον Richard Friedberg η έννοια της Μηχανικής Εξέλιξης που σήμερα αποτελεί τους Γενετικούς Αλγόριθμους (Genetic Algorithms). Η βασική ιδέα της λειτουργίας τους είναι πως αυξάνεται η βελτίωση του προγράμματος δημιουργώντας τυχαίες μεταλλάξεις στον κώδικα και κρατώντας όσες τον βελτιώνουν.

Το 1959 ιδρύθηκε το MIT AI Lab, το Εργαστήριο Τεχνητής Νοημοσύνης του Massachusetts Institute of Technology.

Στις αρχές τις δεκαετίας του 1960 ο Ray Solomonoff δίνει πιο μαθηματικές βάσεις στη θεωρία της Τεχνητής Νοημοσύνης εισάγοντας τις μεθόδους Bayes για επαγωγικό συλλογισμό και προβλέψεις.

Το 1961 ο John Lucas στην εργασία του Minds, Machines and Gödel γράφει πως δεν είναι δυνατόν να υπάρξει η τεχνητή νοημοσύνη με βάση το θεώρημα του Gödel της Μη Πληρότητας του 1931.

Το 1963 ιδρύθηκε από τον John McCarthy το εργαστήριο Τεχνητής Νοημοσύνης στο ινστιτούτο ερευνών του πανεπιστημίου Stanford SRI.

Το 1965 ο Joseph Weizenbaum έφτιαξε το ELIZA, ένα διαδραστικό πρόγραμμα που μπορούσε να κάνει διάλογο στην Αγγλική γλώσσα για οποιοδήποτε θέμα παραφράζοντας τα λόγια του χρήστη.

Το 1967 το πρόγραμμα DENDRAL από τους Edward Feigenbaum, Joshua Lederberg, Bruce Buchanan & Georgia Sutherland ήταν το πρώτο έμπειρο σύστημα (expert system), δηλαδή πρόγραμμα βασισμένο σε γνώση για την εξαγωγή επιστημονικών συμπερασμάτων.

Το 1969 στο SRI δημιουργήθηκε το πρώτο ρομπότ, το Shakey.

Το 1969 οι Arthur Bryson και Yu-Chi Ho περιγράφουν τον αλγόριθμο Back Propagation ως μέθοδο βελτιστοποίησης δυναμικών συστημάτων πολλών επιπέδων.

Το 1969 ο Marvin Minsky και ο Seymour Papert δημοσίευσαν την εργασία τους στα Perceptrons που έδειχνε κάποια όρια στη μορφή feed-forward.

Στα τέλη της δεκαετίας του 1960 ξεκίνησε η εποχή της κριτικής ενάντια στην Τεχνητή Νοημοσύνη, γνωστή και σαν A.I. Winter, τόσο λόγω των ορίων που τότε εμφανίστηκαν (για παράδειγμα με την εργασία Perceptrons των Minsky & Seymour), τα οποία μείωσαν την εμπιστοσύνη των επενδυτών, όσο και του προβλήματος της συνδυαστικής έκρηξης που δεν μπορούσε να ξεπεραστεί εκείνη την εποχή με τα συστήματα συμβολικής προσέγγισης.

Το 1972 ο Alain Colmerauer ανέπτυξε τη γλώσσα Prolog που χρησιμοποιείται μέχρι σήμερα.

Το 1974 ο Ted Shortliffe έδειξε πως το πρόγραμμα MYCIN ήταν ικανό να δώσει ιατρικές διαγνώσεις και η γνώση του δεν προέκυπτε από σύστημα κανόνων όπως το DENDRAL αλλά από συνεντεύξεις με γιατρούς. Το σύστημα περιείχε 450 κανόνες και είχε την ικανότητα να κάνει διάγνωση πάνω σε μολύνσεις του αίματος. Χρησιμοποιούσε επίσης συντελεστές βεβαιότητας με τους οποίους υπολόγιζε την αβεβαιότητα.

Το 1974 ο Paul Werbos περιγράφει την πιθανή χρήση του αλγορίθμου Back Propagation στα νευρωνικά δίκτυα.

Το 1975 ο Marvin Minsky προτείνει την έννοια των Πλαισίων (frames) μέσω του ομότιτλου άρθρου του ως μέσο αναπαράστασης της γνώσης με δομημένο και ιεραρχημένο τρόπο.

Το 1978 οι Mark Stefik και Peter Friedland μέσω του προγράμματος MOLGEN δείχνουν ότι μπορεί να χρησιμοποιηθεί αντικειμενοστραφής προγραμματιστική αναπαράσταση για τη σχεδίαση πειραμάτων κλωνοποίησης γονιδίων.

Το 1979 ο Bill VanMelle γενίκευσε την αναπαράσταση γνώσης και τον τρόπο συλλογισμού του MYCIN με το δικό του πρόγραμμα EMYCIN το οποίο αποτέλεσε βάση για πολλά εμπορικά έμπειρα συστήματα.

Το 1979 οι Jack Myers και Harry Pople ανέπτυξαν το INTERNIST, ένα σύστημα ιατρικών διαγνώσεων βασισμένο σε γνώση.

Στα μέσα της δεκαετίας του '80 η έρευνα στρέφεται και πάλι στα Νευρωνικά Δίκτυα με τη χρήση του αλγορίθμου BackPropagation.

Μέχρι τα τέλη της δεκαετίας του 1980 φαίνεται πως η ανάπτυξη στον τομέα της Τεχνητής Νοημοσύνης εστιάζεται στη γνώση και την εμπορική εκμετάλλευση των αποτελεσμάτων που παρέχει, ενώ αργότερα με την εισαγωγή του Διαδικτύου ξεκινούν οι εφαρμογές της σε αυτό μέσω μηχανών αναζήτησης, βελτίωσης του ηλεκτρονικού εμπορίου, προγραμμάτων

ηλεκτρονικής εκπαίδευσης καθώς επίσης αναπτύσσονται και εφαρμογές διευκόλυνσης χρήσης καθημερινών συσκευών όπως τα έξυπνα περιβάλλοντα διεπαφής χρήστη – μηχανής.

Η Ιστορία της Τεχνητής Νοημοσύνης μπορεί συνοπτικά να χωριστεί σε τέσσερις περιόδους.

- i) Η πρώτη περίοδος είναι η **Προϊστορική**, όπου η Τεχνητή Νοημοσύνη δεν είναι παρά ένα μακρινό όραμα που εμφανίζεται κυρίως μέσα σε μύθους και διηγήματα φαντασίας.
- ii) Έπειτα υπάρχει η **Κλασική Περίοδος** που εκτείνεται μέχρι τα μέσα της δεκαετίας του 1960 και οι εφαρμογές της Τεχνητής Νοημοσύνης εστιάζονται σε συστήματα ικανά να παίζουν παιχνίδια και να λύνουν γρίφους.
- iii) Ακολουθεί η **Ρομαντική Περίοδος** μέχρι και τα μέσα της δεκαετίας του 1970 όπου οι επιστήμονες προσπαθούν να αναπτύξουν συστήματα ικανά να αναγνωρίσουν τη φυσική γλώσσα και να κατανοούν διαλόγους και ιστορίες.
- iv) Μετά έρχεται η **Μοντέρνα Περίοδος** μέχρι και τα τέλη της δεκαετίας του 1980 όπου κυρίως εμφανίζεται η τάση ανάπτυξης συστημάτων βασισμένων σε γνώση και η εμπορική εκμετάλλευση των ερευνητικών αποτελεσμάτων.

Η σημερινή εποχή μπορεί να θεωρηθεί ως η μετα-μοντέρνα περίοδος όπου οι εφαρμογές της Τεχνητής Νοημοσύνης έχουν επηρεασθεί αρκετά από τη χρήση του Διαδικτύου, και εστιάζουν επίσης στη διευκόλυνση της χρήσης νέων συσκευών υψηλής τεχνολογίας και οικιακών συσκευών (έξυπνα σπίτια, αυτόνομα αυτοκίνητα κλπ) καθώς και στην ανάπτυξη προσαρμοστικών διεπαφών ανθρώπου – μηχανής.

1.3 Σύγχρονες Εφαρμογές της Τεχνητής Νοημοσύνης

Η εξελισσόμενη έρευνα στον τομέα της Τεχνητής Νοημοσύνης έχει επιφέρει πολλές σημαντικές επιστημονικές ανακαλύψεις και εφευρέσεις. Πολλές όμως από αυτές δεν θεωρούνται Τεχνητή Νοημοσύνη, τόσο λόγω του ότι δεν αποτελούν αντικείμενο της Ισχυρής Τεχνητής Νοημοσύνης κατά τον John Searle (βλ. σελ. 5) όσο και του λεγόμενου Φαινομένου Τεχνητής Νοημοσύνης (Artificial Intelligence Effect) σύμφωνα με το οποίο “Τεχνητή Νοημοσύνη αποτελεί οτιδήποτε δεν έχει δημιουργηθεί ακόμα” (Hofstadter, D.R. 1979 p. 601). Δηλαδή η υψηλή δημοτικότητα και η ευρεία αποδοχή των εφαρμογών τεχνητής νοημοσύνης τις καθιστά κοινές με αποτέλεσμα να μην αναγνωρίζονται ως τεχνητή νοημοσύνη. Πάραυτα έχουμε εφαρμογές της Τεχνητής Νοημοσύνης σε τομείς όπως η Οικονομία, η Βιομηχανία, η Αεροπορία, το Διαδίκτυο και οι Τηλεπικοινωνίες καθώς και Εμπορικές, Οικιακές και Ιατρικές Εφαρμογές.

Στην οικονομία ήταν ανέκαθεν γνωστή η ανάγκη και η σημασία της ικανότητας προβλέψεων. Συχνά χρησιμοποιούνται συστήματα Τεχνητής Νοημοσύνης για την πρόβλεψη της κίνησης στην κεφαλαιαγορά για καλύτερες επενδύσεις σε μετοχές. Επίσης υπάρχει η ανάγκη πρόβλεψης της αξίας υποτίμησης των προϊόντων ώστε να γίνεται μία ορθή εκτίμηση της κατάστασης των περιουσιακών στοιχείων της εκάστοτε εταιρείας. Ακόμη είναι σημαντική η πρόβλεψη υποτίμησης επισφαλών απαιτήσεων πελατών ή σε κάποιες περιπτώσεις επισφαλών προμηθευτών, η πρόβλεψη καταστροφών και αλλαγής συναλλαγματικών διαφορών.

Σημαντικές για μία εταιρεία είναι και οι προβλέψεις της αγοράς, που χρησιμοποιούνται για τον προσδιορισμό της καλύτερης πιθανής τιμής των προϊόντων και πιθανή αύξηση ή μείωσή της ή την πιθανότητα επιτυχίας ενός προϊόντος σε μία νέα αγορά.

Η Τεχνητή Νοημοσύνη δίνει επίσης τη δυνατότητα σε οικονομικούς οργανισμούς να ανιχνεύσουν αλλαγές στη μέχρι τότε συνηθισμένη οικονομική δραστηριότητα πελατών και να ειδοποιηθούν για περαιτέρω έρευνες σε πιθανές προσπάθειες εξαπάτησης πελατών.

Η βιομηχανία επωφελείται από την τεχνητή νοημοσύνη καθώς ρομπότ με μηχανική όραση και κατάλληλο σχεδιασμό ενεργειών μπορούν και αντικαθιστούν ανθρώπους σε επικίνδυνες ή επίπονες εργασίες. Αποδεικνύονται ακόμα πολύ αποτελεσματικά σε επαναλαμβανόμενες εργασίες στις οποίες οι άνθρωποι μπορεί να κάνουν λάθη λόγω στιγμιαίας μείωσης της προσοχής ή αποφεύγουν καθώς η φύση των επαναλαμβανόμενων εξειδικευμένων εργασιών τους προσδίδει ένα αίσθημα κοινωνικής υποβάθμισης, πλήξης και μειωμένης συνεισφοράς στις

ανθρώπινες ανάγκες και την κοινωνία καθώς αντί για το ολοκληρωμένο έργο βλέπουν μόνο ένα τμήμα του πάνω στο οποίο εργάζονται.

Επίσης υπάρχουν τα Έμπειρα Συστήματα που χρησιμοποιούνται για τον έλεγχο των μονάδων παραγωγής και μέσω της αδιάλειπτης παρακολούθησης της μονάδας και της κατάλληλης επεξεργασίας των σημάτων εισόδου σε συνδυασμό με τη βάση γνώσεων πραγματοποιούν ανίχνευση, διάγνωση και διόρθωση σφαλμάτων, καταγραφή συμβάντων και δράσεων, ειδοποίηση των χειριστών μέσω ευφών συναγεμίων και προγραμματισμό αποθηκών, πόρων, τμημάτων και κύριου εξοπλισμού της παραγωγής.

Στην παροχή ηλεκτρικής ενέργειας η Τεχνητή Νοημοσύνη βοηθάει στην βελτιστοποίηση του προγραμματισμού των μονάδων παραγωγής ενέργειας ώστε να καλύπτουν τις ανάγκες των καταναλωτών χωρίς προβλήματα στο δίκτυο παροχής ενέργειας. Τα Συστήματα Διαχείρισης Ενέργειας (E.M.S. – Energy Management Systems) πραγματοποιούν ένα ταίριασμα της παραγωγής με τη στιγμιαία ζήτηση φορτίου και προβλέπουν τα ημερήσια και εβδομαδιαία φορτία ζήτησης ώστε να αναπτύξουν μία στρατηγική για την ικανοποίηση των αναγκών με το χαμηλότερο δυνατό κόστος και επωφελούνται από τη χρήση προβλέψεων μέσω εφαρμογών Τεχνητής Νοημοσύνης. (Αλαφοδήμος, Κ. 2014)

Η αεροπορία χρησιμοποιεί πολλούς κλάδους της Τεχνητής Νοημοσύνης. Υπάρχουν προσομοιωτές πτήσης για μάχη ή εκπαίδευση αναπληρωτών πιλότων που χρησιμοποιούν τεχνητή νοημοσύνη για επεξεργασία των δεδομένων των πτήσεων. Στην περίπτωση εξομοίωσης μάχης μέσω εφαρμογών τεχνητής νοημοσύνης γίνεται έλεγχος και επιλογή της καλύτερης πιθανής στρατηγικής για επιτυχία μέσω έμπειρων συστημάτων βασισμένων σε κανόνες (rule based expert systems). Τα προγράμματα τεχνητής νοημοσύνης βοηθούν επίσης τους πιλότους κατά τη διάρκεια πτήσεων προτείνοντας ελιγμούς. Ακόμα υπάρχουν και τεχνητοί πιλότοι που οδηγούν τα αεροσκάφη κατά τη διάρκεια εξομοίωσης πτήσεων και βοηθούν πολύ σε περιπτώσεις που χρειάζεται η συλλογή δεδομένων πολλαπλών αεροσκαφών.

Επίσης χρησιμοποιείται η τεχνητή νοημοσύνη με την ανάπτυξη λογισμικού αναγνώρισης ομιλίας (speech recognition software) ώστε οι εκπαιδευόμενοι ελεγκτές εναέριας κυκλοφορίας να μπορούν να κατευθύνουν τους τεχνητούς πιλότους.

Στο Διαδίκτυο η Τεχνητή Νοημοσύνη μέσω επεξεργασίας φυσικής γλώσσας εμφανίζεται σε πολλούς ιστότοπους μέσω Αυτόματων Βοηθών (automated online assistants),

προγράμματα-πράκτορες τα οποία καθοδηγούν τον χρήστη-επισκέπτη ενός ιστότοπου και του απαντούν σε ερωτήματα σύμφωνα με το κείμενο που ο ίδιος πληκτρολογεί.

Επίσης έχουν αναπτυχθεί προγράμματα για φιλτράρισμα της ηλεκτρονικής αλληλογραφίας (e-mail) από ανεπιθύμητα μηνύματα χρησιμοποιώντας Τεχνητή Νοημοσύνη στην αναγνώριση και ταξινόμηση τέτοιων μηνυμάτων για παράδειγμα μέσω του τρόπου γραφής των εισερχόμενων μηνυμάτων, την συχνότητα εμφάνισης λέξεων-κλειδιών που υπάρχουν συνήθως σε ανεπιθύμητα μηνύματα, το όνομα του αποστολέα κ.λ.π.

Κάποιοι ιστότοποι χρησιμοποιούν αλγόριθμους Τεχνητής Νοημοσύνης για να προτείνουν στο χρήστη ταινίες, σειρές, μουσική ή νέα που πιθανόν να είναι εντός των ενδιαφερόντων του χρήστη βασισμένα σε προηγούμενες αξιολογήσεις του σε αντίστοιχα προϊόντα ή αξιολογήσεις χρηστών που εμφανίζουν ομοιότητες με τον εν λόγω χρήστη (ίδιο φύλο, καταγωγή, ενδιαφέροντα κλπ).

Αλγόριθμοι Τεχνητής Νοημοσύνης χρησιμοποιούνται επίσης για μηχανές αναζήτησης καθώς και στην παροχή στοχευμένων διαφημίσεων (targeted advertising) σύμφωνα με τις προτιμήσεις του χρήστη.

Κάποιες από τις εμπορικές και οικιακές εφαρμογές της Τεχνητής Νοημοσύνης αποτελούν η αυτόματη τηλεφωνική εξυπηρέτηση πελατών μέσω συστημάτων αναγνώρισης φωνής, η ανάπτυξη όλο και καλύτερων προσαρμοστικών διεπαφών ανθρώπου-μηχανής και διεπαφών που βοηθούν άτομα με ειδικές ανάγκες όπως άτομα με μειωμένη όραση ή κινητικές δυσκολίες μέσω επεξεργασίας της φυσικής γλώσσας και αναγνώρισης φωνής, οπτικής αναγνώρισης χαρακτήρων, ανάγνωσης κειμένων και παραγωγής φυσικής γλώσσας, και τα αυτόματα κιβώτια ταχυτύπων για τα αυτοκίνητα που χρησιμοποιούν ελεγκτές ασαφούς λογικής (fuzzy logic controllers).

Επίσης στον τομέα των παιχνιδιών υπάρχουν πολλές εφαρμογές της Τεχνητής Νοημοσύνης όπως το πρώτο ευρείας κυκλοφορίας οικιακό ρομπότ, το Furby και το πιο βελτιωμένο σκυλάκι AIBO της Sony με μεγάλη αυτονομία και χαρακτηριστικά νοημοσύνης όπως δυνατότητες αναγνώρισης ομιλίας, έκφρασης συναισθημάτων με κινήσεις και ήχους και μετακίνησης σε διάφορες επιφάνειες.

Στην ιατρική υπάρχει μία πληθώρα εφαρμογών Τεχνητής Νοημοσύνης, από συστήματα που ειδοποιούν τους ιατρούς στην περίπτωση που εμφανιστεί κάποια αλλαγή στην κατάσταση ενός ασθενούς έως συστήματα που αναλύουν μεγάλο όγκο δεδομένων και προτείνουν

διασυνδέσεις μεταξύ περίπλοκων μοτίβων που προηγουμένως δεν θεωρούνταν ότι συσχετίζονται. Θα αναλυθούν εκτενώς οι Ιατρικές Εφαρμογές στο επόμενο κεφάλαιο.

Κεφάλαιο 2:

Ιατρικές Εφαρμογές της Τεχνητής Νοημοσύνης

2.1 Η Τεχνητή Νοημοσύνη στην Ιατρική

Οι επιστήμονες ήδη από τα πρώτα χρόνια στην εξέλιξη της Τεχνητής Νοημοσύνης είχαν καταλάβει τη σημασία που θα μπορούσε να έχει ο τομέας αυτός όταν εφαρμοστεί στην Ιατρική. Όπως φαίνεται και από την εργασία των Ledley, R. & Lusted, L. (1959) οι οποίοι έκαναν μία προσπάθεια ανάλυσης του συλλογισμού ενός ειδικού κατά τη διάρκεια μίας διάγνωσης με σκοπό τη χρήση ηλεκτρονικών υπολογιστών ως βοήθημα κατά τη διαγνωστική διαδικασία, οι πρώτες εφαρμογές της Τεχνητής Νοημοσύνης εστίαζαν στη δημιουργία διαγνωστικών εργαλείων ικανών να προσφέρουν διαγνώσεις το ίδιο καλά αν όχι καλύτερα από έναν ειδικό. Με τα χρόνια δημιουργήθηκε μία κοινότητα επιστημόνων τόσο από τον τομέα των υπολογιστών όσο και από τον τομέα της υγείας που ξεκίνησαν την προσπάθεια για ένα νέο τομέα με την ονομασία “Τεχνητή Νοημοσύνη στην Ιατρική”, Artificial Intelligence in Medicine ή AIM.

Οι πρώτες εφαρμογές της Τεχνητής Νοημοσύνης στην Ιατρική εστίαζαν σε διαγνώσεις που να είναι βασισμένες σε συμβολικά μοντέλα των ασθενειών και τη συσχέτισή τους με διάφορους παράγοντες των ασθενών και κλινικές εκδηλώσεις αντί για απλές στατιστικές και στοχαστικές μεθόδους (Shortliffe, E. & Clancey, W. 1984)

Στη σημερινή εποχή γίνεται τόσο χρήση συμβολικών μοντέλων προσέγγισης των ασθενειών όσο και στατιστικών μεθόδων και η Τεχνητή Νοημοσύνη χρησιμοποιείται λιγότερο στις διαγνώσεις ρουτίνας και περισσότερο στη συνταγογράφηση φαρμάκων, σε κλινικά εργαστήρια και στην κλινική παρακολούθηση καθώς και στις μονάδες εντατικής θεραπείας. Τα σύγχρονα αυτά συστήματα συνήθως περιγράφονται ως Κλινικά Συστήματα Υποστήριξης Αποφάσεων – Clinical Decision Support Systems ή CDSSs. (Coiera, E. 2015)

Τα συστήματα αυτά μέσω της διαχείρισης μεγάλου όγκου δεδομένων και γνώσεων βοηθούν τους εργαζόμενους στον τομέα της υγείας κάνοντας χρήση των βάσεων δεδομένων που έχουν και των σημάτων που λαμβάνουν σε πραγματικό χρόνο. Για παράδειγμα, ένα σύστημα υποστήριξης αποφάσεων μπορεί να έχει το ιστορικό ενός ασθενούς και να ειδοποιεί

τον ειδικό όταν βλέπει αντενδείξεις σε μία συστηνόμενη θεραπεία ή όταν ανιχνεύσει κάποια μοτίβα σε κλινικά δεδομένα που υποδεικνύουν κάποια αλλαγή στην κατάσταση του ασθενή.

Τα συστήματα διάγνωσης αν και στην αρχή ήταν το επίκεντρο της έρευνας του τομέα της Τεχνητής Νοημοσύνης στην Ιατρική, σπάνια χρησιμοποιήθηκαν πέραν από τα εργαστήρια στα οποία αναπτύχθηκαν. Η ανάπτυξη ορισμένων από αυτά όμως προχώρησε και πλέον μέρος τους αποτελεί εκπαιδευτικά συστήματα. Ένα παράδειγμα είναι το DXplain, που σχεδιάστηκε για να βοηθάει στη διαδικασία της διάγνωσης, λαμβάνοντας ως είσοδο ένα σύνολο κλινικών ευρυμάτων όπως συμπτώματα και δεδομένα του εργαστηρίου και έδινε ως έξοδο μία ιεραρχημένη λίστα πιθανών διαγνώσεων μαζί με αιτιολόγηση για κάθε διαφορετική διάγνωση.

Με τη χρήση αναγνώρισης εικόνων και την ερμηνεία τους, εικόνες είτε απλές όπως για παράδειγμα ακτινογραφίες ή πιο περίπλοκες όπως αγγειογραφίες, αξονικές τομογραφίες και μαγνητικές τομογραφίες μπορούν να ερμηνευθούν από ένα σύστημα και να σηματοδοτηθούν οι εικόνες που πιθανώς να αποκλίνουν από το φυσιολογικό ώστε να εξετασθούν περαιτέρω από κάποιον ειδικό.

Στην οργάνωση και αξιολόγηση κάποιας θεραπευτικής διαδικασίας, συστήματα με Τεχνητή Νοημοσύνη μπορούν να ελέγξουν τη συνιστώμενη θεραπεία για ελλείψεις και λάθη ή να βοηθήσουν στη δημιουργία ενός πλάνου θεραπείας βασισμένο σε ειδικές ανάγκες του ασθενή και σε συμφωνία με τις αποδεκτές θεραπευτικές οδηγίες.

Ορισμένα συστήματα υποστήριξης αποφάσεων χρησιμοποιούνται επίσης στις μονάδες βραχείας νοσηλείας για την αυτόματη λειτουργία ιατρικών συσκευών, όπως το SmartCare/PS που είναι ένα σύστημα βασισμένο σε κανόνες για τη διαχείριση της μηχανικής υποστήριξης της αναπνοής στις μονάδες εντατικής θεραπείας και χρησιμοποιείται από το 2004 έως και σήμερα.

Τα έμπειρα συστήματα είναι επίσης πολύ διαδεδομένα στα κλινικά εργαστήρια, όπου αναλύουν αναφορές από δεδομένα που λαμβάνουν, τα ερμηνεύουν και εξάγουν συμπεράσματα, όπως το GermWatcher που ελέγχει για νοσοκομειακές λοιμώξεις, παρακολουθώντας δεδομένα μικροβιολογικών καλλιεργειών από το εργαστηριακό σύστημα

του νοσοκομείου και ελέγχοντας μία βάση κανόνων που περιέχει ένα συνδυασμό εθνικών κριτηρίων και κριτηρίων της πολιτικής υγείας και μολύνσεων του τοπικού νοσοκομείου.

Στον τομέα της έρευνας πάνω στην ιατρική, τα συστήματα με Τεχνητή Νοημοσύνη μπορούν να βοηθήσουν καθώς, έχοντας τη δυνατότητα να μάθουν, μπορούν να συνεισφέρουν στην ανακάλυψη νέων φαινομένων και τη δημιουργία νέων γνώσεων πάνω στην ιατρική. Για παράδειγμα, στις περιπτώσεις όπου υπάρχει μεγάλος όγκος δεδομένων τα οποία δεν φαίνεται να είναι εύκολο να κατηγοριοποιηθούν από έναν ειδικό, συστήματα Τεχνητής Νοημοσύνης μπορούν και αναλύουν τα δεδομένα και βρίσκουν πιο περίπλοκα μοτίβα με απρόσμενες συσχετίσεις μεταξύ τους. Ακόμη μπορούν να αναλύσουν προϋπάρχοντα μοντέλα και να δείξουν πώς οι πειραματικές παρατηρήσεις αντιτίθενται στις υπάρχουσες θεωρίες.

Ένα παράδειγμα αυτής της χρήσης της μηχανικής μάθησης στην ιατρική αποτελεί το KARDIO, το οποίο έχει αναπτυχθεί για να ερμηνεύει ηλεκτροκαρδιογραφήματα, παίρνοντας βάσεις γνώσης έμπειρων συστημάτων ως παραδείγματα και παράγοντας μία συστηματική περιγραφή των κλινικών χαρακτηριστικών που εκφράζουν μοναδικά συγκεκριμένες παθήσεις.

Με παρόμοιο τρόπο αυτή η προσέγγιση εκτείνεται σε περιοχές της ιατρικής για τις οποίες δεν έχουμε αρκετές πληροφορίες ακόμα. Με τη χρήση συστημάτων Τεχνητής Νοημοσύνης είναι δυνατόν να σχεδιάσουμε αυτόματα μοντέλα παθοφυσιολογίας που να περιγράφουν τις συσχετίσεις διαφορετικών μετρήσεων για τις οποίες προηγουμένως δεν ήταν εμφανής κάποια κοινή κατηγοριοποίηση.

Στη δημιουργία νέων φαρμάκων, τα συστήματα Τεχνητής Νοημοσύνης βοηθούν λαμβάνοντας ως παραδείγματα ένα ή περισσότερα φάρμακα που παρουσιάζουν μία συγκεκριμένη δράση και με βάση την περιγραφή της χημικής δομής της ένωσης εξάγουν συμπεράσματα για το ποιά χημικά χαρακτηριστικά είναι απαραίτητα για τη συγκεκριμένη δράση. Οι επιστήμονες μετά μπορούν με τα αποτελέσματα που πήραν από το σύστημα να συνθέσουν τη νέα ένωση που να έχει εκείνα τα χαρακτηριστικά, μειώνοντας το χρόνο παρασκευής ανάλογων ενώσεων των φαρμάκων πάνω στις οποίες θα έπρεπε να πειραματιστούν (King, R. et al., 1992)

Είναι εύκολα αντιληπτό λοιπόν ότι η Τεχνητή Νοημοσύνη έχει πολλές και σημαντικές εφαρμογές στον τομέα της ιατρικής και της παροχής υπηρεσιών υγείας. Οι κύριοι λόγοι για τους οποίους έχει υπάρξει δυσκολία στην εισαγωγή της στους τομείς αυτούς είναι καθώς, ειδικά στις αρχές, δεν είχε υπάρξει η κατάλληλη ενσωμάτωση των συστημάτων αυτών στην κλινική πρακτική καθώς εστίαζε στην επίλυση προβλημάτων που δεν θεωρούνταν ότι είχαν κάποια σοβαρή επίπτωση στην παροχή υπηρεσιών υγείας ή επέβαλλαν εξαναγκαστικές αλλαγές στον τρόπο λειτουργίας των ειδικών στον τομέα της υγείας οι οποίες ήταν χρονοβόρες ή δαπανηρές. Καθώς πλέον έχουν εντοπιστεί αυτά τα προβλήματα και έχει γίνει μία ορθή αξιολόγηση της σημασίας της Τεχνητής Νοημοσύνης στην Ιατρική, οι ειδικοί στην ανάπτυξη συστημάτων Τεχνητής Νοημοσύνης θα πρέπει να εστιάζουν στον ακριβή προσδιορισμό των σημείων του τομέα της ιατρικής που μπορούν να επωφεληθούν από συστήματα Τεχνητής Νοημοσύνης.

2.2 Σακχαρώδης Διαβήτης και Τεχνητή Νοημοσύνη

Ο Σακχαρώδης Διαβήτης είναι η ασθένεια που μπορεί να χαρακτηριστεί και ως η μάστιγα της εποχής, με το ετήσιο οικονομικό κόστος του διαγνωσμένου Διαβήτη στις Ηνωμένες Πολιτείες Αμερικής να φτάνει τα \$245 δισεκατομμύρια το 2012, αυξημένο κατά 41% από το 2007 όπου είχε φτάσει τα \$174 εκατομμύρια (American Diabetes Association, 2013). Ο Σακχαρώδης Διαβήτης αποτελεί ένα σύνδρομο, δηλαδή ένα σύνολο διαταραχών που έχουν ως χαρακτηριστικό τους την υπεργλυκαιμία και τη δυσανεξία στη γλυκόζη είτε λόγω ανεπάρκειας ινσουλίνης ή λόγω μειωμένης αποτελεσματικότητας της δράσης της ινσουλίνης ή συνδυασμού αυτών.

Ο Σακχαρώδης Διαβήτης μπορεί να κατηγοριοποιηθεί σύμφωνα με τα αίτια που τον προκαλούν (DeFronzo, R., et al. 2015). Χωρίζεται λοιπόν σε:

- Διαβήτη Τύπου 1, που περιλαμβάνει τις περιπτώσεις οι οποίες αποδίδονται σε μία αυτοάνοση διαδικασία καθώς και εκείνες στις οποίες υπάρχει καταστροφή των β-κυττάρων στο πάγκρεας για την οποία δεν είναι γνωστά τα αίτια.
- Διαβήτη Τύπου 2, που περιλαμβάνει το πιο κοινό είδος Διαβήτη που είναι αποτέλεσμα ανεπάρκειας στην έκκριση ινσουλίνης, αντίστασης στην ινσουλίνη ή συνδυασμό των δύο.
- Σακχαρώδης Διαβήτης της Κύησης, που είναι μία κατάσταση δυσανεξίας στη γλυκόζη που αναγνωρίζεται αρχικά κατά τη διάρκεια της κύησης και συνήθως φεύγει μετά τον

τοκετό αλλά συσχετίζεται με αυξημένο κίνδυνο μακροπρόθεσμης εμφάνισης Διαβήτη Τύπου 2.

Η ινσουλίνη είναι η βασική ορμόνη που ρυθμίζει την πρόσληψη γλυκόζης από το αίμα στα κύτταρα του σώματος (Alberts, B. et al. 2010). Οπότε η ανεπάρκεια της ινσουλίνης ή η έλλειψη της ικανότητας των υποδοχέων της να αντιδράσουν σε αυτή παίζει ένα σημαντικό ρόλο σε όλους τους τύπους Διαβήτη.

Το σώμα λαμβάνει τη γλυκόζη από τρία κύρια σημεία: την απορρόφηση φαγητού από το έντερο, την αποδόμηση του γλυκογόνου (γλυκογονόλυση) και τη δημιουργία της από μη-υδατανθρακικές αρχικές ενώσεις (γλυκονεογένεση). Η ινσουλίνη παίζει σημαντικό ρόλο καθώς μπορεί και ισορροπεί τα επίπεδα της γλυκόζης στο σώμα, καταστέλοντας την αποδόμηση του γλυκογόνου ή τη διαδικασία της γλυκονεογένεσης και διεγείρει την αποθήκευση της γλυκόζης στη μορφή του γλυκογόνου.

Η ινσουλίνη έρχεται στην κυκλοφορία του αίματος μέσω των β-κυττάρων που βρίσκονται στο πάγκρεας, σαν αντίδραση στα αυξανόμενα επίπεδα της γλυκόζης του αίματος. Αν τα επίπεδα της διαθέσιμης ινσουλίνης δεν είναι επαρκή ή αν τα κύτταρα δεν αντιδρούν όπως θα έπρεπε στην επίδραση της ινσουλίνης (αντίσταση στην ινσουλίνη) ή αν η ίδια η ινσουλίνη είναι ελαττωματική τότε η γλυκόζη δεν θα απορροφηθεί σωστά από τα κύτταρα του σώματος που την χρειάζονται και δεν θα αποθηκευθεί σωστά στο συκώτι και στου μύες. Έτσι τα επίπεδα της γλυκόζης του αίματος παραμένουν υψηλά προκαλώντας τα προβλήματα που παρατηρούνται σε πάσχοντες από Σακχαρώδη Διαβήτη.

Ορισμένες από τις επιπλοκές που δημιουργούνται από τη μακροχρόνια αδυναμία ρύθμισης της γλυκόζης του αίματος είναι

- η αύξηση πιθανότητας εμφάνισης καρδιαγγειακών παθήσεων όπως στεφανιαία νόσος, έμφραγμα, εγκεφαλικό επεισόδιο, αθηροσκλήρωση
- νευροπάθεια, καθώς τα υψηλά επίπεδα γλυκόζης βλάπτουν τα τοιχώματα των τριχοειδών αγγείων που συντηρούν τα νεύρα, ειδικά στα πόδια, δημιουργώντας μούδιασμα και πόνο και αν δεν αντιμετωπισθεί μπορεί να φτάσει σε απώλεια αισθήσεων στα άκρα
- νεφροπάθεια, φτάνοντας μέχρι και σε νεφρική ανεπάρκεια
- παθήσεις των ματιών που μπορούν να φτάσουν σε τύφλωση και αυξημένη πιθανότητα εμφάνισης καταρράκτη και γλαυκώματος

- προβλήματα στα πόδια, καθώς η νευρική βλάβη ή η πτωχή ροή του αίματος στα πόδια αυξάνουν τις πιθανότητες εμφάνισης επιπλοκών όπως πληγές που δεν θεραπεύονται και μπορούν να οδηγήσουν σε ακρωτηριασμό
- παθήσεις του δέρματος, καθώς άτομα με Διαβήτη είναι πιο ευάλωτα σε προβλήματα του δέρματος συμπεριλαμβανομένου βακτηριακών και μυκητιακών μολύνσεων
- προβλήματα στην ακοή, που είναι πιο συχνά σε άτομα με Διαβήτη
- Νόσος Alzheimer's, καθώς άτομα με Διαβήτη τύπου 2 έχουν μεγαλύτερες πιθανότητες για εμφάνιση της νόσου, με μεγαλύτερο κίνδυνο να έχουν όσοι δεν έχουν καλή ρύθμιση των επιπέδων γλυκόζης στο αίμα

Ο Σακχαρώδης Διαβήτης είναι μία χρόνια πάθηση για την οποία δεν υπάρχει ακόμα κάποια θεραπεία. Η αντιμετώπισή του βασίζεται σε τεχνικές διατήρησης των τιμών της γλυκόζης του αίματος κοντά στα φυσιολογικά επίπεδα προσπαθώντας ταυτόχρονα να αποφευχθεί υπογλυκαιμία (χαμηλή τιμή γλυκόζης στο αίμα).

Οπότε η καλύτερη λύση στην επιδημία του Διαβήτη είναι η πρόληψη, κάτι που μπορεί να γίνει εφικτό μόνο μέσω της πρόβλεψης της πιθανότητας εμφάνισης Διαβήτη.

Η πρόβλεψη της πιθανότητας εμφάνισης Διαβήτη μπορεί να γίνει με απλούς στατιστικούς αλγορίθμους ή, για να είναι πιο ακριβής, με τη χρήση μεθόδων Τεχνητής Νοημοσύνης.

Αρχικά γίνεται μία επιλογή των παραγόντων που επηρεάζουν την πιθανότητα εμφάνισης Διαβήτη και έπειτα συλλέγονται δεδομένα πάνω στους παράγοντες αυτούς τόσο από υγιή άτομα όσο και από πάσχοντες από Διαβήτη (train set). Έπειτα αναπτύσσονται μοντέλα κατηγοριοποίησης βασισμένα στα δεδομένα μέσω τεχνικών μηχανικής μάθησης, όπως τα νευρωνικά δίκτυα και οι μηχανές υποστήριξης διανυσμάτων, και συγκρίνεται η απόδοση καθενός έπειτα από πολλές τροποποιήσεις ώστε να βρεθεί το μοντέλο που δίνει το μικρότερο δυνατό σφάλμα στα δεδομένα που υπάρχουν. Τέλος ελέγχεται η απόδοση των υποθετικών μοντέλων με τη χρήση διαφορετικών συνόλων δεδομένων (test set).

Οι παράγοντες που συνήθως λαμβάνονται υπ' όψιν σε ένα μοντέλο πρόβλεψης της πιθανότητας εμφάνισης Διαβήτη είναι:

- το φύλο
- η ηλικία, καθώς οι μεγαλύτερες ηλικίες έχουν υψηλότερα ποσοστά εμφάνισης Διαβήτη

- η συγκέντρωση γλυκόζης στο αίμα έπειτα από κάποια προσδιορισμένη και κοινή εξέταση
- η πίεση του αίματος, καθώς άτομα με υψηλή αρτηριακή πίεση έχουν συνήθως περισσότερες πιθανότητες εμφάνισης Διαβήτη
- η σωματική κατάσταση, καθώς άτομα με παχυσαρκία ή υψηλό δείκτη μάζας σώματος (Body Mass Index ή BMI) είναι σε μεγαλύτερο κίνδυνο για την εμφάνιση Σακχαρώδους Διαβήτη
- το οικογενειακό ιστορικό Διαβήτη, καθώς άτομα που έχουν συγγενικά πρόσωπα που πάσχουν από Διαβήτη έχουν, λόγω γενετικής, μεγαλύτερη προδιάθεση να εμφανίσουν Διαβήτη
- αν γίνεται αναφορά σε κοπέλες, ο αριθμός των κύσεων παίζει σημαντικό ρόλο καθώς αυξάνει την πιθανότητα εμφάνισης Διαβήτη της κύησης και συνεπώς και Διαβήτη τύπου 2 αργότερα

Τα δεδομένα αυτά μπορούν σε συνδυασμό και με άλλα να βοηθήσουν στην ανάπτυξη μοντέλων πρόβλεψης της πιθανότητας εμφάνισης του Σακχαρώδους Διαβήτη μέσω τεχνικών Μηχανικής Μάθησης.

Κεφάλαιο 3:

Παραδείγματα Σύγχρονων Μεθόδων Μηχανικής Μάθησης

3.1 Εισαγωγή στις έννοιες που θα χρησιμοποιηθούν

Ένα σύστημα μηχανικής μάθησης καλείται να λύσει προβλήματα βασισμένο σε κάποια δεδομένα που υπάρχουν. Οι τεχνικές μάθησης ανάλογα με το είδος του προβλήματος μπορούν να χωριστούν σε δύο βασικές κατηγορίες:

- i) **Μάθηση με επίβλεψη** (Supervised Learning), όπου το σύστημα “μαθαίνει” μία έννοια ή συνάρτηση από ένα σύνολο δεδομένων που αποτελεί την περιγραφή ενός μοντέλου και ο “επιβλέπων” παρέχει στο σύστημα τις σωστές τιμές εξόδου της συνάρτησης για τα δεδομένα
- ii) **Μάθηση χωρίς επίβλεψη** (Unsupervised Learning), όπου το σύστημα καλείται να “μάθει” ανακαλύπτοντας από μόνο του συσχετίσεις ή ομάδες ανάμεσα σε ένα σύνολο δεδομένων, δημιουργώντας πρότυπα χωρίς να είναι γνωστό το αν υπάρχουν ή το πόσα και ποια είναι

Πολύ συχνά χρησιμοποιείται και η τεχνική της **Ενισχυτικής Μάθησης** (Reinforcement Learning) στην οποία το σύστημα καλείται να μάθει από την αλληλεπίδραση με το περιβάλλον, προσπαθώντας να μεγιστοποιήσει μία συνάρτηση του σήματος ενίσχυσης (reward, ανταμοιβή) δηλαδή μαθαίνει βελτιώνοντας την απόδοσή του μέσω δοκιμής και αποτυχίας (trial and error).

Στη μάθηση με επίβλεψη το σύστημα πρέπει να μπορέσει να μάθει κάποια συνάρτηση στόχο (target function), που συμβολίζεται με c , η οποία να αποτελεί έκφραση του μοντέλου που περιγράφει σωστά τα δεδομένα και χρησιμοποιείται για την πρόβλεψη (prediction) της τιμής μιας μεταβλητής που ονομάζεται εξαρτημένη μεταβλητή ή μεταβλητή εξόδου $y^{(i)}$ σύμφωνα με ένα σύνολο ανεξάρτητων μεταβλητών ή μεταβλητών εισόδου $x^{(i)}$, όπου το (i) είναι δείκτης και όχι εκθέτης.

Η συνάρτηση έχει ως είσοδο ένα σύνολο διαφορετικών μεταβλητών που είναι το πεδίο ορισμού της και ονομάζεται σύνολο των περιπτώσεων ή στιγμιοτύπων (instances) και συμβολίζεται με X . Κάθε περίπτωση περιγράφεται από ένα σύνολο χαρακτηριστικών (features ή attributes). Το υποσύνολο των περιπτώσεων για το οποίο γνωρίζουμε την τιμή της μεταβλητής εξόδου, δηλαδή το σύνολο από ζεύγη $(x^{(i)}, y^{(i)})$ ονομάζεται σύνολο δεδομένων εκπαίδευσης ή παραδείγματα (training set) και κάθε ζεύγος $(x^{(i)}, y^{(i)})$ ονομάζεται παράδειγμα

εκπαίδευσης (training example). Αν m είναι ο αριθμός των παραδειγμάτων που υπάρχουν τότε το σύνολο δεδομένων εκπαίδευσης (training set) θα είναι

$$\{(x^{(i)}, y^{(i)}); i = 1, \dots, m\}$$

Κατά την προσπάθεια προσέγγισης της συνάρτησης στόχου c , το σύστημα εξετάζει εναλλακτικές συναρτήσεις που ονομάζονται υποθέσεις και συμβολίζονται με h ή h_θ . Το σύνολο των υποθέσεων που πρέπει να εξετασθούν ώστε να βρεθεί η συνάρτηση στόχος συμβολίζεται με H .

Αν θεωρηθεί ότι η μεταβλητή y είναι γραμμικώς εξαρτημένη από τη x , τότε η αναπαράσταση της υπόθεσης μπορεί να γίνει ως εξής:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

με θ_i να είναι οι παράμετροι (ή βάρη) και x_1, x_2 να είναι τα χαρακτηριστικά (features). Για n χαρακτηριστικά και αν θεωρηθεί πως $x_0 = 1$ η παράσταση μπορεί να γραφεί ως:

$$h_\theta(x) = \sum_{i=0}^n \theta_i x_i = \theta^T x$$

Μέσω μίας συνάρτησης κόστους (Cost Function) J γίνεται μια προσπάθεια προσέγγισης των τιμών των παραμέτρων θ που φέρνουν όλες τις τιμές $h(x)$ κοντά στις τιμές των μεταβλητών y και άρα να βρεθεί η συνάρτηση h που να προσεγγίζει τη συνάρτηση c .

Η λειτουργία της μάθησης σε αυτή την περίπτωση στηρίζεται στην υπόθεση της μάθησης με επαγωγή, η οποία περιγράφει πως κάθε συνάρτηση h που έχει βρεθεί να προσεγγίζει καλά τη συνάρτηση στόχο για ένα μεγάλο σύνολο παραδειγμάτων θα προσεγγίζει το ίδιο καλά τη συνάρτηση στόχο και τις περιπτώσεις που δεν έχει ακόμα εξετάσει.

Στη μάθηση με επίβλεψη μπορούν να διακριθούν δύο είδη προβλημάτων:

- i) **Ταξινόμηση (Classification)**, που αφορά τη δημιουργία μοντέλων πρόβλεψης διακριτών τάξεων, δηλαδή η μεταβλητή y θα παίρνει μόνο έναν μικρό αριθμό διακριτών τιμών
- ii) **Παρεμβολή ή Παλινδρόμηση (Regression)**, που αφορά τη δημιουργία πρόβλεψης αριθμητικών τιμών, δηλαδή η μεταβλητή y θα είναι συνεχής.

Οι πιο συνηθισμένες τεχνικές μηχανικής μάθησης με επίβλεψη είναι:

- i) Linear Regression, Γραμμική Παρεμβολή ή Γραμμική Παλινδρόμηση
- ii) Logistic Regression, Λογιστική Παλινδρόμηση, που αποτελεί βελτίωση της μεθόδου γραμμικής παλινδρόμησης
- iii) K-NN Algorithm ή K-Nearest Neighbours Algorithm
- iv) Support Vector Machines, SVMs ή Μηχανές Διανυσμάτων Υποστήριξης
- v) Neural Networks, Νευρωνικά Δίκτυα

Κατά τη δημιουργία ενός συστήματος μηχανικής μάθησης είναι σημαντικό να ξεκινήσει κανείς από έναν απλό αλγόριθμο κατάλληλο για το πρόβλημα που καλείται να επιλύσει και να τον ελέγξει. Ο έλεγχος γίνεται με το διαχωρισμό του αρχικού συνόλου δεδομένων εκπαίδευσης (train set) σε σύνολο διασταυρωμένης επικύρωσης (Cross Validation Set) και σύνολο δεδομένων εκπαίδευσης (Train Set) και το σχεδιασμό των καμπυλών μάθησης που δείχνουν την απόδοση του αλγορίθμου τόσο στο σύνολο εκπαίδευσης όσο και στο σύνολο διασταυρωμένης επικύρωσης.

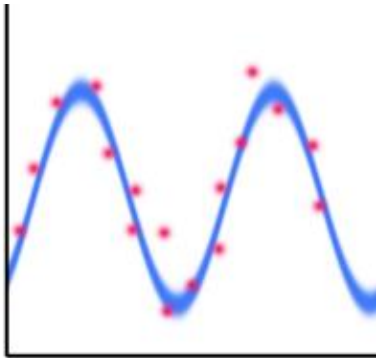
Σχεδιάζοντας της καμπύλες μάθησης είναι δυνατόν να γίνει ο εντοπισμός τυχόν προβλημάτων του αλγορίθμου υλοποίησης, όπως το πρόβλημα της υψηλής διακύμανσης (high variance) που μπορεί να επιφέρει τη λεγόμενη υπερπροσαρμογή (overfit) των δεδομένων, δηλαδή ο αλγόριθμος να προσαρμόζεται υπερβολικά στα δεδομένα του συνόλου εκπαίδευσης και να αδυνατεί να λειτουργήσει το ίδιο αποδοτικά κατά τη γενίκευση (generalization) στα δεδομένα του συνόλου επικύρωσης CV, και το πρόβλημα της υψηλής μεροληψίας (high bias) που μπορεί να επιφέρει υποπροσαρμογή (underfit) των δεδομένων, δηλαδή το λάθος κατά τη διαδικασία της γενίκευσης (generalization) να είναι πολύ μεγάλο και να υπάρχουν σφάλματα ακόμα και κατά την εφαρμογή του αλγορίθμου στο σύνολο δεδομένων εκπαίδευσης (train set).

Το πρόβλημα της υψηλής διακύμανσης (variance) μπορεί να επιλυθεί συνήθως προσθέτοντας περισσότερα δεδομένα εκπαίδευσης (training examples), χρησιμοποιώντας μικρότερα σύνολα χαρακτηριστικών (features), ή τροποποιώντας κατάλληλες παραμέτρους του εκάστοτε αλγορίθμου.

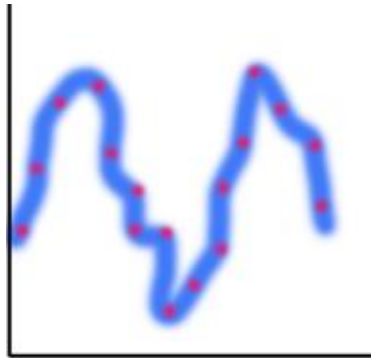
Το πρόβλημα της υψηλής μεροληψίας (bias) μπορεί να αντιμετωπισθεί με την προσθήκη περισσότερων χαρακτηριστικών (features), με την προσθήκη χαρακτηριστικών με πολυώνυμα (polynomial features) ή τροποποιώντας κατάλληλες παραμέτρους του εκάστοτε αλγορίθμου.

Οι εικόνες 3.1.α, 3.1.β, 3.1.γ δείχνουν σχηματικά τις έννοιες Overfitting και Underfitting σε προβλήματα Παρεμβολής:

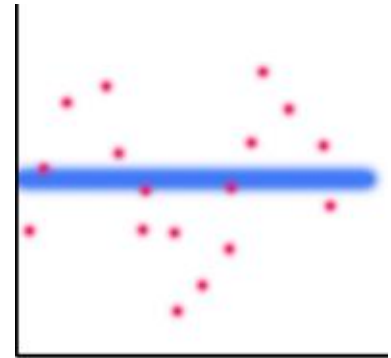
α) Κανονική Προσαρμογή



β) Overfitting



γ) Underfitting



Ενώ οι εικόνες 3.1.δ, 3.1.ε, 3.1.στ δείχνουν τις έννοιες Overfitting και Underfitting σε προβλήματα Ταξινόμησης:

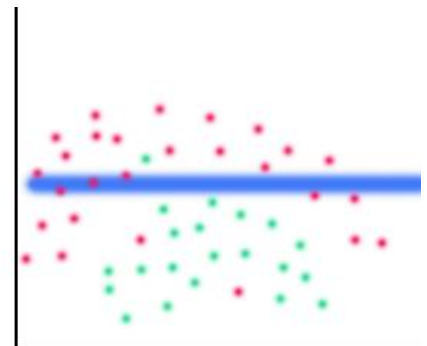
δ) Κανονική Προσαρμογή



ε) Overfitting



στ) Underfitting



Όπως φαίνεται και στις σχηματικές αναπαραστάσεις, κατά την κανονική προσαρμογή υπάρχει περίπτωση να υπάρχουν αποκλίσεις όμως το μοντέλο θα ανταποκριθεί κατάλληλα σε μελλοντικές εισόδους (test set) λόγω σωστής γενίκευσης ενώ αν προσαρμοστεί επακριβώς στο σύνολο δεδομένων εκπαίδευσης όπως γίνεται στην περίπτωση του Overfitting θα υπάρξει υψηλό σφάλμα κατά την απόκρισή του σε νέες μελλοντικές εισόδους τις οποίες θα κληθεί να προβλέψει. Στην περίπτωση του Underfitting θα υπάρξει υψηλή τιμή σφάλματος τόσο στα δεδομένα εκπαίδευσης όσο και σε μελλοντικά δεδομένα.

Απαραίτητη είναι επίσης η εκτίμηση της απόδοσης ενός μοντέλου η οποία γίνεται με διαφορετικές μεθόδους. Στη δυαδική ταξινόμηση (ταξινόμηση των δεδομένων σε δύο κατηγορίες) χρησιμοποιούνται συχνά οι έννοιες True Positive, True Negative, False Positive και False Negative. Αν θεωρηθεί ότι οι κατηγορίες στις οποίες τοποθετούνται τα στοιχεία είναι Positive (Θετικά) και Negative (Αρνητικά), τότε μπορεί να οριστεί ως **True Positive** (TP, Αληθή Θετικά) ο αριθμός των εξόδων που κατηγοριοποιήθηκαν ως θετικά από το σύστημα και ήταν όντως θετικά, ως **True Negative** (TN, Αληθή Αρνητικά) ο αριθμός των εξόδων που κατηγοριοποιήθηκαν ως αρνητικά και ήταν όντως αρνητικά, **False Positive** (FP, Ψευδή Θετικά) ο αριθμός των εξόδων που κατηγοριοποιήθηκαν ως θετικά ενώ ήταν κανονικά αρνητικά και ως **False Negative** (FN, Ψευδή Αρνητικά) ο αριθμός των εξόδων που κατηγοριοποιήθηκαν ως αρνητικά ενώ στην πραγματικότητα ήταν θετικά.

Αυτές οι τέσσερις έννοιες μπορούν να τοποθετηθούν σε ένα πίνακα Συνάφειας (Confusion Matrix) όπως ο Πίνακας 3.1.1:

		Τιμές Πρόβλεψης	
		Θετικά	Αρνητικά
Πραγματικές Τιμές	Θετικά	TP	FN
	Αρνητικά	FP	TN

Πίνακας 3.1.1 Confusion Matrix

Το άθροισμα των τιμών που βρίσκονται στη διαγώνιο του πίνακα (TP + TN) αποτελεί το σύνολο των στοιχείων που κατηγοριοποιήθηκαν σωστά. Έτσι αποτελεί ένα εργαλείο που εύκολα δείχνει την ικανότητα ταξινόμησης των αποτελεσμάτων του συστήματος. Ένα θεωρητικό σύστημα που θα είχε κατηγοριοποιήσει σωστά όλες τις τιμές, θα έδινε ένα διαγώνιο πίνακα, δηλαδή θα υπήρχε 0 σε κάθε άλλη θέση πλην της διαγωνίου

$$(a_{ij} = 0 \forall i \neq j)$$

Μέσω του πίνακα είναι εύκολο να υπολογιστούν τα παρακάτω:

- Ορθότητα (Accuracy): $\frac{TP+TN}{\text{σύνολο}} = \frac{TP+TN}{TP+TN+FP+FN}$
- Ακρίβεια (Precision): $\frac{TP}{\text{προβλεφθέντα θετικά}} = \frac{TP}{TP+FP}$
- Ανάκληση (Recall) ή Αληθής Θετικός Ρυθμός (True Positive Rate - TPR) ή Ευαισθησία (Sensitivity): $\frac{TP}{\text{πραγματικά θετικά}} = \frac{TP}{TP+FN}$
- Ειδικότητα (Specificity) ή True Negative Rate (TNR): $\frac{TN}{TN+FP}$
- Ψευδής Θετικός Ρυθμός (False Positive Rate - FPR): $\frac{FP}{\text{πραγματικά αρνητικά}} = \frac{FP}{FP+TN}$
- F₁ score ή F-measure: $2 \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$
- Matthews Correlation Coefficient (MCC): $\frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

Η Καμπύλη ROC (Καμπύλη Λειτουργικών Χαρακτηριστικών ή Receiver Operating Characteristics - ROC) είναι μία γραφική παράσταση που απεικονίζει τους συνδυασμούς της αναλογίας των ψευδών θετικών περιπτώσεων και της ευαισθησίας και έχει στον οριζόντιο άξονα (x) τον Ψευδή Θετικό Ρυθμό FPR ενώ στον κάθετο άξονα (y) τον Αληθή Θετικό Ρυθμό TPR. Ορίζεται στο μοναδιαίο τετράγωνο [0,1]x[0,1] ξεκινώντας από το σημείο (0,0) και καταλήγοντας στο σημείο (1,1).

Ένα θεωρητικό μοντέλο πρόβλεψης που θα μπορούσε να θεωρηθεί ως το βέλτιστο θα έδινε σημείο εντός της καμπύλης στην πάνω αριστερή γωνία, δηλαδή το σημείο με συντεταγμένες (0,1) κάτι που θα σήμαινε πως έχει 100% Ανάκληση (Recall) και 100% Ειδικότητα (Specificity), καθώς ο Ψευδής Θετικός Αριθμός FPR μπορεί να οριστεί και ως (1- Ειδικότητα).

Σημεία επάνω στη διαγώνιο που αντιπροσωπεύει την εξίσωση $y=x$ αντιστοιχούν σε εντελώς τυχαία πρόβλεψη και άρα καμία προγνωστική ικανότητα. Σημεία πάνω από τη διαγώνιο προς την άνω αριστερή γωνία του διαγράμματος αντιπροσωπεύουν καλά αποτελέσματα ταξινόμησης καθώς έχουν χαμηλό αριθμό ψευδών θετικών περιπτώσεων και υψηλή ευαισθησία (Ντζούφρας, Ι. & Περπέρογλου, Α., 2009). Αντίθετα σημεία κάτω από τη διαγώνιο αντιπροσωπεύουν αποτελέσματα που είναι χειρότερα από τυχαία πρόβλεψη. Γραφικά, όσο πιο κυρτωμένη είναι η καμπύλη προς το άνω αριστερό μέρος του γραφήματος, τόσο καλύτερο είναι το μοντέλο πρόβλεψης (Ασλάνη, Μ. 2013).

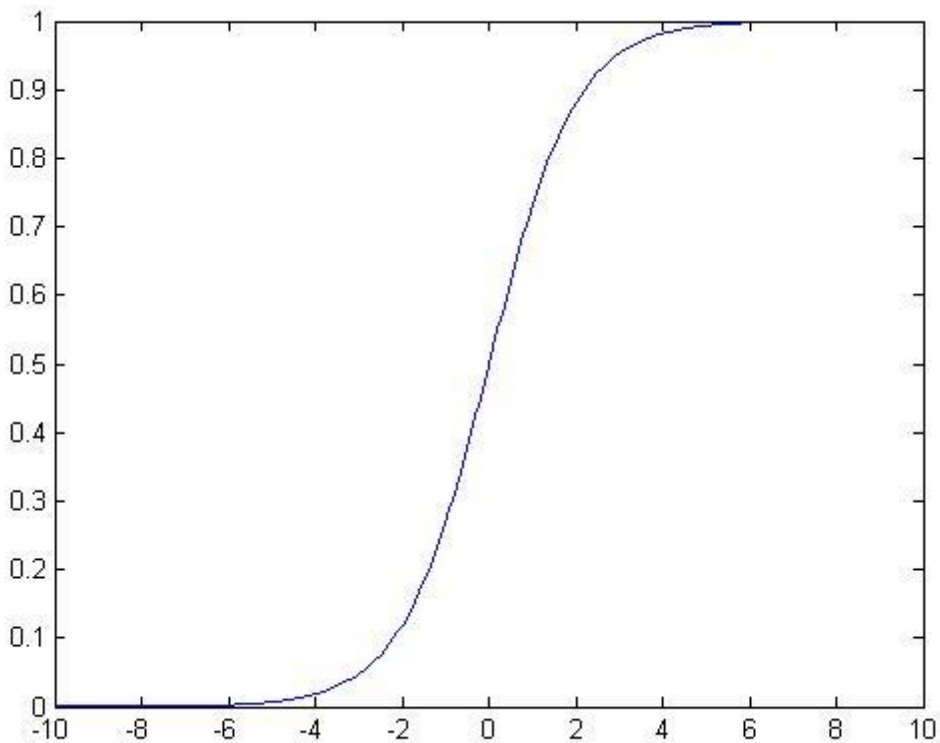
Η Περιοχή Κάτω από την Καμπύλη (AUC - Area Under the Curve) είναι το εμβαδόν κάτω από την καμπύλη ROC, δηλαδή το εμβαδόν που περικλείεται μεταξύ της καμπύλης και του άξονα των x . Μετράει την απόδοση του αλγορίθμου ως ένας αδιάστατος πραγματικός αριθμός με τιμές $0 \leq AUC \leq 1$. Η διαγώνιος που αντιπροσωπεύει την εξίσωση $y=x$ έχει $AUC=0.5$ και αντιστοιχεί σε τυχαία πρόβλεψη οπότε χρησιμοποιείται σαν τιμή αναφοράς για την αξιολόγηση του μοντέλου, που θεωρείται καλύτερο όταν πλησιάζει η τιμή προς το 1.

3.2 Λογιστική Παλινδρόμηση (Logistic Regression)

Η λογιστική παλινδρόμηση (logistic regression) είναι από τους πιο διαδεδομένους αλγορίθμους σήμερα. Το όνομα προέρχεται από την λογιστική συνάρτηση, ή αλλιώς σιγμοειδή συνάρτηση η οποία χρησιμοποιείται στην στατιστική για πληθυσμούς που αυξάνονται γρήγορα μέχρι την χωρητικότητα του περιβάλλοντος. Είναι μια καμπύλη που μπορεί να πάρει οποιοδήποτε αριθμό και να τον μετατρέψει σε μια τιμή μεταξύ του 0 και 1 αλλά ποτέ ακριβώς σε κάποιο από αυτά τα όρια.

Η λογιστική συνάρτηση είναι η εξής:

$$g(z) = \frac{1}{1 + e^{-z}} = \frac{e^z}{e^z + 1}$$



Εικόνα 3.2.α – Απεικόνιση Σιγμοειδούς Συνάρτησης

Χρησιμοποιώντας τη λογιστική παλινδρόμηση, ο τύπος των υποθέσεων $h(x)$ δεν είναι γραμμικός αλλά της μορφής:

$$h_{\theta}(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

όπου διατηρείται η σύμβαση $x_0 = 1$.

Χρησιμοποιώντας πιθανότητες μπορεί να θεωρηθεί ότι:

$$P(y = 1|x; \theta) = h_{\theta}(x) \quad (3.2.1)$$

Δηλαδή η πιθανότητα ότι $y=1$, δεδομένου x , παραμετροποιημένου με θ είναι ίση με $h_{\theta}(x)$ και καθώς οι πιθανότητες για $y=0$ μαζί με τις πιθανότητες για $y=1$ είναι 1

$$P(y = 0|x; \theta) = 1 - h_{\theta}(x) \quad (3.2.2)$$

οι εξισώσεις 3.2.1 και 3.2.2 μπορούν να γραφούν συνοπτικά ως:

$$P(y = 1|x; \theta) = (h_{\theta}(x))^y (1 - h_{\theta}(x))^{1-y}$$

Το επόμενο βήμα είναι ο υπολογισμός των τιμών των παραμέτρων θ .

Γίνεται αρχικά χρήση μίας συνάρτησης κόστους (Cost Function) για τιμές του y πάντα 0 ή 1 της μορφής

$$Cost(h_{\theta}(x^{(i)}), y^{(i)}) = \begin{cases} -\log(h_{\theta}(x^{(i)})) & \text{αν } y = 1 \\ -\log(1 - (h_{\theta}(x^{(i)}))) & \text{αν } y = 0 \end{cases}$$

που μπορεί να γραφεί ως:

$$Cost(h_{\theta}(x^{(i)}), y^{(i)}) = -y^{(i)} \log(h_{\theta}(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))$$

Και έτσι προκύπτει η συνάρτηση κόστους της λογιστικής παλινδρόμησης με τη μορφή:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m Cost(h_{\theta}(x^{(i)}), y^{(i)}) = \\ -\frac{1}{m} \left[\sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \right]$$

Όπου για τον υπολογισμό των παραμέτρων θ υπολογίζονται οι τιμές που ελαχιστοποιούν τη συνάρτηση $J(\theta)$ δηλαδή $\min_{\theta} J(\theta)$.

Η πιο συχνή συνάρτηση που χρησιμοποιείται για αυτό το σκοπό είναι η Gradient Descent (επικλινής κάθοδος ή αλγόριθμος σύγκλισης με ελάττωση της παραγώγου) όπου υπολογίζονται οι μερικοί παράγωγοι της $J(\theta)$ για κάθε θ_j από :

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial j} J(\theta)$$

που ισοδυναμεί με

$$\theta_j := \theta_j - \alpha \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

Έπειτα οι τιμές του θ_j αλλάζουν κάθε φορά ταυτόχρονα για κάθε θ_j μέχρι να υπολογισθεί η ελάχιστη τιμή της $J(\theta)$ δηλαδή $\min_{\theta} J(\theta)$.

Μία άλλη μέθοδος εύρεσης των παραμέτρων θ είναι με χρήση της πιθανοφάνειας (likelihood) και stochastic gradient ascent. Αν θεωρηθεί ότι τα m διαφορετικά παραδείγματα (training examples) του συνόλου δεδομένων εκπαίδευσης (training set) δημιουργήθηκαν ανεξάρτητα, τότε μπορεί να γραφεί η πιθανοφάνεια των παραμέτρων ως:

$$L(\theta) = p(\vec{y}|\mathbf{X}; \theta) = \prod_{i=1}^m p(y^{(i)}|x^{(i)}; \theta) = \prod_{i=1}^m (h_{\theta}(x^{(i)}))^{y^{(i)}} (1 - h_{\theta}(x^{(i)}))^{1-y^{(i)}}$$

Για λόγους ευκολίας χρησιμοποιείται η λογαριθμική πιθανοφάνεια (logarithmic likelihood)

$$l(\theta) = \log L(\theta) = \sum_{i=1}^m y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))$$

Χρήσιμο είναι επίσης να υπολογισθεί η πρώτη παράγωγος της $g(z)$ που είναι

$$g'(z) = g(z)(1-g(z))$$

Έπειτα για να υπολογιστεί η μέγιστη πιθανοφάνεια χρησιμοποιείται η μέθοδος gradient ascent και, γνωρίζοντας πως $g'(z) = g(z)(1-g(z))$ οι αλλαγές των τιμών θ γίνονται ως εξής:

$$\theta_j := \theta_j + \alpha(y^{(i)} - h_{\theta}(x^{(i)}))x_j^{(i)}$$

Δηλαδή μέσω gradient ascent υπολογίζεται το μέγιστο (max) της συνάρτησης πιθανοφάνειας ενώ μέσω gradient descent υπολογίζεται το ελάχιστο (min) της συνάρτησης κόστους.

3.3 Αλγόριθμος K-NN (K-Nearest Neighbours Algorithm)

Ο αλγόριθμος KNN αποτελεί την πιο απλή μέθοδο μηχανικής μάθησης και λειτουργεί μέσω μάθησης από περιπτώσεις (instance-based learning), δηλαδή αντί για γενίκευση συγκρίνει τα νέα παραδείγματα με όσα έχουν ήδη ελεγχθεί και αποθηκευθεί στη μνήμη. Αυτό που χρειάζεται προσοχή είναι ότι αφού αποθηκεύεται όλο το dataset πρέπει να ανανεώνεται συχνά καθώς εισέρχονται νέα δεδομένα και να αφαιρούνται τα εσφαλμένα.

Ο αλγόριθμος KNN κάνει προβλέψεις ψάχνοντας ολόκληρο το σύνολο δεδομένων εκπαίδευσης (training set) για τις K πιο όμοιες περιπτώσεις (“neighbours”, γείτονες) και υπολογίζει την μεταβλητή εξόδου σύμφωνα με αυτές, που για την παρεμβολή μπορεί να είναι ο μέσος όρος των μεταβλητών εξόδου ενώ για την ταξινόμηση μπορεί να είναι η πιο κοινή τιμή.

Για να καθοριστούν ποιές από τις περιπτώσεις του K στο σύνολο δεδομένων εκπαίδευσης (training dataset) είναι οι πιο όμοιες με μια νέα είσοδο χρησιμοποιούνται διάφοροι τρόποι μέτρησης απόστασης. Για πραγματικές τιμές εισόδου ο πιο δημοφιλής τρόπος είναι η Ευκλείδεια Απόσταση (Euclidean distance).

Η απόσταση αυτή υπολογίζεται από την τετραγωνική ρίζα του αθροίσματος των τετραγώνων των διαφορών μεταξύ του νέου σημείου $x_j^{(a)}$ και ενός υπάρχοντος σημείου $x_j^{(k)}$ για όλες τις τιμές του n αριθμού χαρακτηριστικών j.

$$\sqrt{\sum_{j=1}^n (x_j^{(a)} - x_j^{(k)})^2}$$

Πέρα από την Ευκλείδεια Απόσταση υπάρχουν και άλλοι συχνά χρησιμοποιούμενοι τρόποι όπως η Απόσταση Hamming (Hamming Distance) και η Απόσταση Manhattan (Manhattan Distance).

Η Απόσταση Hamming είναι η απόσταση μεταξύ δύο συμβολοσειρών (string) ίδιου μήκους και ισούται με τον αριθμό των θέσεων κατά τις οποίες δύο αντίστοιχα σύμβολα διαφέρουν. Μπορεί επίσης να διατυπωθεί και ως ο αριθμός των αντικαταστάσεων που πρέπει να γίνουν

για να αλλάξει μία συμβολοσειρά σε μία άλλη. Για παράδειγμα, η απόσταση μεταξύ του 10101110 και 10001010 είναι 2.

Η Απόσταση Manhattan είναι η απόσταση μεταξύ δύο πραγματικών διανυσμάτων και χρησιμοποιεί για τον υπολογισμό της το σύνολο των απόλυτων διαφορών τους

$$\sum_{j=1}^n |x_j^{(a)} - x_j^{(k)}|$$

Η Ευκλείδεια Απόσταση είναι καλή για την μέτρηση μεταβλητών που ανήκουν στον ίδιο τύπο (π.χ ύψος). Η Απόσταση Manhattan είναι καλή για την μέτρηση μεταβλητών που ανήκουν σε διαφορετικό τύπο.

Επίσης υπάρχουν και άλλοι τρόποι μέτρησης της απόστασης (Απόσταση Minkowski, Tanimoto, Jaccard) που όμως δεν είναι τόσο δημοφιλείς.

Για να γίνει υπολογισμός της τιμής του K δοκιμάζονται διάφορες τιμές και ελέγχεται τι λειτουργεί καλύτερα στο κάθε πρόβλημα. Η υπολογιστική πολυπλοκότητα του KNN αυξάνεται με το μέγεθος του συνόλου δεδομένων εκπαίδευσης (training dataset).

Ο αλγόριθμος KNN καθώς χρησιμοποιείται σε διαφορετικούς τομείς αναφέρεται και με πολλά διαφορετικά ονόματα:

- Instance-Based Learning: Επειδή χρησιμοποιούνται οι περιπτώσεις (instances) του συνόλου δεδομένων εκπαίδευσης για να γίνουν προβλέψεις ο KNN αναφέρεται και σαν Instance-Based Learning
- Lazy Learning: Δεν γίνεται γενίκευση πέραν του συνόλου εκπαίδευσης δεδομένων μέχρι να γίνει κάποιο αίτημα τη στιγμή που ζητείται πρόβλεψη. Για αυτό τον λόγο ο KNN αναφέρεται και σαν Lazy Learning αλγόριθμος
- Non-Parametric: Ο KNN δεν κάνει υποθέσεις για την λειτουργική μορφή του προβλήματος. Αυτό του δίνει το όνομα του μη παραμετρικού αλγόριθμου

Ο KNN μπορεί να χρησιμοποιηθεί τόσο για προβλήματα ταξινόμησης (classification) όσο και για προβλήματα παρεμβολής ή παλινδρόμησης (regression).

Στην περίπτωση που χρησιμοποιείται ο KNN για προβλήματα παλινδρόμησης η πρόβλεψη βασίζεται στο μέσο όρο ή την διάμεσο των K πιο όμοιων περιπτώσεων.

Αντίθετα όταν ο αλγόριθμος KNN χρησιμοποιείται για ταξινόμηση, η έξοδος μπορεί να υπολογιστεί σαν η κλάση με την μεγαλύτερη συχνότητα από τις περιπτώσεις του K. Κάθε περίπτωση στην ουσία αυξάνει την κλάση της και η υψηλότερη κλάση επιλέγεται για την πρόβλεψη. Οι προβλέψεις της κλάσης για παράδειγμα σε ένα δυαδικό πρόβλημα ταξινόμησης (όπου η κλάση παίρνει τιμή 0 ή 1) μπορούν να υπολογιστούν ως:

$$p(\text{κλάση} = 0) = \frac{\text{αριθμός}(\text{κλάση} = 0)}{(\text{αριθμός}(\text{κλάση} = 0) + \text{αριθμός}(\text{κλάση} = 1))}$$

όπου με την έννοια “αριθμός” συμβολίζεται ο αριθμός της κανονικοποιημένης συχνότητας των δειγμάτων που ανήκουν στην αντίστοιχη κλάση.

Όταν υπάρχει άρτιος αριθμός κλάσεων είναι προτιμότερο να χρησιμοποιείται περιττό K ώστε να αποφεύγονται ισοψηφίες. Αντίστοιχα όταν υπάρχει περιττός αριθμός κλάσεων καλό είναι να γίνεται χρήση ζυγού αριθμού K. Οι ισοψηφίες μπορούν να διορθωθούν αυξάνοντας το K κατά 1 και κοιτώντας την κλάση της επόμενης πιο όμοιας περίπτωσης στο σύνολο δεδομένων εκπαίδευσης. Όσο αυξάνεται ο αριθμός K τα όρια ταξινόμησης γίνονται πιο ομαλά όμως το σφάλμα εκπαίδευσης μπορεί να αυξηθεί.

Ο αλγόριθμος KNN λειτουργεί καλά με μικρό αριθμό μεταβλητών στην είσοδο αλλά αντιμετωπίζει δυσκολία όταν ο αριθμός εισόδων είναι πολύ μεγάλος. Κάθε μεταβλητή εισόδου μπορεί να θεωρηθεί ως μια διάσταση στον χώρο των εισόδων. Για παράδειγμα αν υπάρχουν δύο είσοδοι τότε ο χώρος των εισόδων θα ήταν δύο διαστάσεων. Όσο αυξάνουν οι διαστάσεις ο όγκος των εισόδων αυξάνεται εκθετικά. Σε μεγάλο αριθμό διαστάσεων, σημεία που μπορεί να είναι ίδια ίσως να έχουν μεγάλες αποστάσεις καθώς όλα τα σημεία θα είναι μακριά το ένα από το άλλο.

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.

3.4 Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines, SVMs)

Ο όρος Support Vector Machines, SVMs ή Μηχανές Διανυσμάτων Υποστήριξης, αναφέρεται σε μοντέλα μάθησης με επίβλεψη με συσχετιζόμενους αλγορίθμους μάθησης που αναλύουν δεδομένα τόσο για ταξινόμηση (classification) όσο και για παρεμβολή ή παλινδρόμηση (regression). Η λειτουργία τους στηρίζεται στη θεωρία στατιστικής μάθησης και στα νευρωνικά δίκτυα τύπου Perceptron. Αποτελούν την καλύτερη επιλογή για εφαρμογές αναγνώρισης γραφής και ταξινόμησης κειμένων.

Για την ταξινόμηση τα συστήματα SVMs χρησιμοποιούν ταξινομητή μεγίστου περιθωρίου (Maximal Margin Classifier) ώστε να βρουν μία υπερεπιφάνεια (hypersurface) που να διαχωρίζει στο χώρο των παραδειγμάτων τα αρνητικά από τα θετικά με τρόπο ώστε να διαχωρίζονται με το μέγιστο περιθώριο δηλαδή να απέχει η επιφάνεια αυτή όσο γίνεται περισσότερο από τα κοντινότερα θετικά και αρνητικά παραδείγματα.

Στην περίπτωση ταξινόμησης με μέγιστο περιθώριο, οι αριθμητικές μεταβλητές εισόδου των δεδομένων σχηματίζουν ένα χώρο n διαστάσεων. Για παράδειγμα, δύο μεταβλητές στην είσοδο θα σχημάτιζαν έναν χώρο δύο διαστάσεων. Το υπερεπίπεδο είναι μια γραμμή που διαχωρίζει τον χώρο των μεταβλητών εισόδου. Στο SVM ένα υπερεπίπεδο επιλέγεται για να διαχωρίσει τα σημεία στον χώρο των μεταβλητών εισόδου ανάλογα με την κλάση τους, κλάση 0 ή κλάση 1. Με δύο διαστάσεις γίνεται να σχεδιαστεί αυτή η γραμμή και αν θεωρηθεί ότι όλα τα σημεία εισόδου μπορούν να χωριστούν από αυτήν θα υπάρχει

$$B_0 + (B_1 * X_1) + (B_2 * X_2) = 0$$

Όπου οι συντελεστές B_1 και B_2 που καθορίζουν την κλίση της γραμμής και η τομή B_0 υπολογίζονται από τον αλγόριθμο μάθησης, ενώ τα X_1 και X_2 είναι μεταβλητές εισόδου.

Αυτή η γραμμή μπορεί να χρησιμοποιηθεί για ταξινόμηση και με την εισαγωγή νέων μεταβλητών μπορεί να υπολογισθεί αν το νέο σημείο είναι πάνω ή κάτω από την γραμμή.

- Πάνω από την γραμμή, η εξίσωση έχει αποτέλεσμα μεγαλύτερο του 0 και το σημείο ανήκει στην κλάση 0
- Κάτω από την γραμμή, η εξίσωση έχει αποτέλεσμα μικρότερο του 0 και το σημείο ανήκει στην κλάση 1
- Μια τιμή κοντά στην γραμμή έχει αποτέλεσμα κοντά στο 0 και είναι δύσκολο να ταξινομηθεί
- Αν το μέγεθος της τιμής είναι μεγάλο τότε το μοντέλο θα έχει πιο αξιόπιστη πρόβλεψη

Η ιδανικότερη γραμμή για να διαχωρίσει δύο κλάσεις είναι η γραμμή με το μεγαλύτερο περιθώριο (margin) και ονομάζεται υπερεπίπεδο μεγίστου περιθωρίου. Το περιθώριο υπολογίζεται ως η κάθετη απόσταση από την γραμμή προς τα κοντινότερα σημεία. Μόνο αυτά τα σημεία σχετίζονται με τον καθορισμό της γραμμής και την κατασκευή του ταξινομητή. Τα σημεία αποκαλούνται διανύσματα υποστήριξης και υποστηρίζουν ή καθορίζουν το υπερεπίπεδο. Το υπερεπίπεδο μαθαίνεται από τη Μηχανή Διανυσμάτων Υποστήριξης μέσα από το 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.

Ένα επιπλέον σύνολο συντελεστών εισάγονται που δίνουν στο περιθώριο περισσότερο χώρο σε κάθε διάσταση. Οι συντελεστές αυτοί αποκαλούνται κάποια φορά και χαλαρές μεταβλητές (slack variables). Η πολυπλοκότητα του μοντέλου αυξάνεται μιας και υπάρχουν περισσότερες παράμετροι για το μοντέλο που πρέπει να προσασμόσει στα δεδομένα.

Εισάγεται και μια ρυθμιστική παράμετρος που ονομάζεται C η οποία καθορίζει το μέγεθος της κίνησης που επιτρέπεται σε όλες τις διαστάσεις. Οι παράμετροι C καθορίζουν πόση παραβίαση του περιθωρίου επιτρέπεται. Αν $C=0$ σημαίνει ότι δεν επιτρέπεται καμία παραβίαση και άρα γίνεται λόγος για το άκαμπτο Maximal-Margin Classifier. Όσο μεγαλύτερο είναι το C τόσο περισσότερες παραβάσεις του υπερεπιπέδου επιτρέπονται.

Καθώς το C επηρεάζει τον αριθμό των περιπτώσεων που επιτρέπεται να είναι μέσα στο περιθώριο, το C επηρεάζει επίσης και τον αριθμό των διανυσμάτων υποστήριξης που χρησιμοποιούνται από το μοντέλο.

- Όσο μικρότερο είναι το C τόσο πιο ευεπηρεάστος είναι ο αλγόριθμος στα δεδομένα εκπαίδευσης (training data), άρα υπάρχει υψηλότερη διακύμανση (variance) και χαμηλότερη μεροληψία (bias).
- Όσο μεγαλύτερο είναι το C τόσο λιγότερο ευεπηρεάστος είναι ο αλγόριθμος στο training data άρα υπάρχει χαμηλότερη διακύμανση (variance) και υψηλότερη μεροληψία (bias).

Στην περίπτωση που οι κλάσεις δεν είναι γραμμικά διαχωρίσιμες και χρειάζεται να αποφευχθεί η οποιαδήποτε παραβίαση του περιθωρίου είναι δυνατόν να χρησιμοποιηθεί ο κατάλληλος μη-γραμμικός μετασχηματισμός υποθέτοντας ότι τα μετασχηματισμένα πρότυπα είναι πλέον γραμμικά διαχωρίσιμα (Διαμαντάρας, Κ. 2010).

Μια σημαντική σημείωση είναι ότι η γραμμική μηχανή SVM μπορεί να αναδιατυπωθεί χρησιμοποιώντας το εσωτερικό γινόμενο δύο οποιωνδήποτε παρατηρήσεων αντί για τις ίδιες τις παρατηρήσεις. Το εσωτερικό γινόμενο μεταξύ δύο διανυσμάτων είναι το άθροισμα από τον πολλαπλασιασμό του κάθε ζεύγους τιμών εισόδου. Για παράδειγμα, το εσωτερικό γινόμενο των διανυσμάτων $[4, 2]$ και $[8, 5]$ είναι $4 * 8 + 2 * 5$ ή 42.

Αν κάθε μεταβλητή $x^{(i)}$ αντικατασταθεί με την μετασχηματισμένη μορφή $\Phi(x^{(i)})$ τότε τόσο στη συνάρτηση κόστους όσο και στη βέλτιστη διαχωριστική επιφάνεια εμφανίζονται εσωτερικά γινόμενα της μορφής $\Phi(x^{(i)})^T \Phi(y^{(i)})$ και έτσι ορίζεται η συνάρτηση k :

$$k = \Phi(x^{(i)})^T \Phi(y^{(i)})$$

Η συνάρτηση k ονομάζεται συνάρτηση πυρήνα (kernel).

Ο αλγόριθμος SVM εφαρμόζεται χρησιμοποιώντας ένα πυρήνα (kernel). Η μάθηση του υπερεπιπέδου στο γραμμικό SVM γίνεται μετατρέποντας το πρόβλημα χρησιμοποιώντας γραμμική άλγεβρα.

Η εξίσωση για την πρόβλεψη μιας νέας εισόδου χρησιμοποιώντας το εσωτερικό γινόμενο μεταξύ της εισόδου (x) και κάθε διανύσματος υποστήριξης ($x^{(i)}$) υπολογίζεται ως εξής :

$$f(x) = B_0 + \sum_{i=1}^m a_{(i)} (x \cdot x^{(i)})$$

Αυτή είναι μια εξίσωση που περιλαμβάνει τον υπολογισμό των εσωτερικών γινομένων ενός νέου διανύσματος εισόδου (x) με όλα τα διανύσματα υποστήριξης στο σύνολο δεδομένων εκπαίδευσης (training data) . Οι συντελεστές B_0 και $a_{(i)}$ (για κάθε είσοδο) θα πρέπει να εκτιμηθούν από τα δεδομένα εκπαίδευσης μέσω του αλγορίθμου μάθησης.

Το εσωτερικό γινόμενο στην περίπτωση της Μηχανής Διανυσμάτων Υποστήριξης Γραμμικού πυρήνα ονομάζεται πυρήνας και μπορεί να γραφεί ως:

$$K(x, x^{(i)}) = \sum_{i=1}^m xx^{(i)}$$

Ο πυρήνας καθορίζει την ομοιότητα ή την απόσταση μεταξύ των νέων δεδομένων και των διανυσμάτων υποστήριξης . Το εσωτερικό γινόμενο είναι το μέτρο ομοιότητας που χρησιμοποιείται για γραμμική μηχανή SVM ή ένα γραμμικό πυρήνα επειδή η απόσταση είναι γραμμικός συνδυασμός των εισόδων.

Μπορούν να χρησιμοποιηθούν και άλλοι πυρήνες που μετατρέπουν το χώρο εισόδου σε υψηλότερες διαστάσεις, όπως ένας πυρήνας πολυωνύμου ή ένας πυρήνας ακτίνας . Αυτό ονομάζεται **Kernel trick**.

Είναι επιθυμητό να χρησιμοποιηθούν πιο πολύπλοκοι πυρήνες καθώς επιτρέπουν στις γραμμές να διαχωρίσουν τις κλάσεις που είναι καμπυλωτές ή και πιο πολύπλοκες. Αυτό με τη σειρά του μπορεί να οδηγήσει σε ταξινομητές υψηλότερης ακρίβειας.

Σε Μηχανή Διανυσμάτων Υποστήριξης πυρήνα πολυωνύμου, αντί του γινομένου μπορεί να χρησιμοποιηθεί ένας πυρήνας πολυωνύμου για παράδειγμα:

$$K(x, x^{(i)}) = 1 + \sum_{i=1}^m (xx^{(i)})^d$$

Όπου ο βαθμός του πολυωνύμου d πρέπει να καθορίζεται χειροκίνητα για τον αλγόριθμο μάθησης . Όταν $d=1$, είναι το ίδιο με το γραμμικό πυρήνα. Ο πυρήνας πολυωνύμου επιτρέπει καμπυλωτές γραμμές στο χώρο εισόδου.

Μπορεί επίσης να εφαρμοστεί ένας πολυπλοκότερος πυρήνας ακτίνας. Για παράδειγμα:

$$K(\mathbf{x}, \mathbf{x}^{(i)}) = e^{(-\gamma \sum_{i=1}^m (\mathbf{x} - \mathbf{x}^{(i)})^2)}$$

Όπου γ είναι μία παράμετρος που πρέπει να καθορίζεται στον αλγόριθμο μάθησης. Μια καλή αρχική τιμή για το γ είναι 0,1, ενώ το γ είναι συνήθως μεταξύ των τιμών 0 και 1, δηλαδή $0 < \gamma < 1$. Ο πυρήνας ακτίνας είναι πολύ τοπικός και μπορεί να δημιουργήσει πολύπλοκες περιοχές εντός του χώρου των χαρακτηριστικών, όπως πολύγωνα στο χώρο δύο διαστάσεων.

Προετοιμασία δεδομένων για SVM

- Αριθμητικοί Είσοδοι: Το σύστημα SVM υποθέτει ότι οι είσοδοι είναι αριθμητικοί. Αν υπάρχουν είσοδοι σε κατηγορίες θα πρέπει να μετατραπούν σε δυαδικές ψευδομεταβλητές (μία μεταβλητή για κάθε κατηγορία)
- Δυαδική Ταξινόμηση (Binary Classification): Το βασικό SVM προορίζεται για προβλήματα δυαδικής ταξινόμησης. Πάραυτα έχουν αναπτυχθεί επεκτάσεις για προβλήματα παρεμβολής (regression) και ταξινόμησης πολλαπλών τάξεων (multi-class classification).

Στις εφαρμογές μηχανών διανυσμάτων υποστήριξης η συνάρτηση κόστους (cost function) που πρέπει να ελαχιστοποιηθεί είναι συνήθως της μορφής:

$$\min_{\theta} \sum_{i=1}^m y^{(i)} \text{cost}_1(\theta^T \mathbf{x}^{(i)}) + (1 - y^{(i)}) \text{cost}_0(\theta^T \mathbf{x}^{(i)}) + \frac{1}{2} \sum_{j=0}^n \theta_j^2$$

όπου

$$\text{cost}_1 = -\log h_{\theta}(x^{(i)})$$

και

$$\text{cost}_0 = -\log(1 - h_{\theta}(x^{(i)}))$$

Και με συνάρτηση υπόθεσης της πιο πολλές φορές να δίνεται από τον τύπο:

$$h_{\theta}(x) = \begin{cases} 1 & \text{αν } \theta^T \mathbf{x} \geq 0 \\ 0 & \text{στις υπόλοιπες περιπτώσεις} \end{cases}$$

3.5 Νευρωνικά Δίκτυα (Neural Networks)

Μία από τις σημαντικότερες μεθόδους μηχανικής μάθησης είναι τα Νευρωνικά Δίκτυα ή Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks), τα οποία είναι ένα σύνολο μοντέλων δημιουργημένων κατ'αναλογία με τα βιολογικά νευρωνικά δίκτυα και χρησιμοποιούνται στην εκτίμηση συναρτήσεων που βασίζονται σε εισόδους και είναι κατά κανόνα άγνωστες.

Ένας βιολογικός νευρώνας αποτελείται από το σώμα (body), που αποτελεί τον πυρήνα του, τους δενδρίτες (dendrites), μέσω των οποίων λαμβάνει σήματα από γειτονικούς νευρώνες (σημεία εισόδου) και τον άξονα (axon), που είναι η έξοδος του νευρώνα και το μέσο σύνδεσής του με άλλους νευρώνες. Σε κάθε δενδρίτη υπάρχει ένα απειροελάχιστο κενό που ονομάζεται σύναψη (synapse). Οι συνάψεις μέσω χημικών διεργασιών επιταχύνουν ή επιβραδύνουν τη ροή ηλεκτρικών φορτίων προς το σώμα του νευρώνα. Τα ηλεκτρικά σήματα που εισέρχονται στο σώμα των νευρώνων μέσω των δενδριτών συνδυάζονται και αν το αποτέλεσμα ξεπερνάει κάποια τιμή κατωφλίου (threshold) το σήμα διαδίδεται με τη βοήθεια του άξονα προς άλλους νευρώνες (Βλαχάβας, I. et al. 2006).

Σε αντιστοιχία με τον βιολογικό νευρώνα έχει σχεδιαστεί και ο τεχνητός νευρώνας, ο οποίος λαμβάνει ως σήματα εισόδου αντί για ηλεκτρικούς παλμούς κάποιες συνεχείς μεταβλητές. Τα σήματα αυτά μεταβάλλονται από κάποια τιμή βάρους θ (ή w , από το weight) με ρόλο αντίστοιχο της σύναψης στο βιολογικό νευρώνα, δηλαδή θετική τιμή να δρα όπως η επιταχυντική λειτουργία της σύναψης και αρνητική τιμή όπως η επιβραδυντική της λειτουργία. Στον τεχνητό νευρώνα το “σώμα” χωρίζεται στον αθροιστή που προσθέτει τα επηρεασμένα από τα βάρη σήματα εισόδου παράγοντας την τιμή S και τη συνάρτηση ενεργοποίησης (activation function) η οποία διαμορφώνει την τιμή της εξόδου y σύμφωνα με την τιμή S και την τιμή κατωφλίου της συνάρτησης ενεργοποίησης.

Συχνά κατά τη σχεδίαση ενός τεχνητού νευρωνικού δικτύου προστίθεται ένα ακόμα βάρος θ_0 που ονομάζεται πόλωση (bias) ή παράγοντας προδιάθεσης του νευρώνα και επιδρά πάντα σε μία τιμή εισόδου $x_0 = 1$

Η συνάρτηση ενεργοποίησης ενός νευρώνα είναι συνήθως μία μη γραμμική συνάρτηση όπως είναι η βηματική συνάρτηση, η συνάρτηση προσήμου ή η λογιστική συνάρτηση.

Τα τεχνητά νευρωνικά δίκτυα είναι ένα σύνολο οργανωμένων τεχνητών νευρώνων σε δομές παρόμοιες με αυτές του ανθρώπινου εγκεφάλου. Συνήθως η οργάνωση γίνεται σε σειρές από επίπεδα (layers), με πρώτο το επίπεδο εισόδου (input layer), ακολουθούμενο συνήθως από

τα ενδιάμεσα ή κρυφά επίπεδα (hidden layers) και στο τέλος από το επίπεδο εξόδου (output layer). Η αναφορά στη δομή ενός πολυεπίπεδου νευρωνικού δικτύου έχει συνήθως τη μορφή $(p, m_1, m_2, \dots, m_q, n)$ με p αριθμό εισόδων, n αριθμό εξόδων και m_i αριθμό νευρώνων του i κρυφού επιπέδου.

Η εκπαίδευση ενός τεχνητού νευρωνικού δικτύου αποτελεί τη διαδικασία της τροποποίησης των τιμών των βαρών του δικτύου ώστε να παράγεται συγκεκριμένο διάνυσμα εξόδου για συγκεκριμένο διάνυσμα εισόδου. Όταν γίνεται μάθηση με επίβλεψη τα ζευγάρια διανυσμάτων εισόδου και (επιθυμητής) εξόδου δίνονται στο δίκτυο και εκείνο παράγει μία έξοδο με την τρέχουσα κατάσταση των βαρών η οποία διαφέρει από την επιθυμητή. Η διαφορά αυτή ονομάζεται σφάλμα (error) και βάσει της τιμής αυτής και του εκάστοτε αλγορίθμου εκπαίδευσης γίνεται η αναπροσαρμογή των βαρών. Στη μάθηση υπό επίβλεψη οι πιο διαδεδομένοι αλγόριθμοι εκπαίδευσης είναι οι εξής:

- Αλγόριθμοι βασισμένοι στον κανόνα Δέλτα (delta rule learning) όπου η διαφορά μεταξύ πραγματικής και επιθυμητής τιμής εξόδου ελαχιστοποιείται μέσω μιας διαδικασίας ελαχίστων τετραγώνων
- Αλγόριθμος ανάστροφης μετάδοσης λάθους (back propagation) όπου η μεταβολή των βαρών βασίζεται στον υπολογισμό της συνεισφοράς του κάθε βάρους στο συνολικό σφάλμα
- Ανταγωνιστική μάθηση (competitive learning) όπου οι νευρώνες συναγωνίζονται μεταξύ τους και μόνο ο νευρώνας με τη μεγαλύτερη απόκριση σε κάποια είσοδο τροποποιεί τα βάρη του
- Τυχαία μάθηση (random learning) όπου οι μεταβολές στα βάρη εισάγονται τυχαία και ανάλογα με το αν η έξοδος βελτιώνεται ή όχι οι μεταβολές αυτές υιοθετούνται ή απορρίπτονται

Τα δεδομένα εκπαίδευσης χρησιμοποιούνται σε κύκλους εκπαίδευσης (epochs). Σε έναν κύκλο εκπαίδευσης τα δεδομένα εισέρχονται στο δίκτυο ως διανύσματα εκπαίδευσης, αθροίζεται η μεταβολή στην τιμή των βαρών από κάθε διάνυσμα και αναπροσαρμόζονται στο τέλος τα βάρη μέσω της συσσωρευμένης μεταβολής.

Η εκπαίδευση του δικτύου συνήθως τελειώνει όταν εκπληρώνεται κάποια απαιτούμενη συνθήκη μέσω κάποιου κριτηρίου αξιολόγησης και ελέγχου της ποιότητας του δικτύου, όπως

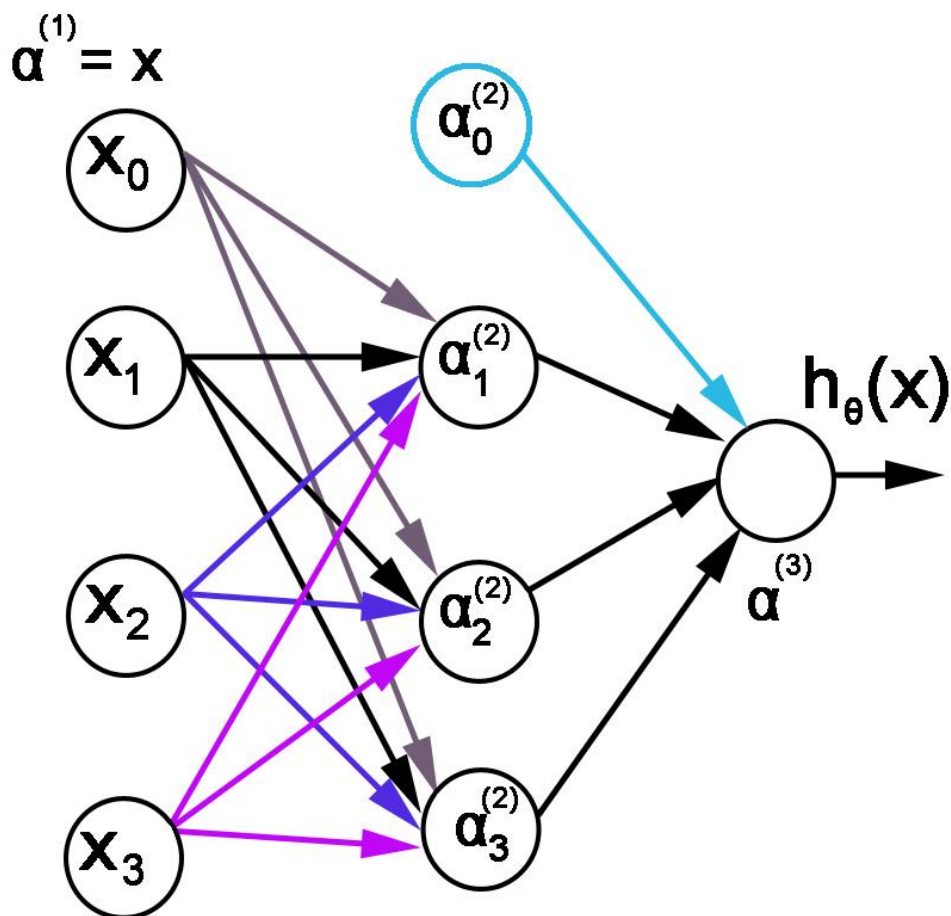
για παράδειγμα χαμηλή τιμή μέσου σφάλματος, ή όταν έχει συμπληρωθεί ένας προεπιλεγμένος αριθμός κύκλων εκπαίδευσης (epochs).

Κατά τη σχεδίαση ενός τεχνητού νευρωνικού δικτύου, πρώτα γίνεται η επιλογή της αρχιτεκτονικής του, με τον αριθμό των εισόδων να ισούται με τη διάσταση των χαρακτηριστικών (features), τον αριθμό των εξόδων να ισούται με τον αριθμό των κλάσεων και με αριθμό κρυφών νευρώνων να είναι πολύ συχνά 1, ενώ στις περιπτώσεις περισσότερων από 1 κρυφών νευρώνων να συνηθίζεται η χρήση ίδιου αριθμού νευρώνων σε κάθε κρυφό επίπεδο.

Με $\alpha_i^{(j)}$ συμβολίζεται η ενεργοποίηση της μονάδας (του νευρώνα δηλαδή) i στο επίπεδο j

Με $\theta^{(j)}$ συμβολίζεται ο πίνακας από τα βάρη των επιπέδων j έως $j+1$

Οι τιμές x_j αντιπροσωπεύουν τα χαρακτηριστικά κάθε παραδείγματος που παρέχεται στο νευρωνικό δίκτυο ως είσοδος, με την τιμή x_0 να αντιστοιχεί στην πόλωση (bias) και να ισούται πάντα με 1.



Εικόνα 3.5.α – Σχηματική Αναπαράσταση Τεχνητού Νευρωνικού Δικτύου

Στο συγκεκριμένο παράδειγμα της Εικόνας 3.5.α αν θεωρηθεί ότι χρησιμοποιείται ως συνάρτηση ενεργοποίησης η λογιστική συνάρτηση δηλαδή η $g(z) = \frac{1}{1+e^{-z}}$ τότε η ενεργοποίηση κάθε μονάδας μπορεί να γραφεί ως:

$$\begin{aligned}\alpha_1^{(2)} &= g(\theta_{10}x_0 + \theta_{11}x_1 + \theta_{12}x_2 + \theta_{13}x_3) \\ \alpha_2^{(2)} &= g(\theta_{20}x_0 + \theta_{21}x_1 + \theta_{22}x_2 + \theta_{23}x_3) \\ \alpha_3^{(2)} &= g(\theta_{30}x_0 + \theta_{31}x_1 + \theta_{32}x_2 + \theta_{33}x_3)\end{aligned}$$

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$z^2 = \theta^{(1)}x \Rightarrow z^2 = \theta^{(1)}\alpha^1$$

$\alpha^2 = g(z^2)$ και κανονικά θα έπρεπε να ανήκει στο \mathbb{R}^3 αλλά προσθέτοντας τον όρο α_0^2 ανήκει πλέον στο \mathbb{R}^4

Αντίστοιχα η $z^3 = \theta^{(2)}\alpha^{(2)}$

Και η υπόθεση είναι $h_\theta(x) = \alpha^{(3)} = g(z^3)$

Για την εκπαίδευση του παραπάνω δικτύου, μία προσέγγιση είναι πρώτα να γίνει τυχαία αρχικοποίηση των τιμών των βαρών, ύστερα να εφαρμοσθεί ο αλγόριθμος forward propagation για να υπολογισθεί η τιμή της $h_\theta(x^{(i)})$ για κάθε παράδειγμα $x^{(i)}$, μετά να υπολογισθεί η αντίστοιχη συνάρτηση κόστους $J(\theta)$ και να εφαρμοσθεί ο αλγόριθμος back propagation μαζί με Gradient Descent για την εύρεση των μερικών παραγώγων και των υπολογισμό των παραμέτρων θ (βάρη).

Κεφάλαιο 4:

Μοντέλα Πρόβλεψης της Πιθανότητας Εμφάνισης Διαβήτη

4.1 Εισαγωγή – Χρήση Dataset

Στο κεφάλαιο αυτό γίνεται μία παρουσίαση μοντέλων πρόβλεψης της πιθανότητας εμφάνισης Σακχαρώδους Διαβήτη με τη χρήση διαφορετικών τεχνικών μέσω του προγράμματος MATLAB. Τα στοιχεία προέρχονται από τη βάση δεδομένων του University of California, Irvine, Machine Learning Repository (UCI Machine Learning Repository)

Στον αντίστοιχο ιστότοπο <http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes> υπάρχει ένα σύνολο δεδομένων εκπαίδευσης με παραδείγματα από 768 άτομα. Οι μετρήσεις έγιναν όλες σε γυναίκες της ινδιάνικης φυλής Pima. Κάθε παράδειγμα περιέχει 8 χαρακτηριστικά που συσχετίζονται με την ύπαρξη ή πιθανότητα εμφάνισης Διαβήτη καθώς και την τιμή της κλάσης εξόδου που χαρακτηρίζεται 0 όταν το άτομο δεν πάσχει από Σακχαρώδη Διαβήτη και 1 όταν πάσχει.

Τα 8 χαρακτηριστικά που παρουσιάζονται είναι ο αριθμός των κυήσεων, η συγκέντρωση γλυκόζης στο πλάσμα του αίματος μετά από εξέταση ανοχής στη γλυκόζη, η διαστολική πίεση του αίματος, η μέτρηση δερματοπτυχής του τρικέφαλου, μέτρηση ινσουλίνης, ο δείκτης μάζας σώματος, η συνάρτηση ιστορικού Διαβήτη και η ηλικία.

Αρχικά έγινε εισαγωγή των δεδομένων στο πρόγραμμα και έπειτα ήταν αναγκαία μία προεπεξεργασία τους. Πρώτα διαχωρίστηκε κάθε χαρακτηριστικό και κάθε πίνακας μετατράπηκε στον ανάστροφό του με τη συνάρτηση `transpose` ώστε να υπάρχουν 8 διανύσματα διαστάσεων 1×768 . Παρατηρήθηκε μετά ότι υπήρχαν μηδενικές τιμές σε σημεία που δεν ήταν βιολογικά δυνατόν (πίεση του αίματος, δείκτης μάζας σώματος, κ.λ.π) οπότε δημιουργήθηκε μία συνάρτηση αντικατάστασης των τιμών 0 με τη μέση τιμή των μη-μηδενικών τιμών κάθε διανύματος χαρακτηριστικών.

Ο κώδικας για τη συνάρτηση αντικατάστασης μηδενικών είναι ο εξής:

```
function [ Output_Vals ] = CountZero( Input_Vals )
%Replaces Zero Values with Average

size=length(Input_Vals);
counterzero=0;
counter=0;
sum=0;
zeropositions=[1,1];
```

```

for i=1:size;
    if (Input_Vals(i)==0)
        counterzero=counterzero+1;
        zeropositions(1,counterzero)=i;
    else
        counter=counter+1;
        sum=sum+Input_Vals(i);
    end
end

Average=sum/counter;

for i=1:counterzero;
    Input_Vals(zeropositions(1,i))=Average;
end
Output_Vals=Input_Vals;
end

```

Ο κώδικας αυτός αφού υπολογίσει τον αριθμό τιμών του διανύσματος μέσω της συνάρτησης *length*, κάνει μία μέτρηση του συνολικού αριθμού των μηδενικών και των μη-μηδενικών στοιχείων καθώς και των θέσεων των μηδενικών στοιχείων στο διάνυσμα. Έπειτα υπολογίζει τη μέση τιμή των μη-μηδενικών στοιχείων και τέλος αντικαθιστά με την τιμή του μέσου όρου κάθε μηδενικό στοιχείο.

Τα τελικά διανύσματα που προέκυψαν συνδυάστηκαν σε ένα πίνακα διαστάσεων 8x768 με την ονομασία *Input*.

Το διάνυσμα εξόδου με την ονομασία *Diabetes* μετατράπηκε μέσω της συνάρτησης *transpose* ώστε να έχει διαστάσεις 1x768.

Έπειτα για τη χρήση του διανύσματος εξόδου ως *Output* σε ορισμένες από τις εφαρμογές, το διάνυσμα *Diabetes* με διαστάσεις 1x768, μέσω μίας μίας συνάρτησης δημιουργίας δεύτερης στήλης στο υπάρχον διάνυσμα μετατράπηκε σε 2x768, με την πρώτη στήλη να είναι η αρχική όπου το 0 δηλώνει τη μη ύπαρξη Σακχαρώδη Διαβήτη και το 1 να δηλώνει την ύπαρξη της νόσου και άρα η πρώτη στήλη να μπορεί να χαρακτηριστεί ως η κλάση “ύπαρξη Διαβήτη” και τη δεύτερη στήλη να έχει τις αντίθετες τιμές για να μπορεί να χαρακτηριστεί ως η κλάση “μη ύπαρξη Διαβήτη”.

Ο κώδικας της συνάρτησης δημιουργίας δεύτερης στήλης ήταν ο εξής:

```

function [ Output_Vals ] = OutputRefinement( Input_Vals )
%OutputRefinement Make the Output suitable for a NN

```



```

%Create a second line in the output matrix with the opposite
value

size=length(Input_Vals);
for i=1:size;
    if Input_Vals(1,i)==0
        Input_Vals(2,i)=1;
    else Input_Vals(2,i)=0;
    end
end

Output_Vals=Input_Vals;

end

```

4.2 Μοντέλο με χρήση Νευρωνικού Δικτύου

Μετά την προεπεξεργασία των δεδομένων υπάρχει στο workspace η μεταβλητή Input με διαστάσεις 8x768 και η μεταβλητή Output με διαστάσεις 2x768.

Η μεταβλητή Input είναι, όπως είχε αναφερθεί πιο πριν, ο συνδυασμός των επι μέρους διανυσμάτων των χαρακτηριστικών ως εξής:

```

Input=[NoPregnant; Glucose; BloodPressure; Triceps; Insulin;
BMI; Pedigree; Age];

```

Έπειτα ο κώδικας προχωράει με τη δημιουργία του νευρωνικού δικτύου

```

hidden=10;
MainNet=patternnet(hidden);

```

Μετά το σύνολο δεδομένων εκπαίδευσης χωρίζεται τυχαία ώστε το 70% του να παραμείνει ως σύνολο δεδομένων εκπαίδευσης (train set) και 15% να γίνει σύνολο διασταυρωμένης επικύρωσης (Cross Validation Set) ενώ το 15% να χρησιμοποιηθεί για τον τελικό έλεγχο ως test set.

```

MainNet.divideFcn = 'dividerand'; % divide randomly
MainNet.divideMode = 'sample'; % static network
MainNet.divideParam.trainRatio = 70/100;
MainNet.divideParam.valRatio = 15/100;
MainNet.divideParam.testRatio = 15/100;

```

Έπειτα γίνεται επιλογή της συνάρτησης εκπαίδευσης του δικτύου, όπου επιλέγεται η συνάρτηση Scaled Conjugate Gradient Backpropagation. καθώς και της συνάρτησης ελέγχου της απόδοσης του δικτύου

```
MainNet.trainFcn = 'trainscg'; % Scaled conjugate gradient  
MainNet.performFcn = 'mse'; % Mean squared error
```

Μετά γίνεται η εκπαίδευση του δικτύου

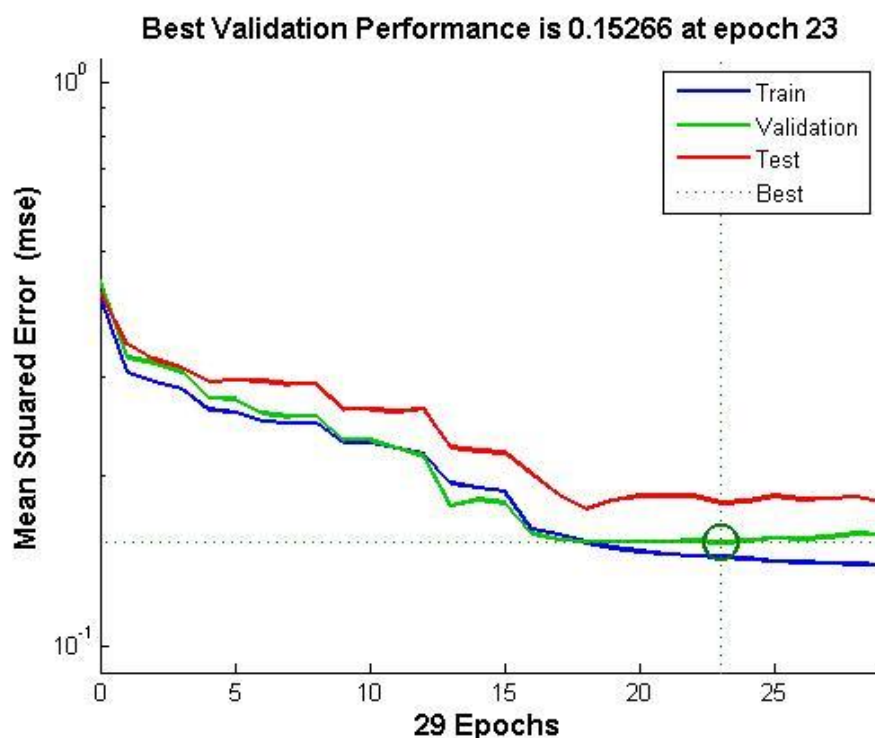
```
[MainNet,tr] = train(MainNet,Input,Output);
```

Και γίνεται έλεγχος της απόδοσης

```
outputs = MainNet(Input);  
errors = gsubtract(Output,outputs);  
performance = perform(MainNet,Output,outputs)  
trainTargets = Output .* tr.trainMask{1};  
valTargets = Output .* tr.valMask{1};  
testTargets = Output .* tr.testMask{1};  
trainPerformance = perform(MainNet,trainTargets,outputs)  
valPerformance = perform(MainNet,valTargets,outputs)  
testPerformance = perform(MainNet,testTargets,outputs)
```

Τέλος είναι δυνατό να σχεδιαστεί το γράφημα της απόδοσης του δικτύου στο Train, Cross Validation και Test Set με την εντολή

```
plotperform(tr)
```



Στο γράφημα φαίνεται πως κατά τη διάρκεια της εκπαίδευσης του νευρωνικού δικτύου μειώνεται το σφάλμα καθώς οι παράμετροι (βάρη) του τεχνητού νευρωνικού δικτύου τροποποιήθηκαν κατάλληλα. Δεν φαίνεται κάποιο overfitting καθώς το σφάλμα στο Cross Validation Set δεν αυξάνεται απότομα αλλά σταματάει η εκπαίδευση πριν να συμβεί κάτι τέτοιο και δεν αυξάνεται το σφάλμα στο Test Set. Ακόμη δεν φαίνεται να υπάρχει underfitting καθώς το σφάλμα τόσο στο Train όσο και στο Cross Validation Set δεν είναι μεγάλο.

Επίσης χρήσιμος είναι ο Πίνακας Συνάφειας Confusion Matrix για την αξιολόγηση του μοντέλου μέσω της εντολής:

`plotconfusion(Output, outputs)`



ή μέσω του γραφικού περιβάλλοντος nntaintool το οποίο εμφανίζει εύκολα τους εξής πίνακες:



Επίσης για τον υπολογισμό του Confusion Matrix μπορεί να χρησιμοποιηθούν οι παρακάτω εντολές:

Πρώτα γίνεται η μετατροπή των πινάκων σε διανύσματα μέσω του εξής κώδικα:

```
Output1=Output(1,:)  
outputs1=outputs(1,:)
```

Για την στρογγυλοποίηση των αποτελεσμάτων των εξόδων του νευρωνικού δικτύου, θέτοντας όλες τις τιμές ≥ 0.5 ως 1 και τις τιμές < 0.5 ως 0 χρησιμοποιείται η παρακάτω εντολή:

```
outputs1=round(outputs1);
```

Υπολογισμός του Confusion Matrix με την ονομασία ConfusionMatrixMain:

```
ConfusionMatrixMain=confusionmat(Output1.',outputs1.','order',  
[1 0])
```

Για το παραπάνω δίκτυο εμφανίζεται ο εξής πίνακας:

ConfusionMatrixMain =

```
156    112  
59     441
```

Από τον Πίνακα Συνάφειας (Confusion Matrix) γίνεται ο υπολογισμός της Ορθότητας (Accuracy), της Ακρίβειας (Precision), της Ανάκλησης (Recall), του F₁score και του Matthews Correlation Coefficient (MCC) ως εξής:

```
AccuracyMain=(ConfusionMatrixMain(1,1)+ConfusionMatrixMain(2,2))/  
768;
```

```
PrecisionMain=ConfusionMatrixMain(2,2)/(ConfusionMatrixMain(2,  
2)+ConfusionMatrixMain(1,2));
```

```
RecallMain=ConfusionMatrixMain(2,2)/(ConfusionMatrixMain(2,2)+  
ConfusionMatrixMain(2,1));
```

```
F1Main=(2*ConfusionMatrixMain(2,2))/(2*ConfusionMatrixMain(2,2)  
+ConfusionMatrixMain(1,2)+ConfusionMatrixMain(2,1));
```

```
MCCMain=(ConfusionMatrixMain(1,1)*ConfusionMatrixMain(2,2)-  
ConfusionMatrixMain(1,2)*ConfusionMatrixMain(2,1))/sqrt((Confu  
sionMatrixMain(2,2)+ConfusionMatrixMain(1,2))*(ConfusionMatrix  
Main(2,2)+ConfusionMatrixMain(2,1))*(ConfusionMatrixMain(1,1)+  
ConfusionMatrixMain(1,2))*(ConfusionMatrixMain(1,1)+ConfusionM  
atrixMain(2,1)));
```

με αποτέλεσμα να προκύπτουν οι εξής τιμές:

```
AccuracyMain =  
0.7773
```

```
PrecisionMain =  
    0.7975
```

```
RecallMain =  
    0.8820
```

```
F1Main =  
    0.8376
```

```
MCCMain =  
    0.4927
```

Το μέσο τετραγωνικό σφάλμα υπολογίζεται με τον εξής τρόπο:

```
MSEMain=immse (Output1.',outputs1.');
```

Και εμφανίζει την τιμή:

```
MSEMain=  
    0.1526
```

Η Καμπύλη ROC υπολογίζεται ως εξής:

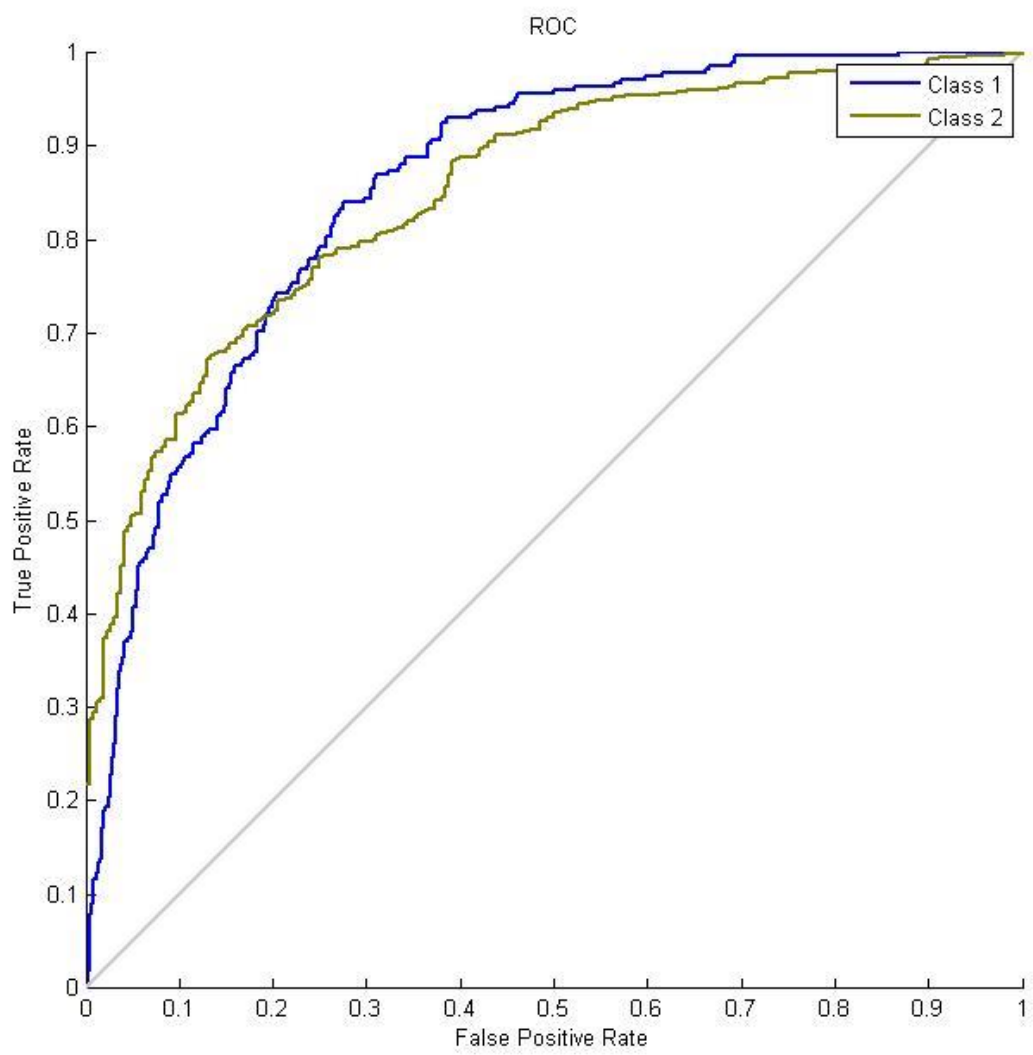
```
plotroc (Output,outputs)
```

Και ο υπολογισμός του AUC γίνεται ως εξής:

```
[Xmain,Ymain,Tmain,AUCMain]=perfcurve (Output1.',outputs1.',1)
```

Το οποίο δίνει για το συγκεκριμένο παράδειγμα αποτέλεσμα:

```
AUCMain =  
    0.7320
```



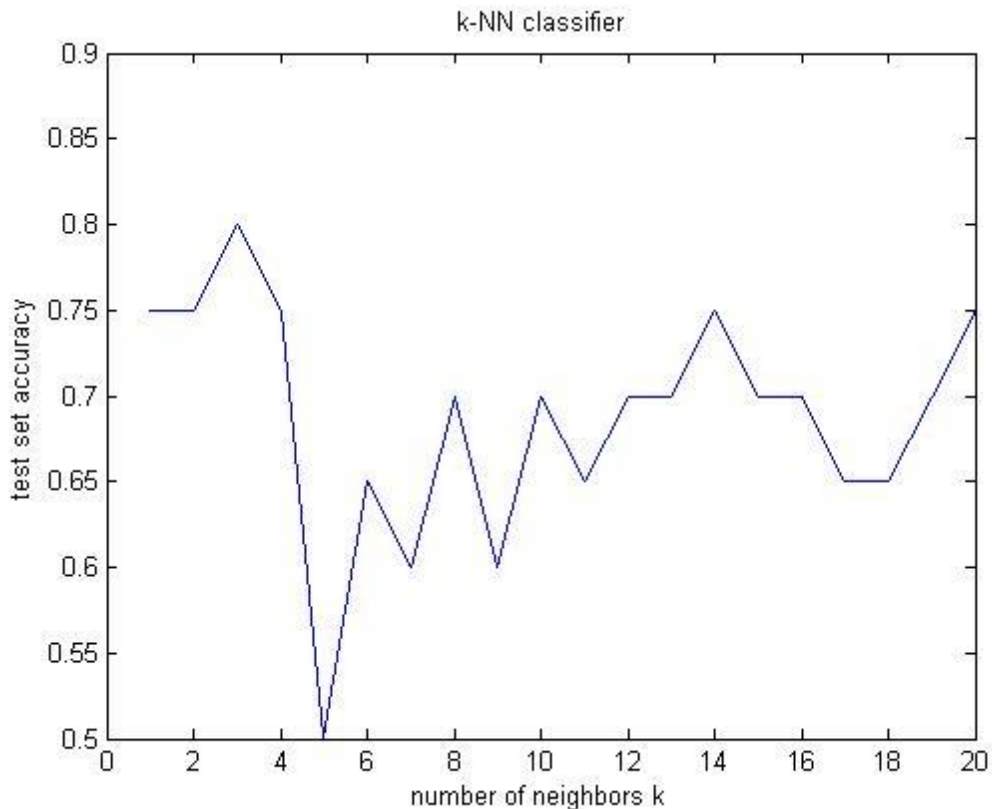
4.2.β Καμπύλη ROC Νευρωνικού Δικτύου

4.3 Μοντέλο με προσέγγιση K-NN

Ο παρακάτω κώδικας στοχεύει στην εύρεση του βέλτιστου K. Στις πρώτες γραμμές εισάγονται και τυχαιοποιούνται τα δεδομένα. Μετά χρησιμοποιούνται οι συναρτήσεις function knn και evaluate που αναλύονται παρακάτω. Εντός της επανάληψης for δοκιμάζονται τιμές του K από το 1 ως το 20 ώστε να βρεθεί η πιο ορθή τιμή.

```
data=[Input.',Diabetes.'];
data=data(51:150,:);
indx = randperm(size(data,1)); % Shuffle the data randomly
data = data(indx,:);
train_labels = data(1:80,end) - 1;
train_data = data(1:80,1:(end-1));
test_labels = data(81:end,end) - 1;
test_data = data(81:end,1:(end-1));
predictions=knn(train_data,train_labels,test_data,1);
acc = evaluate(predictions,test_labels);
fprintf('Accuracy',acc);
accuracies = zeros(20,1);
for k=1:20
    predictions = knn(train_data,train_labels,test_data,k);
    accuracies(k) = evaluate(predictions,test_labels);
end
figure
plot(1:20, accuracies);
title('k-NN classifier');
xlabel('number of neighbors k');
ylabel('test set accuracy');
```

Από εκεί προκύπτει μία γραφική παράσταση που δείχνει το βέλτιστο αριθμό K



4.3.α Γραφική Παράσταση Εύρεσης Βέλτιστου K

Λόγω της γραφικής παράστασης 4.2.α επιλέγεται ο αριθμός K=3.

Συνάρτηση Function knn

Η συνάρτηση αυτή παίρνει σαν είσοδο τα train data και τα train labels καθώς και τα test data και στην συνέχεια γίνεται έλεγχος για κάποια συνηθισμένα σφάλματα. Μετά γίνεται υπολογισμός της Ευκλείδειας απόστασης και γίνεται ταξινόμηση των αποτελεσμάτων σε αύξουσα σειρά. Γίνεται μέτρηση των τιμών των ετικετών με κάθε περίπτωση να αυξάνει την τάξη της και μετά στην έξοδο της συνάρτησης υπάρχει ένα διάνυσμα με τις προβλέψεις του test data.

```
function [predictions] = knn(train_data, train_labels,
test_data, k)

if (nargin ~= 4) % Check number of arguments
    error('Make sure you have entered these 4 arguments;
train, labels, test, k');
end

[N,D] = size(train_data);
N2 = length(train_labels);
[M,D2] = size(test_data);
```

```

predictions = zeros(M,1); % Initialize predictions vector

for m=1:M

    % Euclidean distance between test point m and each
    training point n
    score = zeros(N,1);
    for n=1:N
        score(n) = norm(test_data(m,:)-train_data(n,:),2);
    end

    % Sort distances
    [sorted,indx] = sort(score);

    % For the "k nearest neighbors" take into account their
    labels
    ll = unique(train_labels);
    counts = zeros(size(ll));
    for kk=1:k
        mylabel = train_labels(indx(kk));
        counts(find(ll==mylabel)) = counts(find(ll==mylabel))
+ 1;
    end

    % The prediction is the label value which has the most
    counts
    predictions(m) = ll(find(max(counts)==counts,1));

end

```

Συνάρτηση Function evaluate

Αυτή η συνάρτηση παίρνει το διάνυσμα με τις προβλέψεις σαν είσοδο και συγκρίνει το μήκος του διανύσματος test_labels με τον αριθμό των σωστών προβλέψεων. Έτσι μπορεί να υπολογιστεί η ευστοχία του ταξινομητή.

```

function[accuracy] = evaluate(predictions, test_labels)

N = length(test_labels);
accuracy = length(find(predictions - test_labels == 0)) / N;

```

Μοντέλο KNN

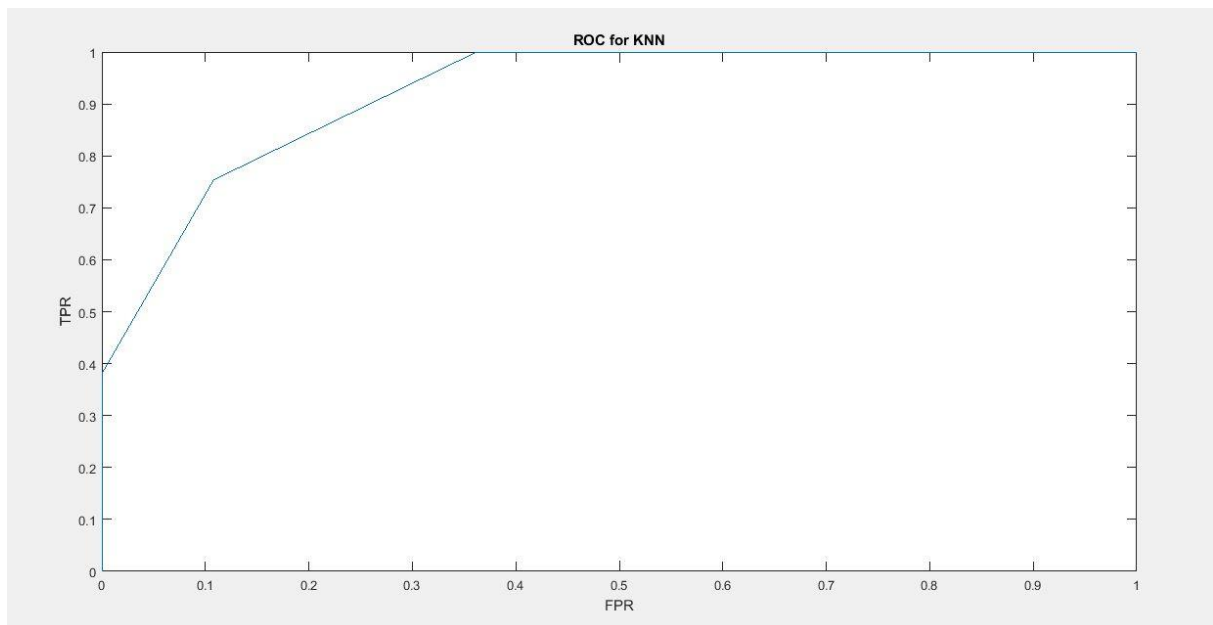
Αφού υπολογιστεί το k απο τη γραφική παράσταση, γίνεται χρήση της συνάρτησης fitcknn όπου εισάγονται είσοδοι και labels και μετά γίνεται ρύθμιση του k=3. Έτσι το μοντέλο έχει την καλύτερη δυνατή απόδοση.

```
knnmodel=fitcknn(Input.',Diabetes.','NumNeighbors',3,'Standard
ize',1);
[knncalc,knnval]=predict(knnmodel,Input.');
```

Έπειτα γίνεται υπολογισμός της απόδοσης του μοντέλου, αρχικά υπολογίζοντας τη γραφική παράσταση ROC και την τιμή AUC:

```
knnval=knnval(:,2);
[Xknn,Yknn,Tknn,AUCknn]=perfcurve(Diabetes.',knnval,1);
plot(Xknn,Yknn)
xlabel('FPR');ylabel('TPR')
title('ROC for KNN')
```

που δίνει το εξής αποτέλεσμα:



$AUC_{knn} =$

0.9220

4.3.β Καμπύλη ROC KNN

Μετά γίνεται υπολογισμός του Πίνακα Συνάφειας (Confusion Matrix), της Ορθότητας (Accuracy), της Ακρίβειας (Precision), της Ανάκλησης (Recall), του F_1 score και του Matthews Correlation Coefficient (MCC) ως εξής

```
[Cknn]=confusionmat(Diabetes.',knncalc,'order',[0 1]);
```

```
AccuracyKnn=(Cknn(1,1)+Cknn(2,2))/768
```

$$PrecisionKnn = Cknn(2,2) / (Cknn(2,2) + Cknn(1,2))$$

$$RecallKnn = Cknn(2,2) / (Cknn(2,2) + Cknn(2,1))$$

$$F1Knn = (2 * Cknn(2,2)) / (2 * Cknn(2,2) + Cknn(1,2) + Cknn(2,1))$$

$$MCCKnn = (Cknn(1,1) * Cknn(2,2) - Cknn(1,2) * Cknn(2,1)) / \sqrt{(Cknn(2,2) + Cknn(1,2)) * (Cknn(2,2) + Cknn(2,1)) * (Cknn(1,1) + Cknn(1,2)) * (Cknn(1,1) + Cknn(2,1))}$$

Το οποίο εμφανίζει τις εξής τιμές:

Cknn =

202 66

54 446

AccuracyKnn =

0.8438

PrecisionKnn =

0.8711

RecallKnn =

0.8920

F1Knn =

0.8814

MCCKnn =

0.6529

Ο υπολογισμός του μέσου τετραγωνικού σφάλματος γίνεται ως εξής:

knnmse = *immse*(*Diabetes.*' , *knncalc*);

Το οποίο δίνει:

knnmse=0.1598

4.4 Μοντέλο με τη μέθοδο Logistic Regression

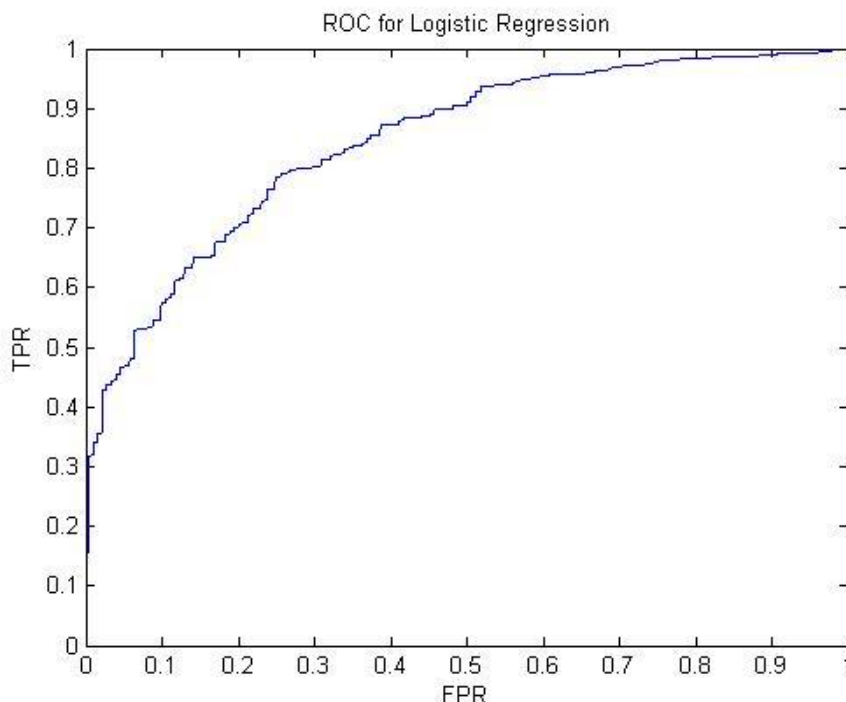
Εφαρμόζεται η συνάρτηση `mnrfit` για multinomial logistic regression. Τα δεδομένα που εισάγονται είναι ο πίνακας που δημιουργήθηκε για τις εισόδους (`Input`) αλλά υπολογίζεται ο ανάστροφος πίνακας `transpose`. Τα labels που εισάγονται εδώ είναι στο διάνυσμα `Diabetes` και προστίθεται 1 γιατί το function δέχεται ακέραιες τιμές μεγαλύτερες του 0.

Logistic Regression

```
logfit=mnrfit(Input.',Diabetes.'+1);
```

Για την εύρεση της καμπύλης ROC (receiver operating characteristic) εφαρμόζεται ο παρακάτω κώδικας:

```
%Plot ROC  
logval=mnrval(logfit,Input. ');  
logval=logval(:,1);  
[Xlog,Ylog,Tlog,AUClog]=perfcurve(Diabetes.'+1,logval,1);  
plot(X,Y)  
xlabel('FPR');ylabel('TPR')  
title('ROC for Logistic Regression')
```



4.4.α Καμπύλη ROC Logistic Regression

`AUClog =`

`0.8446`

Έπειτα υπολογίζεται ο Πίνακας Συνάφειας (Confusion Matrix) και η Ορθότητα (Accuracy), η Ακρίβεια (Precision), η Ανάκληση (Recall), το F_1 score και ο συντελεστής Matthews Correlation Coefficient (MCC) ως εξής:

```
logground=round(logval1);
[Clog]=confusionmat(Diabetes.',logground,'order',[0 1]);
AccuracyLog=(Clog(1,1)+Clog(2,2))/768
PrecisionLog=Clog(2,2)/(Clog(2,2)+Clog(1,2))
RecallLog=Clog(2,2)/(Clog(2,2)+Clog(2,1))
F1Log=(2*Clog(2,2))/(2*Clog(2,2)+Clog(1,2)+Clog(2,1))
MCCLog=(Clog(1,1)*Clog(2,2)-
Clog(1,2)*Clog(2,1))/sqrt((Clog(2,2)+Clog(1,2))*(Clog(2,2)+Clog(2,1))*
(Clog(1,1)+Clog(1,2))*(Clog(1,1)+Clog(2,1)))
```

Που εμφανίζει τις εξής τιμές:

```
AccuracyLog =
    0.7773
```

```
PrecisionLog =
    0.7277
```

```
RecallLog =
    0.5784
```

```
F1Log =
    0.6445
```

```
MCCLog =
    0.4923
```

Και τέλος το μέσο τετραγωνικό σφάλμα δίνεται από τον τύπο:

```
logmse = immse(Diabetes.',logground);
```

που δίνει την τιμή:

```
logmse = 0.2227
```

4.5 Μοντέλο με SVM

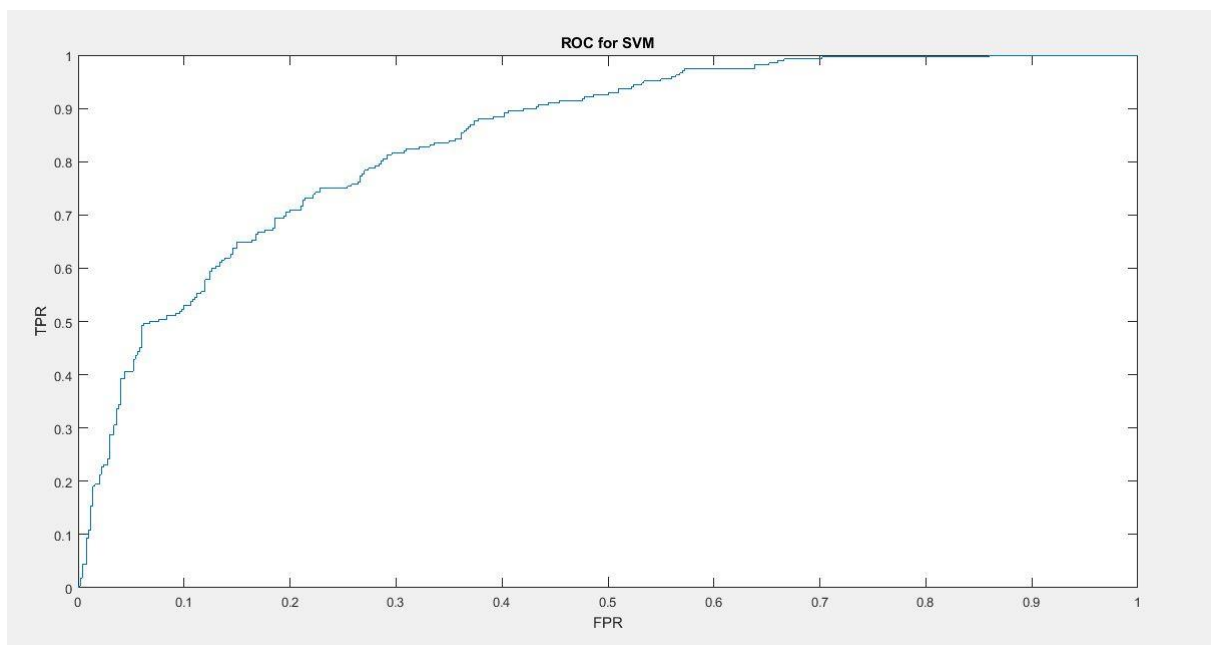
Για το μοντέλο της μηχανής SVM δεν χρειάστηκε να προετοιμαστούν τα δεδομένα αφού αρκεί να χρησιμοποιηθεί η έτοιμη συνάρτηση `fitcsvm` με τους ανάστροφους πίνακες (transpose) των εισόδων και των εξόδων από το dataset.

SVM

```
SVMfit=fitcsvm(Input.',Diabetes.');
```

Η καμπύλη ROC υπολογίζεται ως εξής:

```
[SVMcalc,SVMval]=predict(SVMfit,Input. ');  
SVMval=SVMval(:,2);  
[Xsvm,Ysvm,Tsvm,AUCsvm]=perfcurve(Diabetes.',SVMval,1);  
plot(Xsvm,Ysvm)  
xlabel('FPR');ylabel('TPR')  
title('ROC for SVM')
```



4.5.α Καμπύλη ROC SVM

$AUC_{svm} =$

0.8435

Μετά γίνεται ο υπολογισμός του Πίνακα Συνάφειας (Confusion Matrix), της Ορθότητας (Accuracy), της Ακρίβειας (Precision), της Ανάκλησης (Recall), του F_1 score και του Matthews Correlation Coefficient (MCC) ως εξής:

```
[CSVM]=confusionmat(Diabetes.',SVMcalc,'order',[0 1]);
AccuracySVM=(CSVM(1,1)+CSVM(2,2))/768
PrecisionSVM=CSVM(2,2)/(CSVM(2,2)+CSVM(1,2))
RecallSVM=CSVM(2,2)/(CSVM(2,2)+CSVM(2,1))
F1SVM=(2*CSVM(2,2))/(2*CSVM(2,2)+CSVM(1,2)+CSVM(2,1))
MCCSVM=(CSVM(1,1)*CSVM(2,2)-
CSVM(1,2)*CSVM(2,1))/sqrt((CSVM(2,2)+CSVM(1,2))*(CSVM(2,2)+CSVM(2,1))+CSVM(2,1)*(CSVM(1,1)+CSVM(1,2))*(CSVM(1,1)+CSVM(2,1)))
```

που εμφανίζει τις εξής τιμές:

AccuracySVM =

0.7682

PrecisionSVM =

0.7184

RecallSVM =

0.5522

F1SVM =

0.6245

MCCSVM =

0.4693

Και τέλος ο υπολογισμός του μέσου τετραγωνικού σφάλματος:

```
SVMmse =immse (Diabetes.', SVMcalc)
```

Που δίνει την τιμή:

```
SVMmse = 0.2318
```

Κεφάλαιο 5:

Σύγκριση Διαφορετικών Μοντέλων Πρόβλεψης

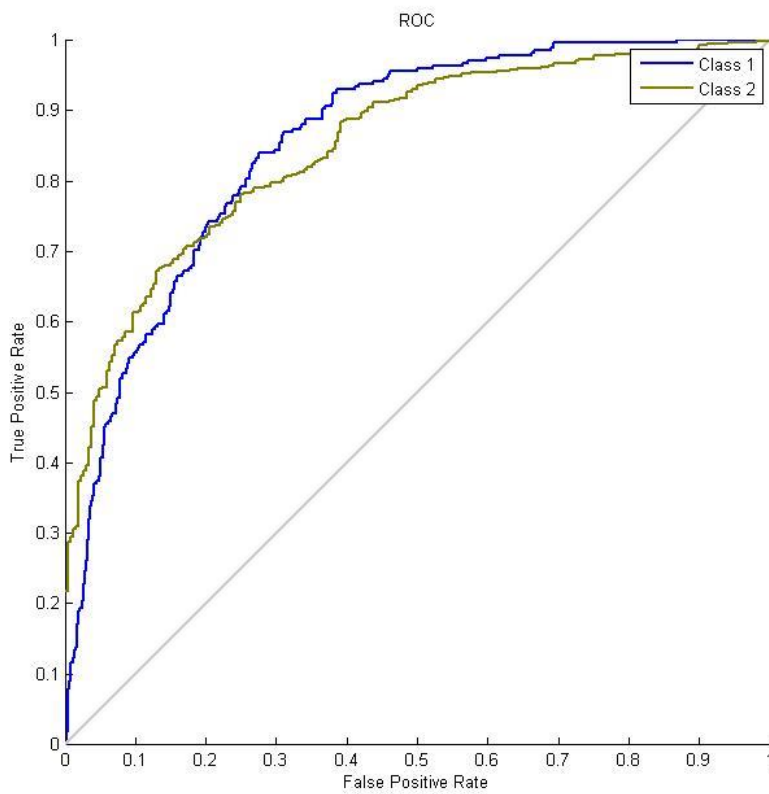
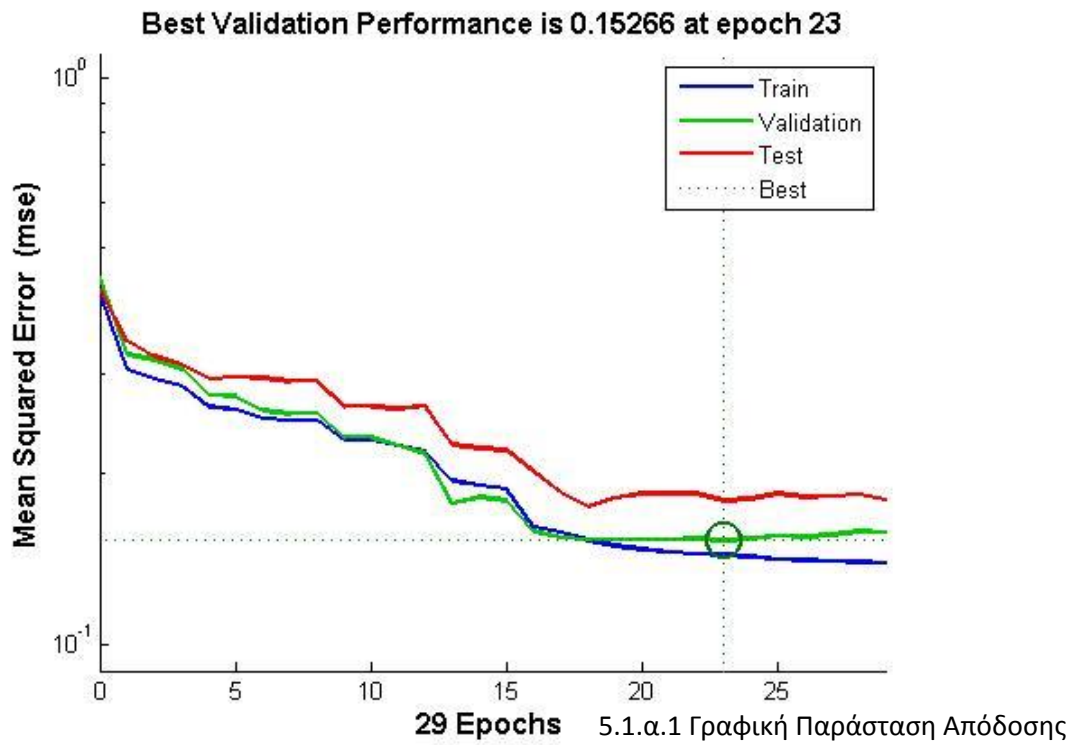
5.1 Σύγκριση της Απόδοσης Διαφορετικών Συναρτήσεων Εκπαίδευσης Νευρωνικών Δικτύων

Στο κεφάλαιο 4.2 παρουσιάστηκε ένα μοντέλο πρόβλεψης της πιθανότητας εμφάνισης Σακχαρώδους Διαβήτη μέσω Νευρωνικού Δικτύου που εκπαιδεύεται με τη Συνάρτηση Scaled Conjugate Gradient Backpropagation. Παρακάτω παρατίθενται μοντέλα νευρωνικών δικτύων με διαφορετικές συναρτήσεις εκπαίδευσης μαζί με τη γραφική παράσταση της απόδοσης ορισμένης ως το μέσο τετραγωνικό σφάλμα ανά κύκλο εκπαίδευσης (Mean squared error – mse vs. epoch), την Καμπύλη ROC (Καμπύλη Λειτουργικών Χαρακτηριστικών), τον Πίνακα Συνάφειας (ConfusionMatrix), την Ορθότητα (Accuracy), την Ακρίβεια (Precision), την Ανάκληση (Recall), το F_1 score, το συντελεστή Matthews Correlation Coefficient (MCC) και το εμβαδόν κάτω από την καμπύλη ROC (AUC - Area Under the Curve) για κάθε ένα από εκείνα.

Οι συναρτήσεις εκπαίδευσης που θα χρησιμοποιηθούν είναι οι:

- Scaled Conjugate Gradient Backpropagation
- Conjugate gradient backpropagation with Powell-Beale restarts
- Conjugate gradient backpropagation with Fletcher-Reeves updates
- Conjugate gradient backpropagation with Polak-Ribière updates
- Gradient descent with adaptive learning rate backpropagation
- Gradient descent with momentum backpropagation
- Gradient descent with momentum and adaptive learning rate backpropagation
- BFGS quasi-Newton backpropagation
- Levenberg-Marquardt backpropagation
- One-step secant backpropagation
- Resilient backpropagation

- Scaled Conjugate Gradient Backpropagation (εντολή όπως παρουσιάστηκε στο κεφάλαιο 4.2)



5.1.α.2 Καμπύλη ROC

ConfusionMatrix

156 112

59 441

Accuracy=0.7773

Precision=0.7975

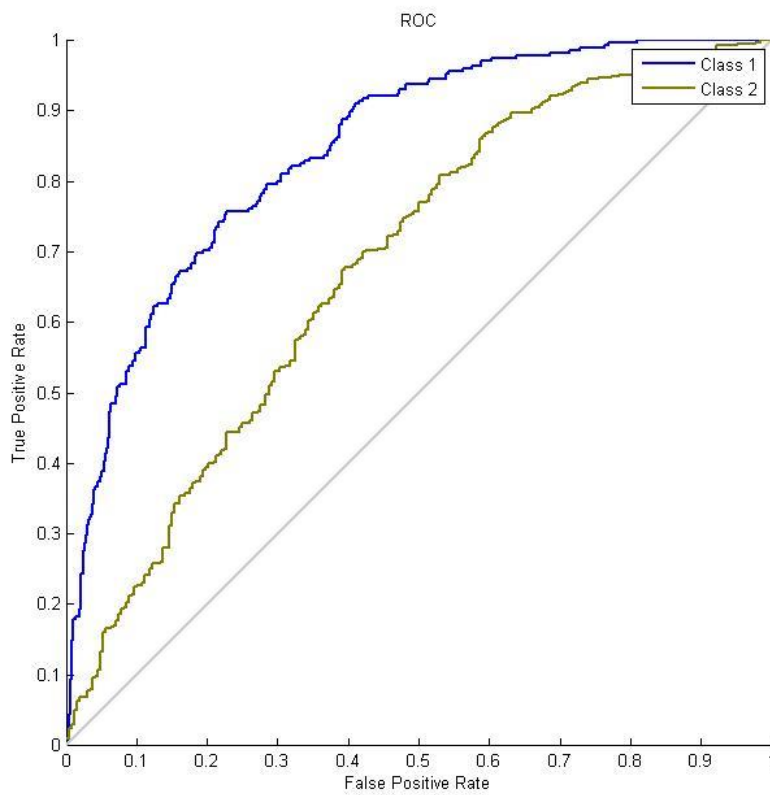
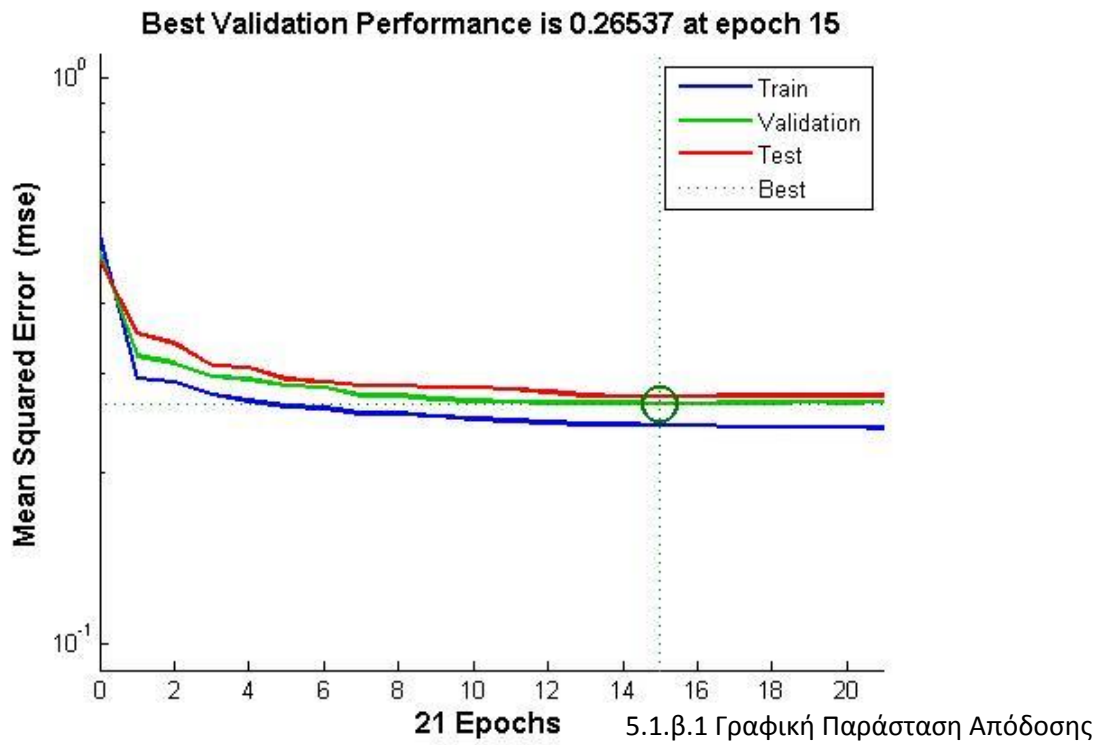
Recall =0.8820

F1=0.8376

MCCM=0.4927

AUC= 0.7320

- Conjugate gradient backpropagation with Powell-Beale restarts (εντολή Traincgb)



5.1.β.2 Καμπύλη ROC

ConfusionMatrix =

166 102

62 438

Accuracy = 0.7865

Precision = 0.8111

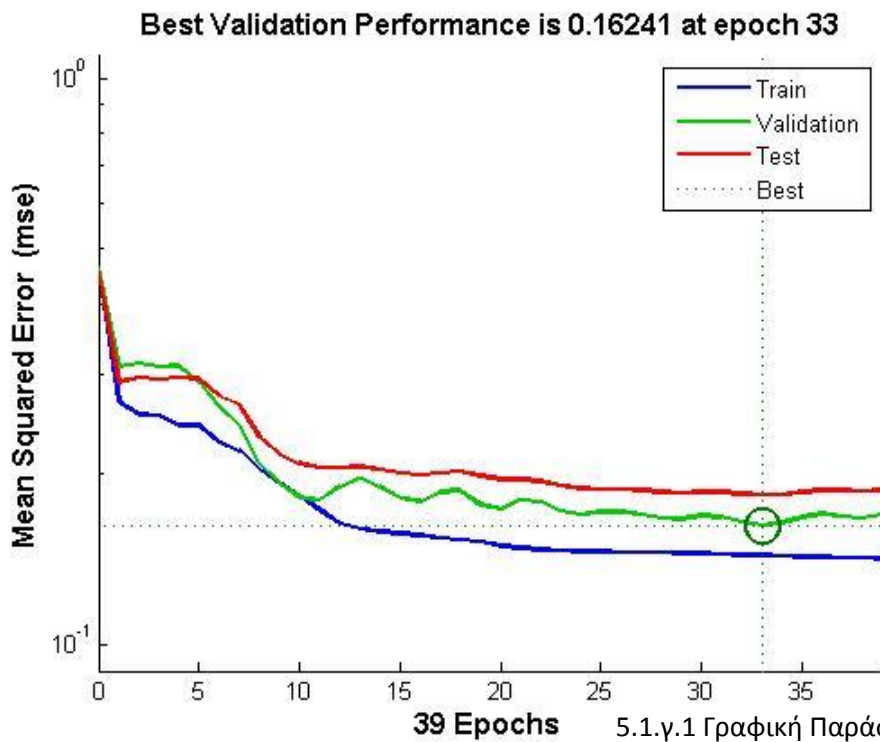
Recall = 0.8760

F1 = 0.8423

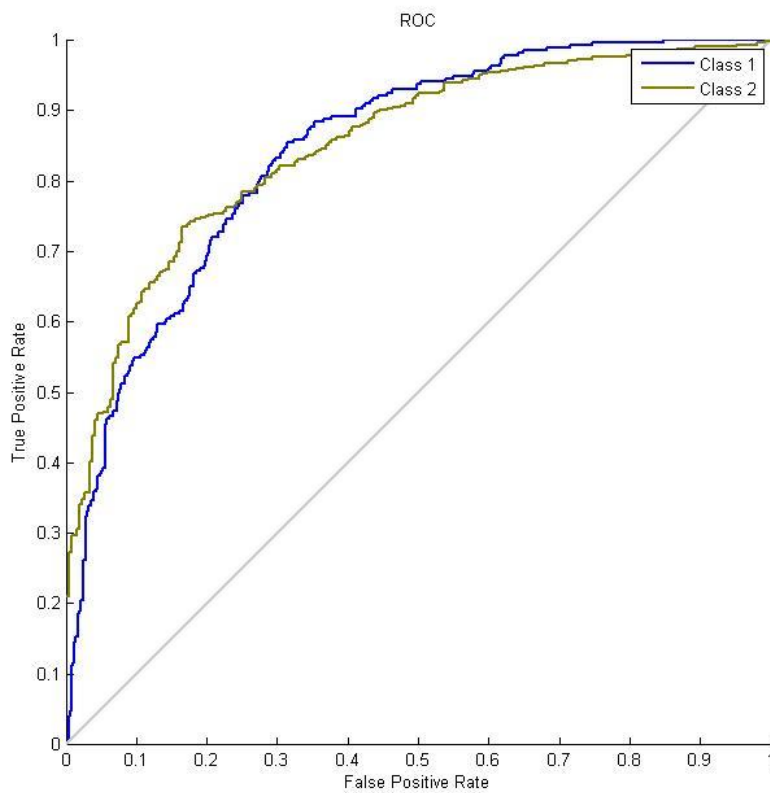
MCC = 0.5168

AUC = 0.8446

- Conjugate gradient backpropagation with Fletcher-Reeves updates (εντολή Traincgf)



5.1.γ.1 Γραφική Παράσταση Απόδοσης



5.1.γ.2 Καμπύλη ROC

ConfusionMatrix =

155 113

63 437

Accuracy = 0.7708

Precision = 0.7945

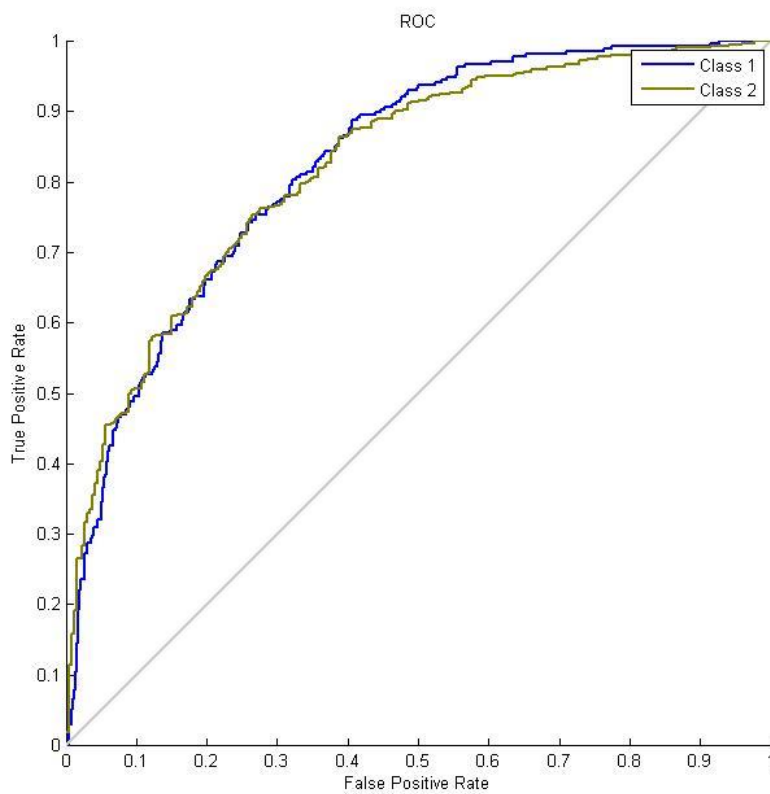
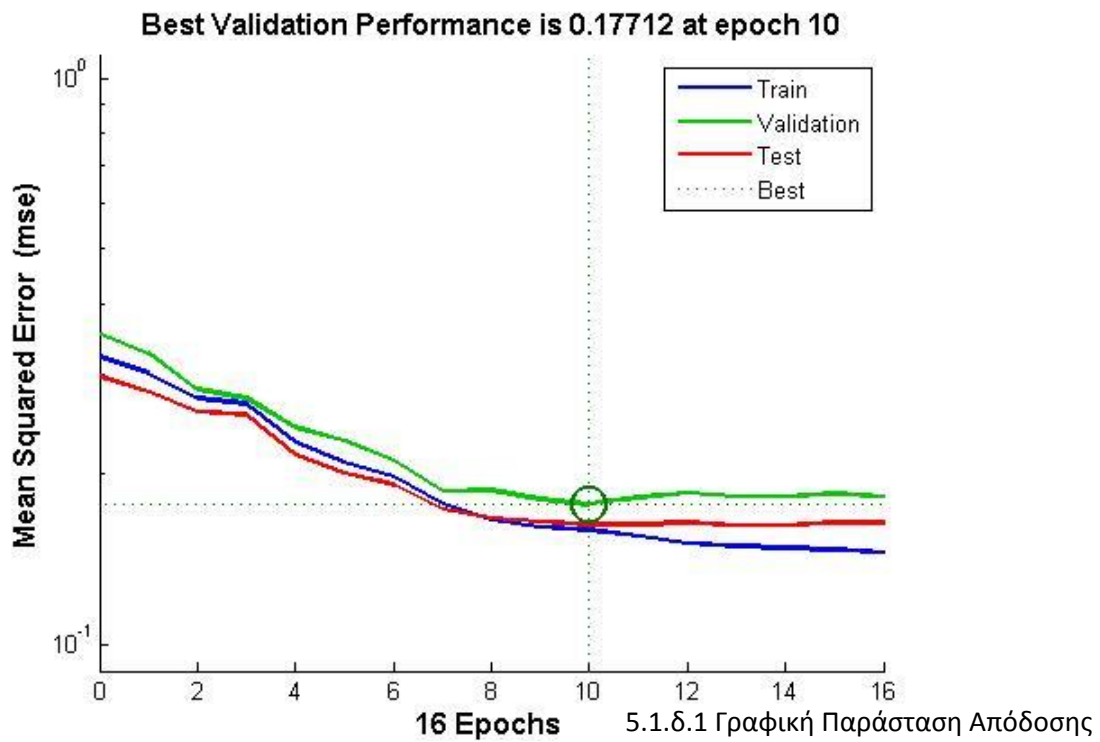
Recall = 0.8740

F1 = 0.8324

MCC = 0.4782

AUC = 0.7262

- Conjugate gradient backpropagation with Polak-Ribière updates (εντολή Traincgp)



ConfusionMatrix =

169 99

88 412

Accuracy = 0.7565

Precision = 0.8063

Recall = 0.8240

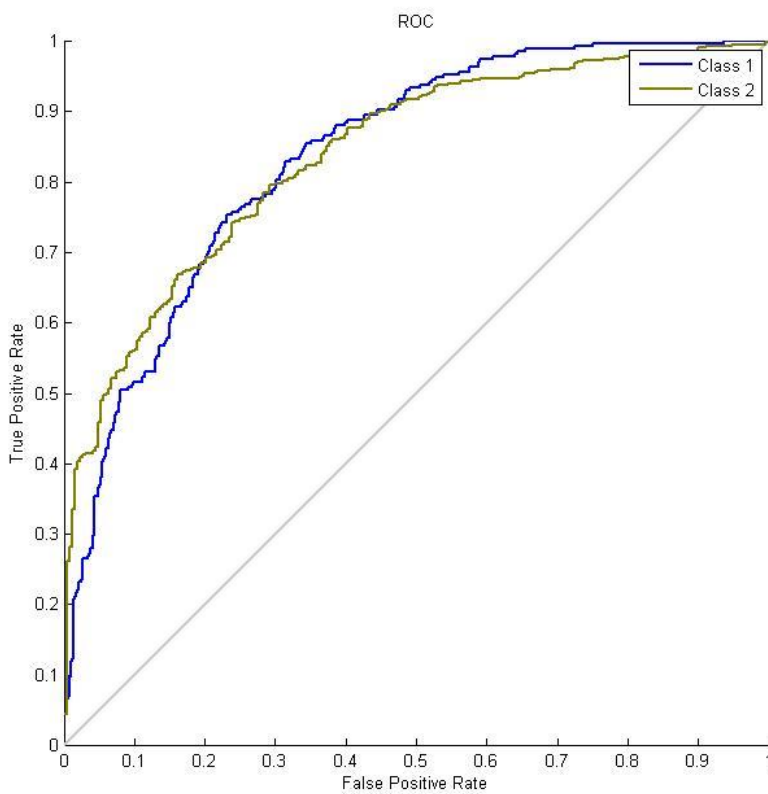
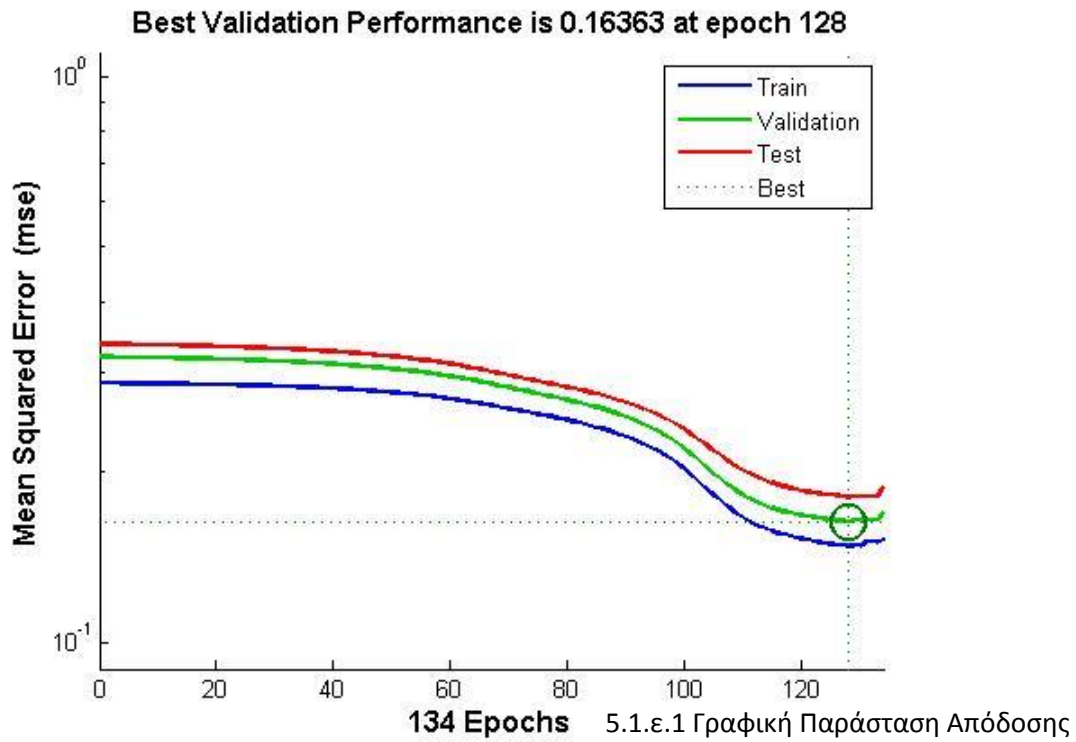
F1 = 0.8150

MCC = 0.4592

AUC = 0.7273

5.1.δ.2 Καμπύλη ROC

- Gradient descent with adaptive learning rate backpropagation (εντολή Traingda)



5.1.ε.2 Καμπύλη ROC

ConfusionMatrix =

155 113

73 427

Accuracy = 0.7578

Precision = 0.7907

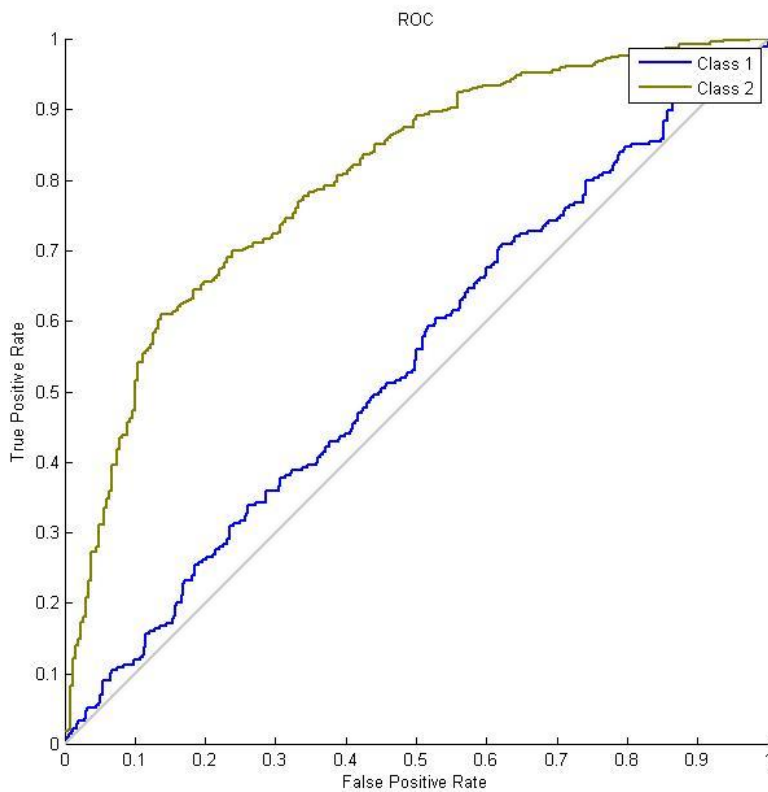
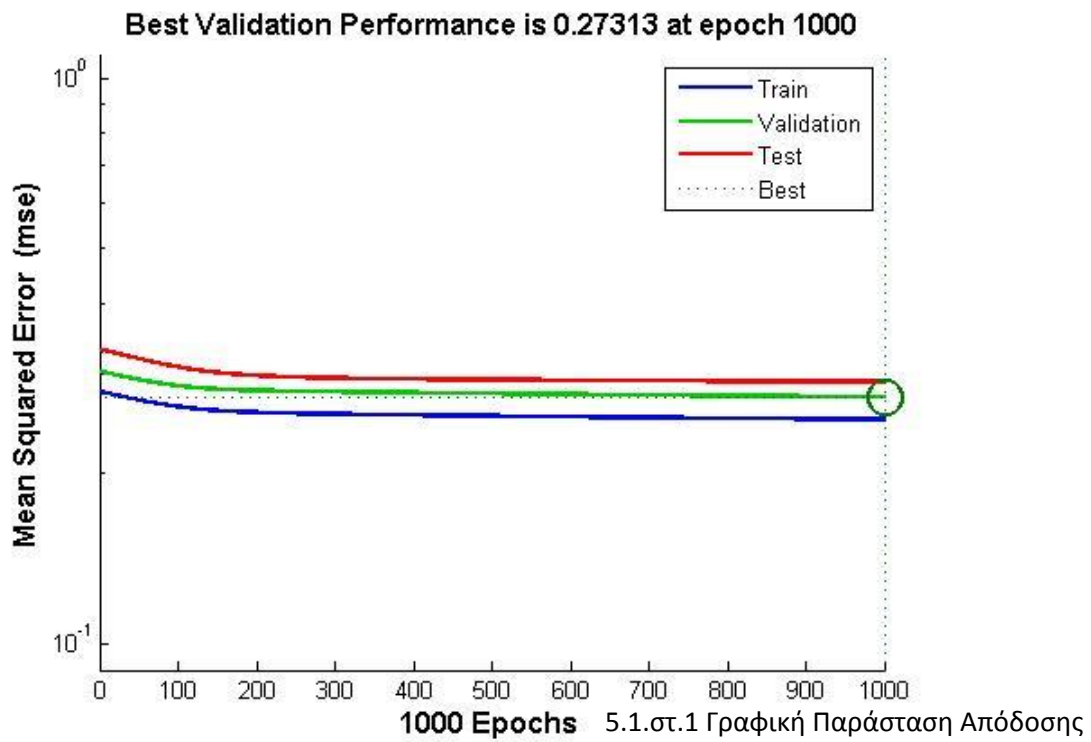
Recall = 0.8540

F1 = 0.8212

MCC = 0.4511

AUC = 0.7162

- Gradient descent with momentum backpropagation (εντολή Trainingdm)



5.1.σ.2 Καμπύλη ROC

ConfusionMatrix =

```
0 268
0 500
```

Accuracy = 0.6510

Precision = 0.6510

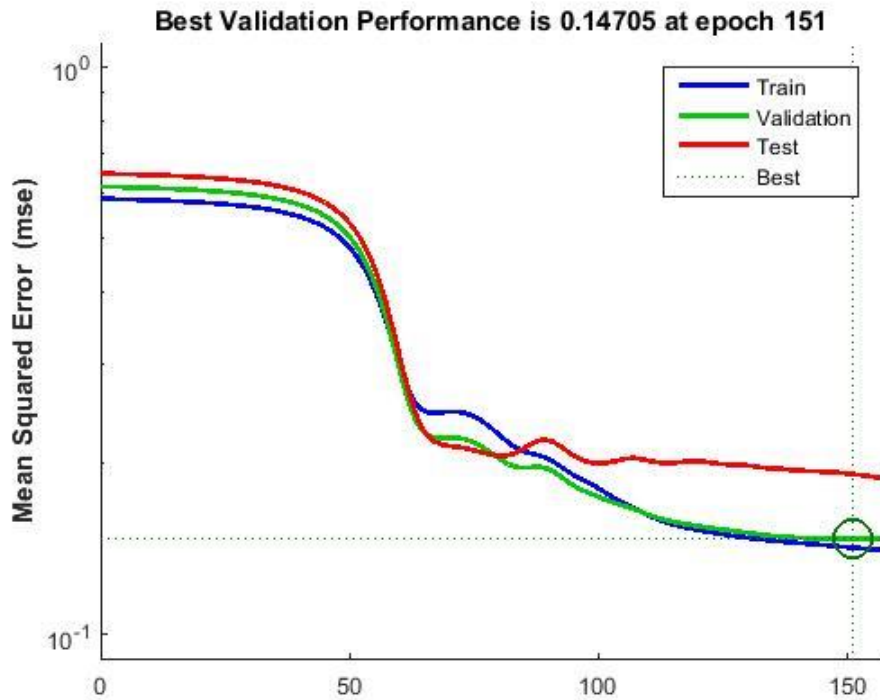
Recall = 1

F1 = 0.7886

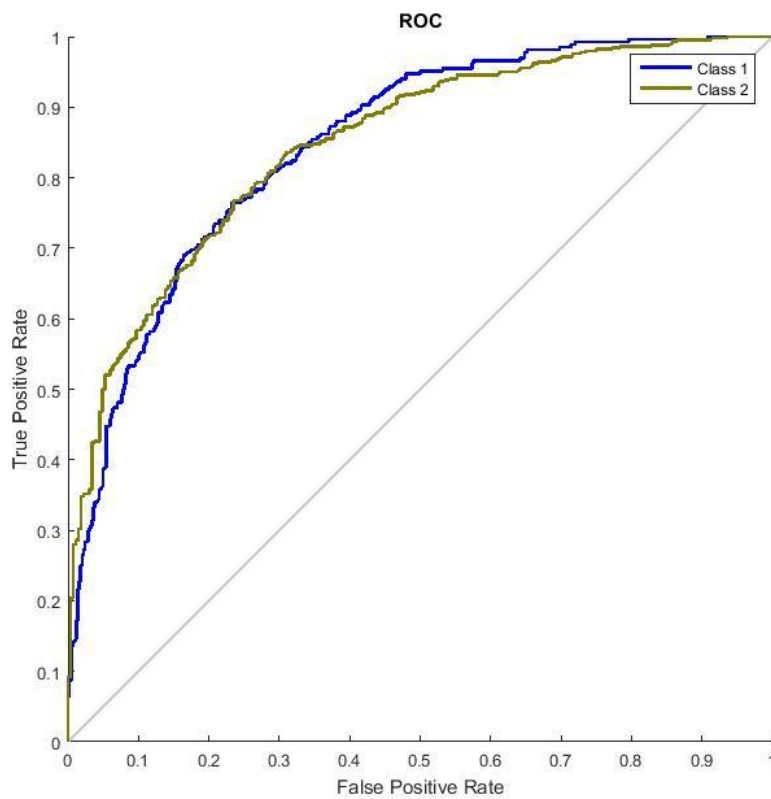
MCC = NaN

AUC = 0.5000

- Gradient descent with momentum and adaptive learning rate backpropagation (εντολή Traingdx)



157 Epochs 5.1.ζ.1 Γραφική Παράσταση Απόδοσης



5.1.ζ.2 Καμπύλη ROC

ConfusionMatrix =

170 98

72 428

Accuracy = 0.7786

Precision = 0.8137

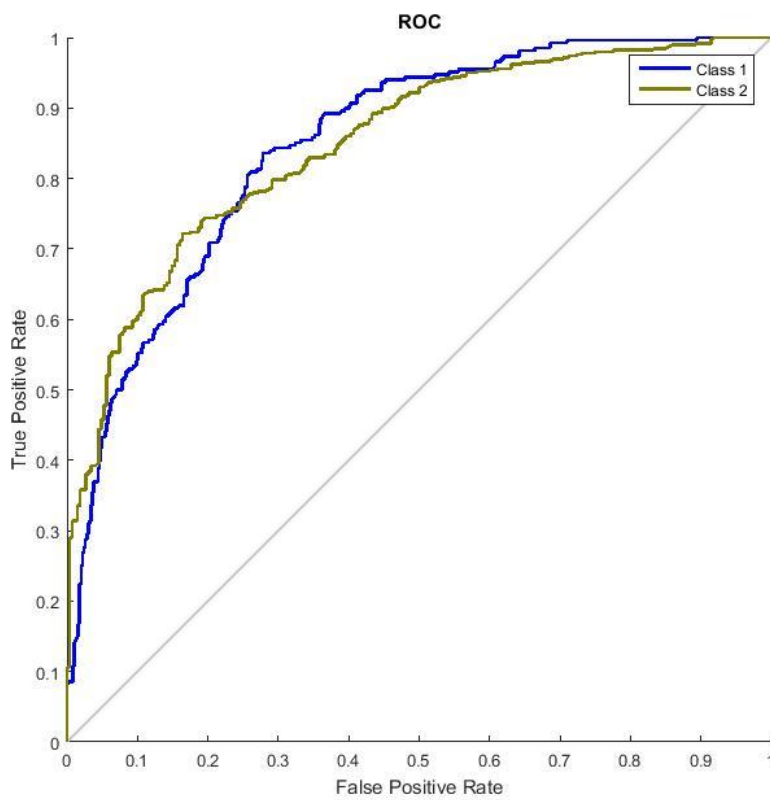
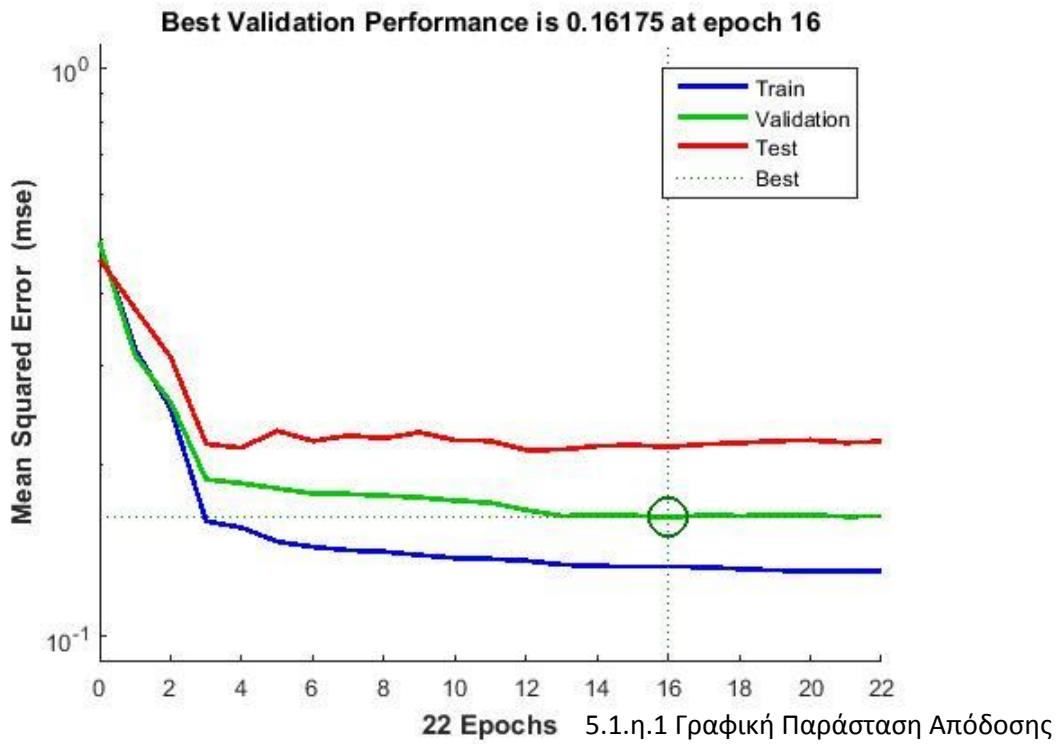
Recall = 0.8560

F1 = 0.8343

MCC = 0.5031

AUC = 0.7452

- BFGS quasi-Newton backpropagation (εντολή Trainbfg)



5.1.η.2 Καμπύλη ROC

ConfusionMatrix =

157	111
63	437

Accuracy = 0.7734

Precision = 0.7974

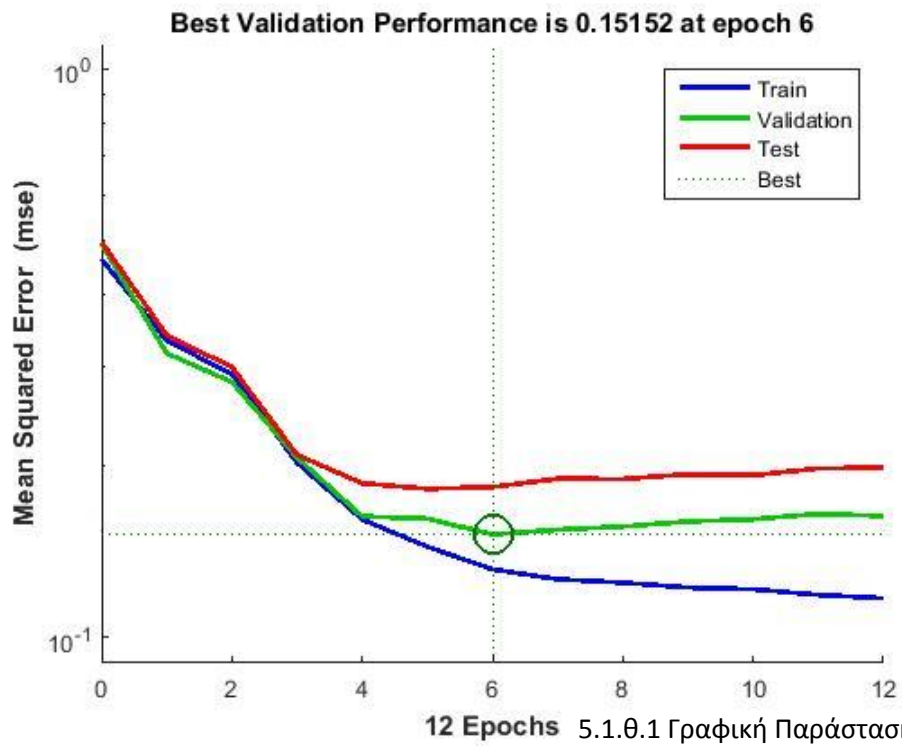
Recall = 0.8740

F1 = 0.8340

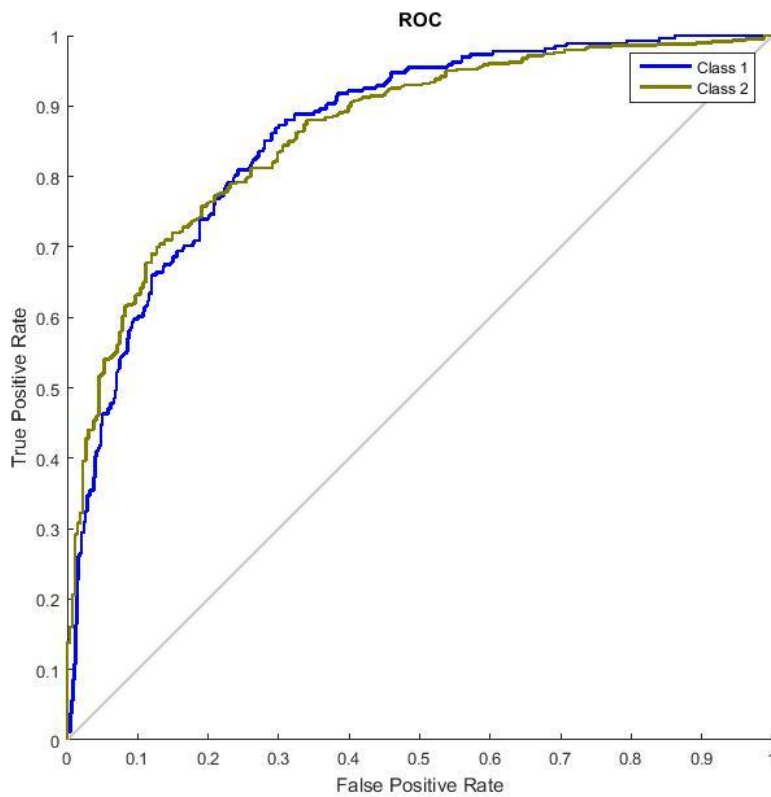
MCC = 0.4848

AUC = 0.7299

- Levenberg-Marquardt backpropagation (εντολή Trainlm)



5.1.θ.1 Γραφική Παράσταση Απόδοσης



5.1.θ.2 Καμπύλη ROC

ConfusionMatrix =

160 108

49 451

Accuracy = 0.7956

Precision = 0.8068

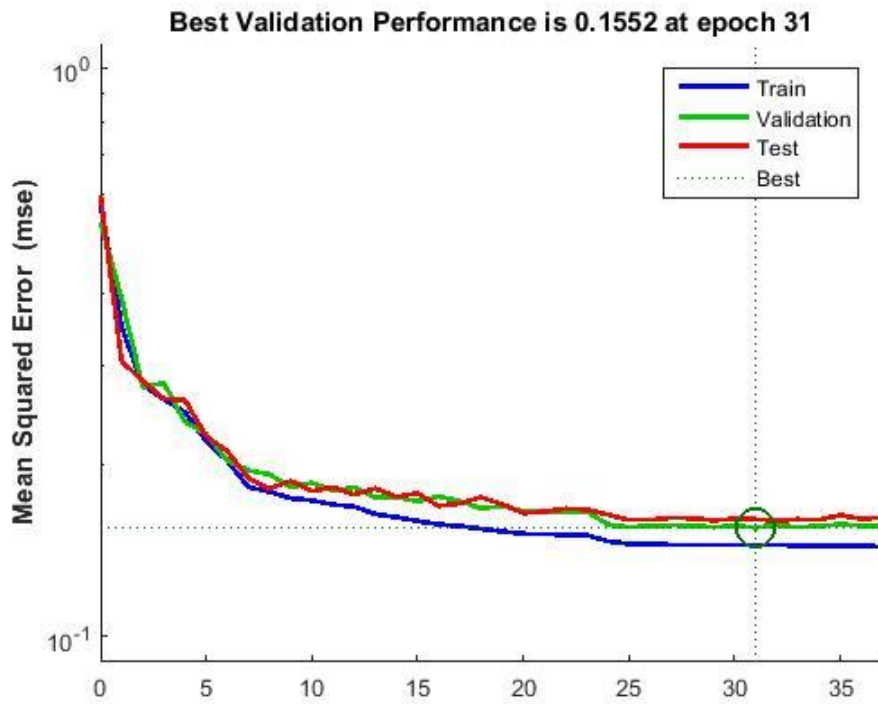
Recall = 0.9020

F1 = 0.8517

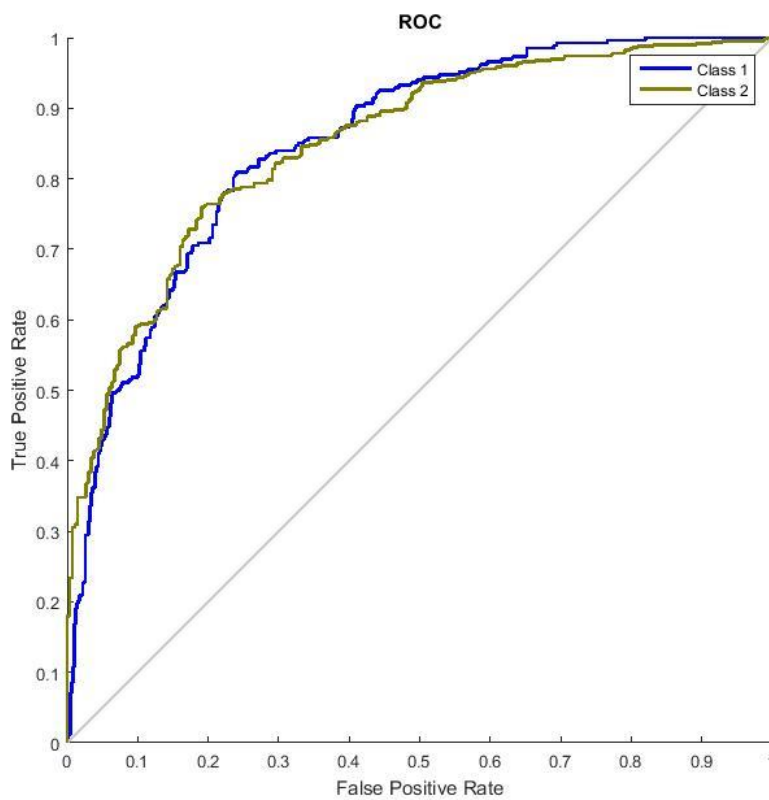
MCC = 0.5344

AUC = 0.7495

- One-step secant backpropagation (εντολή Trainoss)



37 Epochs 5.1.1.1 Γραφική Παράσταση Απόδοσης



5.1.1.2 Καμπύλη ROC

ConfusionMatrix =

162 106

64 436

Accuracy = 0.7786

Precision = 0.8044

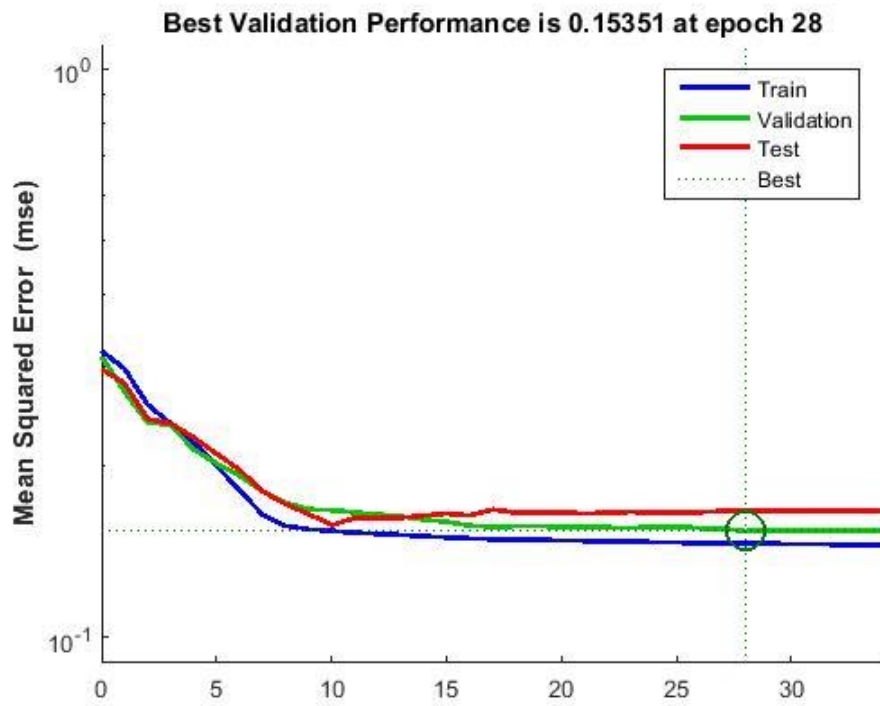
Recall = 0.8720

F1 = 0.8369

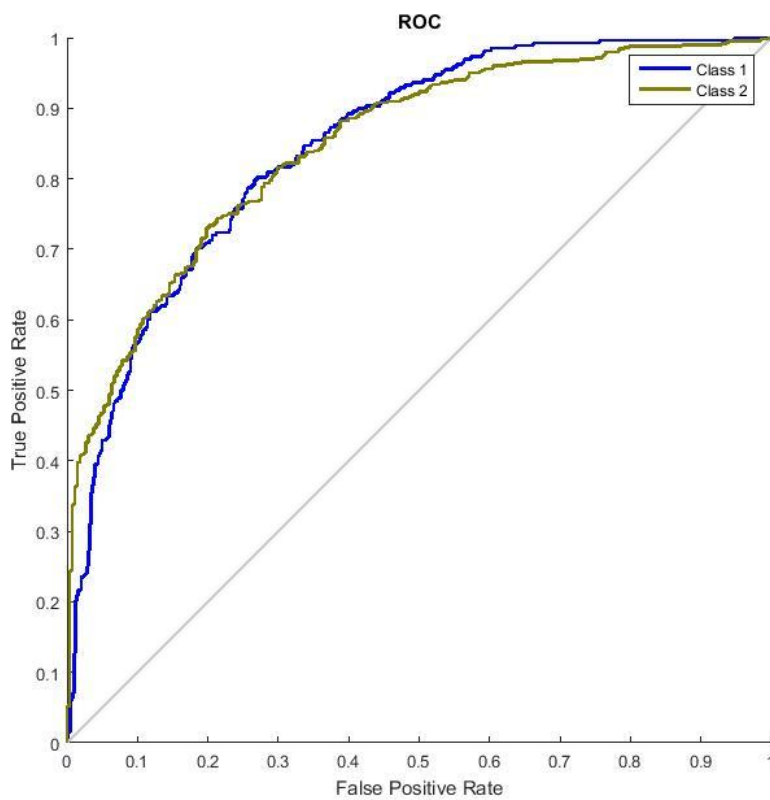
MCC = 0.4984

AUC = 0.7382

- Resilient backpropagation (εντολή Trainrp)



5.1.ια.1 Γραφική Παράσταση Απόδοσης



5.1.ια.2 Καμπύλη ROC

ConfusionMatrix =

164 104

59 441

Accuracy = 0.7878

Precision = 0.8092

Recall = 0.8820

F1 = 0.8440

MCC = 0.5187

AUC = 0.7470

Συνοπτικός Πίνακας Τιμών Αξιολόγησης Απόδοσης των Νευρωνικών Δικτύων

	F ₁ score	MCC	MSE	AUC
Scaled Conjugate Gradient Backpropagation	0.8376	0.4927	0.1526	0.7320
Conjugate gradient backpropagation with Powell-Beale restarts	0.8423	0.5168	0.2653	0.8446
Conjugate gradient backpropagation with Fletcher-Reeves updates	0.8324	0.4782	0.1624	0.7262
Conjugate gradient backpropagation with Polak-Ribière updates	0.8150	0.4592	0.1771	0.7273
Gradient descent with adaptive learning rate backpropagation	0.8212	0.4511	0.1636	0.7162
Gradient descent with momentum backpropagation	0.7886	NaN	0.2731	0.5000
Gradient descent with momentum and adaptive learning rate backpropagation	0.8343	0.5031	0.1470	0.7452
BFGS quasi-Newton backpropagation	0.8340	0.4848	0.1617	0.7299
Levenberg-Marquardt backpropagation	0.8517	0.5344	0.1515	0.7495
One-step secant backpropagation	0.8369	0.4984	0.1552	0.7382
Resilient backpropagation	0.8440	0.5187	0.1535	0.7470

Πίνακας 5.1.1

Συμπεράσματα:

Παρατηρείται παρόμοια απόδοση στα περισσότερα μοντέλα παρά τη χρήση διαφορετικών συναρτήσεων εκπαίδευσης, με εξαίρεση την περίπτωση της συνάρτησης Gradient descent with momentum backpropagation όπου παρά το πέρας του ορίου των 1000 κύκλων εκπαίδευσης (epochs) το σύστημα δεν έχει ρυθμίσει τις παραμέτρους κατάλληλα ώστε να έχει το ελάχιστο σφάλμα και τη μέγιστη απόδοση και αδυνατεί να ταξινομήσει μία εκ των δυο κλάσεων, κάτι που φαίνεται από τον πίνακα συνάφειας. Για τη συνάρτηση εκείνη έγιναν δοκιμές αυξάνοντας το μέγιστο αριθμό κύκλων εκπαίδευσης σε 2000, 3000 και 10000 με αποτέλεσμα να καταφέρει

να κάνει μία καλύτερη ταξινόμηση όμως πάλι με υψηλό μέσο τετραγωνικό σφάλμα και με το μειονέκτημα ότι καθυστερεί αρκετά καθώς χρειάζεται υπερβολικά μεγάλο αριθμό κύκλων εκπαίδευσης.

5.2 Σύγκριση της Απόδοσης Διαφορετικών Μοντέλων Πρόβλεψης

Στο Κεφάλαιο 4 έγινε μία παρουσίαση διαφορετικών μεθόδων Μηχανικής Μάθησης για την πρόβλεψη εμφάνισης Σακχαρώδους Διαβήτη. Συνοπτικά οι τιμές των βασικότερων κριτηρίων αξιολόγησης του κάθε συστήματος παρατίθενται στον παρακάτω πίνακα:

	Νευρωνικό Δίκτυο	K-NN	Logistic Regression	SVM
Accuracy	0.7773	0.8438	0.7773	0.7682
Precision	0.7975	0.8711	0.7277	0.7184
Recall	0.8820	0.8920	0.5784	0.5522
F ₁ score	0.8376	0.8814	0.6445	0.6245
MCC	0.4927	0.6529	0.4923	0.4693
MSE	0.1526	0.1598	0.2227	0.2318
AUC	0.7320	0.9220	0.8446	0.8435

Πίνακας 5.2.1

Από τον παραπάνω πίνακα φαίνεται πως η βέλτιστη πρόβλεψη τιμών των κλάσεων επιτυγχάνεται μέσω του μοντέλου K-NN (AUC=0.9220). Επίσης το μοντέλο K-NN έχει την υψηλότερη Ορθότητα (Accuracy), αλλά η μέτρηση της Ορθότητας μόνο δεν επαρκεί καθώς για παράδειγμα ένα μοντέλο που θα προέβλεπε μόνο 0 σε οποιαδήποτε είσοδο θα μπορούσε να έχει υψηλή Ορθότητα αν τα παραδείγματα αποτελούνταν κυρίως από σύνολα δεδομένων εκπαίδευσης με μεταβλητές εξόδου $y^{(i)}=0$. Οπότε προτιμότερη είναι η χρήση του F_1 score ή του συντελεστή Matthews Correlation Coefficient (MCC), και παρατηρείται ότι το μοντέλο πρόβλεψης K-NN έχει και εκεί την υψηλότερη απόδοση (F_1 score=0.8814 και MCC=0.6529). Ως προς το μέσο τετραγωνικό σφάλμα MSE, καλύτερη απόδοση έχει το Νευρωνικό Δίκτυο με το μικρότερο σφάλμα (MSE=0.1526) και ακολουθεί το K-NN με μικρή διαφορά (MSE=0.1598).

Επειδή όμως το Νευρωνικό Δίκτυο έχει χαμηλότερες τιμές τόσο ως προς την ακρίβεια μέσω F_1 score και Matthews Correlation Coefficient (MCC) όσο και προς την ικανότητα να ταξινομεί σωστά όπως φαίνεται μέσω του εμβαδού κάτω από την καμπύλη ROC (AUC), ως βέλτιστη μέθοδος για την πρόβλεψη της πιθανότητας εμφάνισης Σακχαρώδους Διαβήτη μέσω του συνόλου δεδομένων εκπαίδευσης που χρησιμοποιήθηκε θεωρείται το μοντέλο K-NN.

Και στις 4 μεθόδους έγινε ένας επιπλέον έλεγχος με δύο ακραίες υποθετικές περιπτώσεις s1 και s2, με την s1 να έχει τιμές και στα 8 χαρακτηριστικά που να δείχνουν λογικά πως το άτομο σίγουρα πάσχει ή θα πάσχει από Διαβήτη (υψηλή γλυκόζη στο αίμα, χαμηλή ινσουλίνη, μεγάλη ηλικία, οικογενειακό ιστορικό Διαβήτη κλπ) και την s2 να έχει τιμές που να δείχνουν πως δεν θα πάσχει, και παρατηρήθηκε πως και οι 4 μέθοδοι ανταποκρίθηκαν όπως ήταν αναμενόμενο και κατηγοριοποίησαν σωστά τις δύο νέες αυτές περιπτώσεις.

Συνοψίζοντας, φαίνεται πως οι παραπάνω μέθοδοι Μηχανικής Μάθησης είναι δυνατόν να κάνουν αρκετά ακριβείς προβλέψεις ως προς την πιθανότητα εμφάνισης Σακχαρώδους Διαβήτη εισάγοντας δεδομένα με τα ίδια χαρακτηριστικά (features). Ιδιαίτερα καλή απόδοση για το συγκεκριμένο πρόβλημα είχαν το μοντέλο K-NN και τα Νευρωνικά Δίκτυα.

Βιβλιογραφία

Alberts, B. et al. (2010)

Alberts, B., Bray, D., Hopkin, K., Johnson, A., Lewis, J., Raff, M., Roberts, K., Walter, P.
“Essential Cell Biology” 3rd Edition, Garland Science

American Diabetes Association (2013)

“Economic Costs of Diabetes in the U.S. in 2012”, Diabetes Care 36: p1033-1046

Coiera, E. (2015)

“Guide to Health Informatics”, 3rd Edition, CRC Press

Cortes, C. & Vapnik, V. (1995)

“Support Vector Networks”, Machine Learning 20(3):p273-297

DeFronzo, R., et al. (2015)

DeFronzo, R., Ferrannini, E., Zimmet, P., Alberti, K.G.

“International Textbook of Diabetes Mellitus”, 4th Edition, John Wiley & Sons Ltd.

Estes, W.K. (1975)

“Handbook of Learning and cognitive processes”, Hillsdale, N.J., L. Erlbaum Assoc.

Hofstadter, D.R. (1979)

“Gödel, Escher, Bach: An Eternal Golden Braid” Basic Books, Inc.

King, R. et al. (1992)

King R., Muggleton S., Lewis R., Sternberg M.

“Drug design by machine learning: The use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase”,
Proceedings of the National Academy of Sciences of the United States of America 89(23):
p11322-11326

Langley, P. (2011)

“The Changing Science of Machine Learning” *Machine Learning* 82 (3): p275-279

Ledley, R. & Lusted, L. (1959)

“Reasoning Foundations of Medical Diagnosis”, *Science* 130(3366): p9-21

Minsky, M. (1986)

“The Society of Mind”, N.Y., Simon & Schuster

Russel, S. & Norvig, P. (1995)

“Artificial Intelligence: A Modern Approach”, New Jersey, Prentice Hall

Searle, J. R. (1980)

“Minds, brains and programs” *Behavioral and Brain Sciences* 3(3): p417-457

Shortliffe, E. & Clancey, W. (1984)

“Readings in Artificial Intelligence: The First Decade”, Addison Wesley

Αλαφοδήμος, Κ. (2014)

“Έλεγχος Παραγωγικών Διεργασιών”, Ανώτατο Εκπαιδευτικό Ίδρυμα Πειραιά Τ.Τ., Τμήμα Μηχανικών Αυτοματισμού, Πειραιάς 2014

Ασλάνη, Μ. (2013)

“Πρόβλεψη Επισημασμένων Δανείων”, MSc, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Διατμηματικό Πρόγραμμα Μεταπτυχιακών Σπουδών “Πληροφορική και Διοίκηση” Τμημάτων Πληροφορικής και Οικονομικών Επιστημών

Βλαχάβας, Ι. et al. (2006)

Βλαχάβας Ι., Κεφαλάς Π., Βασιλειάδης Ν., Κόκκορας Φ., Σακελλαρίου Η.
“Τεχνητή Νοημοσύνη” 3rd Edition, Β.Γκιούρδας Εκδοτική

Διαμαντάρας, Κ. (2010)

“Τεχνητά Νευρωνικά Δίκτυα”, Εκδόσεις Κλειδάριθμος

Καλλιγερόπουλος, Δ. & Βασιλειάδου, Σ. (2005)

Ιστορία της Τεχνολογίας και των Αυτομάτων, Αθήνα, Σύγχρονη Εκδοτική

Νάκας, Χ. (2002)

“Προσαρμογή καμπύλης, Στατιστική Συμπερασματολογία, Επεκτάσεις και Εφαρμογές στην Ανάλυση των Καμπυλών Λειτουργικού Χαρακτηριστικού Δέκτη”, PhD, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Σχολή Θετικών Επιστημών, Τμήμα Μαθηματικών

Ντζούφρας, Ι. & Περπέρογλου, Α. (2009)

“Εισαγωγή στην Βιοστατιστική και την Επιδημιολογία”, 4^η έκδοση
Αθήνα, 2009

Πόρποδας, Κ.Δ., (1996)

“Γνωστική Ψυχολογία, Τόμος 1, η Διαδικασία της Μάθησης”, Αθήνα, Ελληνικά Γράμματα

Ιστότοποι από τους οποίους αντλήθηκε υλικό:

<http://www.mayoclinic.org/diseases-conditions/diabetes/basics/complications/con-20033091>
προσπελάστηκε στις 20/07/2016

<http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes>
προσπελάστηκε στις 20/07/2016

<https://www.coursera.org/learn/machine-learning>
προσπελάστηκε στις 20/07/2016

<http://cs229.stanford.edu/materials.html>
προσπελάστηκε στις 20/07/2016

<http://machinelearningmastery.com/>
προσπελάστηκε στις 20/07/2016

<http://www.robots.ox.ac.uk/~az/lectures/ml>
προσπελάστηκε στις 20/07/2016

Παράρτημα

Κατάλογος Εικόνων

Εικόνα 3.1.α Κανονική Προσαρμογή σε Πρόβλημα Παρεμβολής.....	25
Εικόνα 3.1.β Overfitting σε Πρόβλημα Παρεμβολής.....	25
Εικόνα 3.1.γ Underfitting σε Πρόβλημα Παρεμβολής.....	25
Εικόνα 3.1.α Κανονική Προσαρμογή σε Πρόβλημα Ταξινόμησης.....	25
Εικόνα 3.1.β Overfitting σε Πρόβλημα Ταξινόμησης.....	25
Εικόνα 3.1.γ Underfitting σε Πρόβλημα Ταξινόμησης.....	25
Εικόνα 3.2.α Απεικόνιση Σιγμοειδούς Συνάρτησης.....	29
Εικόνα 3.5.α Σχηματική Αναπαράσταση Τεχνητού Νευρωνικού Δικτύου.....	42
Εικόνα 4.2.α Απόδοση Τεχνητού Νευρωνικού Δικτύου.....	47
Εικόνα 4.2.β Καμπύλη ROC Τεχνητού Νευρωνικού Δικτύου.....	52
Εικόνα 4.3.α Γραφική Παράσταση Εύρεσης Βέλτιστου Κ.....	54
Εικόνα 4.3.β Καμπύλη ROC μοντέλου KNN.....	56
Εικόνα 4.4.α Καμπύλη ROC μοντέλου με Logistic Regression.....	58
Εικόνα 4.5.α Καμπύλη ROC μοντέλου με SVM.....	60
5.1.α.1 Γραφική Παράσταση Απόδοσης για Scaled Conjugate Gradient Backpropagation....	64
5.1.α.2 Καμπύλη ROC για Scaled Conjugate Gradient Backpropagation.....	64
5.1.β.1 Γραφική Παράσταση Απόδοσης για Conjugate gradient backpropagation with Powell-Beale restarts.....	65
5.1.β.2 Καμπύλη ROC για Conjugate gradient backpropagation with Powell-Beale restarts.....	65
5.1.γ.1 Γραφική Παράσταση Απόδοσης για Conjugate gradient backpropagation with Fletcher-Reeves updates.....	66
5.1.γ.2 Καμπύλη ROC για Conjugate gradient backpropagation with Fletcher-Reeves updates.....	66
5.1.δ.1 Γραφική Παράσταση Απόδοσης για Conjugate gradient backpropagation with Polak-Ribière updates.....	67
5.1.δ.2 Καμπύλη ROC για Conjugate gradient backpropagation with Polak-Ribière updates.....	67
5.1.ε.1 Γραφική Παράσταση Απόδοσης για Gradient descent with adaptive learning rate backpropagation.....	68

5.1.ε.2 Καμπύλη ROC για Gradient descent with adaptive learning rate backpropagation.....	68
5.1.στ.1 Γραφική Παράσταση Απόδοσης για Gradient descent with momentum backpropagation.....	69
5.1.στ.2 Καμπύλη ROC για Gradient descent with momentum backpropagation.....	69
5.1.ζ.1 Γραφική Παράσταση Απόδοσης για Gradient descent with momentum and adaptive learning rate backpropagation.....	70
5.1.ζ.2 Καμπύλη ROC για Gradient descent with momentum and adaptive learning rate backpropagation.....	70
5.1.η.1 Γραφική Παράσταση Απόδοσης για BFGS quasi-Newton backpropagation.....	71
5.1.η.2 Καμπύλη ROC για BFGS quasi-Newton backpropagation.....	71
5.1.θ.1 Γραφική Παράσταση Απόδοσης για Levenberg-Marquardt backpropagation.....	72
5.1.θ.2 Καμπύλη ROC για Levenberg-Marquardt backpropagation.....	72
5.1.ι.1 Γραφική Παράσταση Απόδοσης για One-step secant backpropagation.....	73
5.1.ι.2 Καμπύλη ROC για One-step secant backpropagation.....	73
5.1.ια.1 Γραφική Παράσταση Απόδοσης για Resilient backpropagation.....	74
5.1.ια.2 Καμπύλη ROC για Resilient backpropagation.....	74

Κατάλογος Πινάκων

Πίνακας 3.1.1 Πίνακας Συνάφειας Confusion Matrix.....	26
Πίνακας 5.1.1 Συνοπτικός Πίνακας Τιμών Αξιολόγησης Απόδοσης Νευρωνικών Δικτύων...	75
Πίνακας 5.2.1 Πίνακας Τιμών Αξιολόγησης Διαφορετικών Μοντέλων Πρόβλεψης.....	76