



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΕΚΠΑΙΔΕΥΣΗ ΒΑΡΩΝ ΣΕ ΔΙΚΤΥΑ ΑΚΤΙΝΙΚΗΣ ΒΑΣΗΣ

Αντώνιος Τόμπρος

Επιβλέπων: Ιωάννης Γ. Τσούλος
Αναπληρωτής Καθηγητής

Αρτα, Νοέμβριος 2020

Weight training in Radial Basis Function

Εγκρίθηκε από τριμελή εξεταστική επιτροπή
Τόπος, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπων καθηγητής
Ιωάννης Γ. Τσούλος,
2. Μέλος επιτροπής
Τζάλλας Αλέξανδρος,
3. Μέλος επιτροπής
Χαριλόγης Βασίλειος,

© Τόμπρος, Αντώνιος, 2020.
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα πτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Τόμπρος, Αντώνιος

Υπογραφή

ΕΥΧΑΡΙΣΤΙΕΣ

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, Ιωάννη Γ. Τσούλο για την συνεργασία, την υποστήριξη αλλά και τις γνώσεις που μου προσέφερε όλο αυτό το διάστημα και κατά τη διάρκεια των σπουδών μου.

Στη συνέχεια, θα ήθελα να ευχαριστήσω τους καθηγητές του τμήματος Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Ιωαννίνων για τις γνώσεις, τις εμπειρίες και τον χρόνο που μου χάρισαν λύνοντας κάθε μου απορία.

Τέλος, το μεγαλύτερο ευχαριστώ, το οφείλω στην οικογένειά μου για την στήριξη που μου παρείχαν, σε πολλά επίπεδα, όλα αυτά τα χρόνια καθώς και τους φίλους-συμφοιτητές που μοιράστηκα μαζί τους αξέχαστα φοιτητικά χρόνια.

ΠΕΡΙΛΗΨΗ

Οι γενετικοί αλγόριθμοι είναι εμπνευσμένοι από τη φύση και συγκεκριμένα από τη θεωρία εξέλιξης των ειδών. Έχουν εφαρμοστεί σε πολλούς επιστημονικούς τομείς και πολλά από τα προβλήματα που επιλύουν είναι μείζονος σημασίας για αυτό και τα τελευταία χρόνια μεγαλώνει το ενδιαφέρον των ανθρώπων για αυτούς. Στη παρούσα εργασία εξηγούνται διάφορες επιστημονικές έννοιες όπως η μηχανική μάθηση και οι γενετικοί αλγόριθμοι. Στη συνέχεια, υλοποιήθηκε μια εφάρμογη σε γλώσσα Java στο περιβάλλον IntelliJ για την κατηγοροποίηση κάποιων συγκεκριμένων δεδομένων που διαβάζονται μέσα από τρία διαφορετικά αρχεία. Τέλος, παρουσιάζεται η εφαρμογή που υλοποιήθηκε και τα αποτελέσματα που βγάζει χρησιμοποιώντας το γνωστό αρχείο iris σε μορφή csv.

Λέξεις-κλειδιά: Νευρωνικά Δίκτυα, Δίκτυα Ακτινικής Βάσης, Μηχανική Μάθηση, Γενετικοί Αλγόριθμοι

ABSTRACT

Genetic algorithms are inspired by nature and specifically by the theory of species evolution. They have been applied in many scientific areas and many of the problems they solve are of major importance for this, and in recent years people's interest has grown for them. This paper explains various scientific concepts such as machine learning and genetic algorithms. A Java application was then implemented in the IntelliJ environment to categorize certain data read through three different files. Finally, the application implemented and the results obtained using the known iris file in csv format are presented.

Keywords: Neural Networks, Radial Basis Functions, Machine Learning, Genetic Algorithms

Περιεχόμενα

1	Εισαγωγικές έννοιες	4
1.1	Προσαρμογή δεδομένων	4
1.2	Κατηγοριοποίηση δεδομένων	4
1.3	Σκοπός της παρούσας εργασίας	4
2	Μηχανική μάθηση	5
2.1	Ορισμός	5
2.2	Μάθηση χωρίς επίβλεψη	5
2.3	Μάθηση με επίβλεψη	5
2.4	Ενισχυτική μάθηση	6
2.5	Δένδρα απόφασης	6
2.6	Νευρωνικά δίκτυα	6
2.7	Γενετικοί αλγόριθμοι	7
3	Δίκτυα RBF	7
3.1	Ομαδοποίηση	7
3.2	Αλγόριθμος K-Μέσων	8
3.3	Συναρτήσεις ακτινικής βάσης	9
3.4	Εκπαίδευση Δικτύου Συναρτήσεων Ακτινικής Βάσης	10
3.5	Χρήση Γενετικών Αλγορίθμων στην Εκπαίδευση RBF	10
4	Η εφαρμογή	11
4.1	Αρχεία εισόδου	11
4.2	Η κατηγορία GaussianNode	16
4.3	Η κατηγορία OutputNode	17
4.4	Η κατηγορία RBFNetwork	18

4.5	Η κατηγορία RBFClassifier	20
4.6	Η κατηγορία main	24
4.5	Παράδειγμα εκτέλεσης και αποτελέσματα	26

ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ/ΕΙΚΟΝΩΝ

3.1	Συνάρτηση K-Μέσων.	8
3.2	Διάγραμμα αρχιτεκτονικής του Radial Basis Function Network.	9
4.1	iris.csv.	12
4.2	seeds.csv.	15
4.3	prestige.csv.	16
4.4	Κατηγορία GaussianNode.	17
4.5	Κατασκευαστής και Συναρτήσεις Κατηγορίας GaussianNode.	17
4.6	Κατηγορία OutputNode.	17
4.7	Κατασκευαστής και Συνάρτηση Κατηγορίας OutputNode.	18
4.8	Κατηγορία RBFNetwork.	19
4.9	Κατασκευαστής Κατηγορίας RBFNetwork.	19
4.10	Συνάρτηση get_output της Κατηγορίας RBFNetwork.	19
4.11	Συνάρτηση back_propagate της Κατηγορίας RBFNetwork.	19
4.12	Κατηγορία RBFClassifier.	20
4.13	Κατασκευαστής και Συνάρτηση classify της Κατηγορίας RBFClassifier.	20
4.14	Συνάρτηση learn της Κατηγορίας RBFClassifier.	21
4.15	Συναρτήσεις set_weights_w και set_weights_w της Κατηγορίας RBFClassifier.	21
4.16	Αρχή της Συνάρτησης kmeans της Κατηγορίας RBFClassifier.	22
4.17	Μέση της Συνάρτησης kmeans της Κατηγορίας RBFClassifier.	22
4.18	Τέλος της Συνάρτησης kmeans της Κατηγορίας RBFClassifier.	23
4.19	Συνάρτηση euclidean_distance της Κατηγορίας RBFClassifier.	23
4.20	Κατηγορίας main.	24
4.21	Συνάρτηση csv_to_float της Κατηγορίας main.	24
4.22	Συνάρτηση float_to_csv της Κατηγορίας main.	25
4.23	Κύρια Συνάρτηση της Κατηγορίας main.	26
4.24	Εισαγωγή των συνολικών εισόδων.	27
4.25	Εισαγωγή των κέντρων/κρυφών κόμβων.	27
4.26	Εισαγωγή αρχείου.	27
4.27	Εμφάνιση πλήθους σωστά ταξινομημένων προτύπων.	27
4.28	Αρχείο αποθήκευσης αποτελεσμάτων.	30

Κεφάλαιο 1

1. Εισαγωγικές Έννοιες

1.1 Προσαρμογή Δεδομένων (Data Fitting)

Δεδομένου ενός συνόλου παρατηρήσεων, κάποιος συχνά θέλει να συμπυκνώσει και να συνοψίσει τα δεδομένα δημιουργώντας ένα "μοντέλο" που εξαρτάται από ρυθμιζόμενες παραμέτρους. Μερικές φορές το μοντέλο είναι απλά μια κατηγορία συναρτήσεων, όπως πολυώνυμα ή τριγωνομετρικές συναρτήσεις, και η προσαρμογή καθορίζει τους κατάλληλους συντελεστές. Η μοντελοποίηση μπορεί επίσης να χρησιμοποιηθεί ως ένα είδος περιορισμένης παρεμβολής, όπου μπορούν να επεκταθούν μερικά σημεία δεδομένων σε μια συνεχή συνάρτηση.

1.2 Κατηγοριοποίηση

Η κατηγοριοποίηση είναι η διαδικασία πρόβλεψης της κατηγορίας-κλάσης των δεδομένων που δόθηκαν. Οι κλάσεις-κατηγορίες ονομάζονται και ως στόχοι/ ετικέτες ή κατηγορίες. Η κατηγοριοποίηση της προγνωστικής μοντελοποίησης είναι το έργο της προσέγγισης μιας λειτουργίας συνάρτησης (f) από μεταβλητές εισόδου (x) σε διακριτές μεταβλητές εξόδου (y). [1] Εφόσον η δημιουργία του μοντέλου υλοποιηθεί, το επόμενο βήμα είναι η αξιολόγησή του. Για να επιτευχθεί αυτό, χρησιμοποιούμε τα δεδομένα εισόδου και το μοντέλο χρησιμοποιείται για να κατηγοριοποιήσει τα δεδομένα αυτά. [2] Εάν η μεταβλητή εξόδου είναι ίδια με αυτή που δόθηκε ως κατηγορία-κλάση των δεδομένων εισόδου τότε η κατηγοριοποίηση έγινε σωστά. Ωστόσο η ακρίβεια του μοντέλου υπολογίζεται από το ποσοστό των δειγμάτων δοκιμής που κατηγοριοποιήθηκαν σωστά από το μοντέλο. Στην περίπτωση που το μοντέλο κριθεί αποδεκτό, τότε μπορεί να χρησιμοποιηθεί για την κατηγοριοποίηση μελλοντικών δειγμάτων δεδομένων, των οποίων η κατηγοριοποίηση είναι άγνωστη. [3]

1.3 Σκοπός της παρούσας εργασίας

Στόχος της παρούσας εργασίας είναι η κατανόηση της έννοιας “μηχανική μάθηση” αφού είναι μία από τις σημαντικότερες λειτουργίες των τεχνητών νευρωνικών δικτύων. Ο ρόλος της με τα νευρωνικά δίκτυα. Κυρίως όμως η πλήρης κατανόηση των δικτύων ακτινικής βάσης (Radial Basis Functions) και η σύνδεσή τους με την μηχανική μάθηση αλλά και τους γενετικούς αλγορίθμους.

Κεφάλαιο 2

Μηχανική Μάθηση (Machine Learning)

2.1 Ορισμός

Η Μηχανική Μάθηση είναι μια από τις σημαντικότερες λειτουργίες των τεχνητών νευρωνικών δικτύων. Με τον όρο Μηχανική Μάθηση εννοείτε η δημιουργία προτύπων ή μοντέλων μέσα από ένα σύνολο δεδομένων, από ένα υπολογιστικό σύστημα. [4] Ο όρος μάθηση ορίζει την ικανότητα να μαθαίνουν από το περιβάλλον και να βελτιώνουν την απόδοσή τους μέσω της εκπαίδευσης. Αυτό επιτυγχάνεται μέσα από την αλλαγή στις τιμές των συναπτικών βαρών του νευρωνικού δικτύου που ουσιαστικά είναι ο τρόπος για να μαθαίνει σχετικά με το περιβάλλον του. [5]

2.2 Μηχανική Μάθηση χωρίς επίβλεψη (Unsupervised Learning)

Η Μηχανική Μάθηση χωρίς επίβλεψη (Unsupervised Learning) ή μάθηση από παρατήρηση είναι μια πολλά υποσχόμενη υπόθεση του μέλλοντος. Η Μηχανική Μάθηση είναι μια διαδικασία όπου ο αλγόριθμος κατασκευάζει ένα μοντέλο για κάποιο σύνολο εισόδων χωρίς όμως να γνωρίζει τις επιθυμητές εξόδους δηλαδή την κατηγοριοποίησή τους. [5]

Αυτό το είδος μάθησης ονομάζεται και προσωπική μάθηση καθώς τα δίκτυα σε αντίθεση με τη τεχνική μηχανικής μάθησης με επίβλεψη που θα περιγραφεί παρακάτω, δεν χρησιμοποιεί καμία εξωτερική επιρροή για να γίνει προσαρμογή στα βάρη. Η εκπαίδευση πραγματοποιείται μέσω μιας ανεξάρτητης συνάρτησης. [7] Τα δίκτυα αυτής της τεχνικής ψάχνουν τυχόν ομοιότητες, τάσεις ή χαρακτηριστικά στο σύνολο των εισόδων τους που επεξεργάζεται και τα κατατάσσει σε κατηγορίες-κλάσεις ή να δημιουργεί και νέες εφόσον αυτό κρίνεται απαραίτητο. Χρησιμοποιείται κυρίως σε προβλήματα ανάλυσης συσχετισμών (Association Analysis), ομαδοποίησης (Clustering) και ανίχνευση ανωμαλιών (Anomaly Detection). [6]

2.3 Μηχανική Μάθηση με επίβλεψη (Supervised Learning)

Ονομάζεται Μηχανική Μάθηση με επίβλεψη ή μάθηση με παραδείγματα διότι είναι η διαδικασία ενός αλγορίθμου που κατασκευάζει μια συνάρτηση που απεικονίζει τις δεδομένες εισόδους ή αλλιώς σύνολο εκπαίδευσης σε γνωστές για τον αλγόριθμο επιθυμητές εξόδους. Έχει ως απώτερο στόχο τη γενίκευση της συνάρτησης αυτής και για εισόδους με άγνωστη έξοδο. Για να επιτευχθεί ο στόχος τα βάρη προσαρμόζονται σταδιακά με σκοπό να μειωθεί το σφάλμα μεταξύ της απόκρισης του δικτύου και του επιθυμητού αποτελέσματος. [8] Χρησιμοποιείται κυρίως σε προβλήματα ταξινόμησης (Classification), παλινδρόμησης (Regression) και πρόβλεψης (Prediction).

2.4 Ενισχυτική Μάθηση (Reinforcement Learning)

Όπως και οι υπόλοιποι τρόποι μάθησης έτσι και η Ενισχυτική Μάθηση αποτελείται από τον αλγόριθμο που μαθαίνει μια στρατηγική ενεργειών μέσα από άμεση αλληλεπίδραση με το περιβάλλον. Δηλαδή πρόκειται για τη λήψη κατάλληλων ενεργειών για τη μεγιστοποίηση της ανταμοιβής σε μια συγκεκριμένη κατάσταση. [9] Η ενισχυτική μάθηση διαφέρει από την μηχανική μάθηση με επίβλεψη διότι στην μάθηση με επίβλεψη το σύνολο εκπαίδευσης έχει γνωστές για τον αλγόριθμο τις επιθυμητές εξόδους. Έτσι, ο παράγοντας ενίσχυσης αποφασίζει τι να κάνει για να εκτελέσει τη δεδομένη εργασία και το σύνολο εκπαίδευσης είναι υποχρεωμένο να μάθει από την εμπειρία του.

Υπάρχουν δύο τρόποι ενισχυτικής μάθησης, ο θετικός (Positive) και ο αρνητικός (Negative). Η θετική ενισχυτική μάθηση ορίζεται ως όταν ένα συμβάν, συμβαίνει λόγω μιας συγκεκριμένης συμπεριφοράς, αυξάνει τη δύναμη και τη συχνότητα της συμπεριφοράς. Ενώ η αρνητική ορίζεται ως ενίσχυση μιας συμπεριφοράς επειδή μια αρνητική κατάσταση σταματά ή αποφεύγεται. Τέλος, χρησιμοποιείται κυρίως σε προβλήματα σχεδιασμού (Planning).

2.5 Δένδρα Απόφασης (Decision Trees)

Τα Δένδρα Απόφασης είναι μια εποπτευόμενη τεχνική μηχανικής μάθησης για την δημιουργία ενός δέντρου αποφάσεων από δεδομένα εκπαίδευσης. Γενικότερα, είναι ο γνωστότερος αλγόριθμος επιβλεπόμενης επαγωγικής μάθησης που έχει εφαρμοστεί με επιτυχία σε πολλούς τομείς όπου απαιτείται ταξινόμηση. Στις δομές των δέντρων, τα φύλλα τους αντιπροσωπεύουν κατηγορίες ταξινόμησης, οι κόμβοι είναι χαρακτηριστικά, ενώ οι κλάδοι δείχνουν τις συνδέσεις χαρακτηριστικών που οδηγούν στις ταξινομήσεις. [10]

2.6 Νευρωνικά Δίκτυα

Τα Νευρωνικά Δίκτυα είναι τεχνητά συστήματα που εμπνεύστηκαν από βιολογικά νευρικά δίκτυα. Νευρωνικό δίκτυο ονομάζεται ένα κύκλωμα διασυνδεδεμένων μονάδων επεξεργασίας που ονομάζουμε Νευρώνες. Ο χαρακτηρισμός του ως δίκτυο συμβαίνει γιατί αποτελείται από ένα σύνολο υπολογιστικών κόμβων που συνδέονται μεταξύ τους, καθώς κάθε υπολογιστικός κόμβος δέχεται ένα σύνολο εισόδων, εκτελεί υπολογισμούς με βάση αυτών και παράγει μια έξοδο. Στους υπολογιστές είναι ένα υπολογιστικό μοντέλο που χρησιμοποιείται για την επίλυση κάποιου υπολογιστικών προβλημάτων. Τα συστήματα αυτά μαθαίνουν να εκτελούν εργασίες εκθέτοντας σε διάφορα σύνολα δεδομένων και παραδείγματα χωρίς κανόνες για συγκεκριμένες εργασίες, έτσι το σύστημα δημιουργεί χαρακτηριστικά αναγνώρισης από τα δεδομένα που έχουν περάσει χωρίς να προγραμματιστεί με μια προγραμματισμένη κατανόηση των συνόλων δεδομένων. [11]

Τα συστατικά ενός νευρωνικού δικτύου περιλαμβάνουν νευρώνες, βάρη, συνδέσεις, προκαταλήψεις, λειτουργία διάδοσης και τέλος έναν κανόνα μάθησης.

2.7 Γενετικοί Αλγόριθμοι (Genetic Algorithms)

Μια από τις μεθόδους που χρησιμοποιεί η Τεχνητή Νοημοσύνη είναι και οι Γενετικοί Αλγόριθμοι. Με αυτό τον τρόπο η Τεχνητή Νοημοσύνη δημιουργεί συστήματα Μηχανικής Μάθησης που βασίζονται στην Εξελικτική Μάθηση (Evolutionary Learning). Όντας γνωστικά συστήματα που μιμούνται την αναπαραγωγική διαδικασία των πληθυσμών απλών βιολογικών οργανισμών κατά τη φάση της φυσικής εξέλιξης. Στην ουσία ο αλγόριθμος αναπαριστά τη διαδικασία της φυσικής επιλογής και φυσικής γενετικής, όπου προκειμένου να παραχθούν απόγονοι της επόμενης γενιάς επιλέγονται τα πιο κατάλληλα άτομα. [12]

Οι αντιστοιχίσεις των γονιδίων είναι σε στοιχείο πίνακα, ο γενότυπος και το χρωμόσωμα σε πίνακα, ο φαινότυπος σε συνάρτηση του προβλήματος ή συνάρτηση καταλληλότητας, διασταύρωση σε ανταλλαγή στοιχείων μεταξύ πινάκων και η μετάλλαξη σε τυχαία αλλαγή στοιχείων σε πίνακες.

Κεφάλαιο 3

Δίκτυα Ακτινικής Βάσης (Radial Basis Functions)

3.1 Ομαδοποίηση (Clustering)

Η ομαδοποίηση είναι μια βασική μέθοδος ανάλυσης δεδομένων καθώς και από τις πιο διάσημες τεχνικές μηχανικής μάθησης χωρίς επίβλεψη. Είναι η οργάνωση των δεδομένων που δεν ανήκουν σε κάποια κατηγορία, σε ομάδες ομοιότητας που ονομάζονται συστάδες. Οι συστάδες ή αλλιώς σύμπλεγμα είναι μια συλλογή στοιχείων δεδομένων τα οποία είναι παρόμοια μεταξύ τους με βάση κάποια χαρακτηριστικά τους και ανόμοια με άλλες ομάδες συστάδων. [14] Οι τεχνικές ομαδοποίησης χωρίζονται σε 3 κατηγορίες, ιεραρχικές (Hierarchical), μερικές (Partitional) και Bayesian και οι κάθε κατηγορία χωρίζεται σε υποκατηγορίες.

Τέλος, είναι μια τεχνική που χρησιμοποιείται σε πολλούς τομείς όπως η μηχανική μάθηση, η εξόρυξη δεδομένων, η αναγνώριση προτύπων, η ανάλυση εικόνων και η βιοπληροφορική.

3.2 Αλγόριθμος K-Μέσων (K-Means)

Ο αλγόριθμος K-Μέσων (MacQueen, 1967) είναι μια τεχνική κεντρικής μερικής (Partitional) ομαδοποίησης, χαρακτηρίζοντάς τους από τους απλούστερους και δημοφιλείς αλγόριθμους μηχανικής μάθησης χωρίς επίβλεψη. Ο στόχος του αλγόριθμου K-Μέσων είναι η ομαδοποίηση παρόμοιων δεδομένων μαζί και να ανακαλυφθούν τα υποκείμενα μοτίβα. Για την επίτευξη αυτού του στόχου, ο αλγόριθμος αναζητά για ένα σταθερό αριθμό (k) συστάδων στο σύνολο δεδομένων. Αρχικά ορίζεται έναν αριθμό στόχου (k), ο οποίος αναφέρεται στον αριθμό κέντρων ή κεντροειδών που χρειάζεται για το σύνολο δεδομένων. [16] Οι κεντροειδείς είναι η φανταστική ή πραγματική τοποθεσία που αντιπροσωπεύει το κέντρο της συστάδας. Κάθε σημείο δεδομένων κατανέμεται σε κάθε μία από τις συστάδες μειώνοντας το άθροισμα των τετραγώνων εντός των συστάδων.

Η συνάρτηση που χρησιμοποιείται είναι:

$$V = \sum_{i=1}^k \sum_{x_j \in S_i} |x_j - \mu_i|^2$$

3.1 Συνάρτηση K-Μέσων.

όπου υπάρχουν k συστάδες S_i , $i = 1, 2, \dots, k$ και μ_i είναι το κεντροειδές ή το μεσαίο σημείο από όλα τα σημεία.

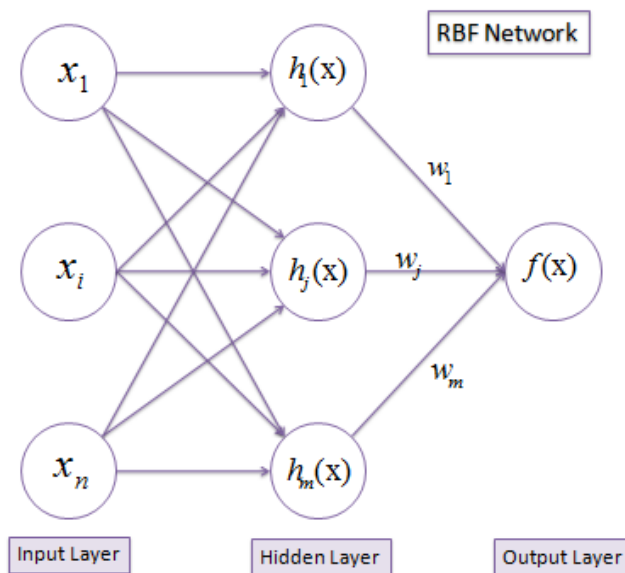
Τα βασικά βήματα του αλγόριθμου είναι τα εξής:

1. Επιλογή του αριθμού των συστάδων.
2. Τυχαία δημιουργία (k) συστάδων και ορισμός των κεντροειδών των συστάδων.
3. Μεταβίβαση του κάθε σημείου στο κεντροειδές της κοντινότερης συστάδας.
4. Υπολογισμός των νέων κεντροειδών των συστάδων.
5. Επανάληψη μέχρι να συγκλίνει ο αλγόριθμος σε κάποιο κριτήριο.

Ο αλγόριθμος ξεκινά διαχωρίζοντας τα αρχικά σημεία σε (k) αρχικά σύνολα είτε τυχαία είτε χρησιμοποιώντας ευριτικά δεδομένα. Στη συνέχεια υπολογίζει το μεσαίο ή το κεντροειδές του κάθε συνόλου, υλοποιεί νέο διαχωρισμό ώστε το κάθε σημείο να σχετίζεται με το κοντινότερο κεντροειδές. Έπειτα τα κεντροειδή ξαναυπολογίζονται για τις νέες ομάδες, ο αλγόριθμος επαναλαμβάνει τα δυο βήματα ωσότου τα σημεία δεν μπορούν να αλλάξουν συστάδες (ή εναλλακτικά τα κεντροειδή παραμένουν αμετάβλητα).

3.3 Συναρτήσεις Ακτινικής Βάσης (Radial Basis Functions)

Τα δίκτυα συναρτήσεων ακτινικής βάσης (RBF) είναι ένας κοινώς χρησιμοποιούμενος τύπος τεχνητού νευρωνικού δικτύου για προβλήματα προσέγγισης της συνάρτησης για προβλήματα μηχανικής μάθησης με επίβλεψη. Τα δίκτυα συναρτήσεων ακτινικής βάσης διακρίνονται από άλλα νευρωνικά δίκτυα λόγω της καθολικής προσέγγισής τους και της ταχύτερης ταχύτητας εκμάθησης. [17] Ένα δίκτυο ακτινικής βάσης είναι ένας τύπος νευρωνικού δικτύου τροφοδοσίας που αποτελείται από τρία επίπεδα, δηλαδή το επίπεδο εισόδου (Input Layer), το κρυφό στρώμα (Hidden Layer) και το επίπεδο εξόδου (Output Layer).



$$f(x) = \sum_{j=1}^m w_j h_j(x)$$

$$h(x) = \exp\left(-\frac{(x-c)^2}{r^2}\right)$$

3.2 Διάγραμμα αρχιτεκτονικής του Radial Basis Function Network.

Οι κρυμμένοι νευρώνες υπολογίζουν τις συναρτήσεις $h(x)$ (με όρισμα το διάνυσμα εισόδου x) που ονομάζονται ακτινικές συναρτήσεις βάσης (Radial Basis Functions). Συνήθως έχουν την ακόλουθη μορφή: υπάρχει κάποιο σημείο (που ονομάζεται κέντρο) για το οποίο παρέχουν μέγιστη τιμή και καθώς απομακρυνόμαστε ακτινικά από το κέντρο η τιμή της συνάρτησης μειώνεται και σχεδόν εκμηδενίζεται για σημεία x που είναι μακριά από το κέντρο. Χαρακτηριστικό παράδειγμα είναι η συνάρτηση Gauss, που είναι γνωστή και ως πυρήνας RBF (RBF kernel).

Το c καθορίζει το κέντρο της συνάρτησης, ενώ η παράμετρος r^2 καθορίζει την ακτίνα της συνάρτησης. Παίρνει τη μέγιστη τιμή στο κέντρο (για $x=c$) και η τιμή ελαττώνεται εκθετικά καθώς απομακρυνόμαστε ακτινικά από το κέντρο. Ο ρυθμός μείωσης καθορίζεται από την τιμή της ακτίνας r^2 . Για μικρές τιμές της ακτίνας ο ρυθμός μείωσης είναι μεγάλος, ενώ το αντίθετο συμβαίνει για μεγάλες τιμές της ακτίνας.

3.4 Εκπαίδευση Δικτύου Συναρτήσεων Ακτινικής Βάσης (RBF)

Οι παράμετροι (βάρη) του δικτύου ακτινικής βάσης που πρέπει να καθοριστούν κατά τη διάρκεια της εκπαίδευσης είναι αρχικά τα κέντρα c και οι ακτίνες r των νευρώνων RBF και μετά τα βάρη w_j . Τα κέντρα c αναπαριστούν τις συντεταγμένες των κέντρων των κρυμμένων νευρώνων ενώ τα βάρη w έχουν την τυπική σημασία των νευρώνων εσωτερικού γινομένου και ορίζουν την εξίσωση υπερεπιπέδου για τους νευρώνες εξόδου. Γενικότερα, μπορεί να χρησιμοποιηθεί διαφορετική εκπαίδευση για τα δύο είδη βαρών.

Αρχικά, εκπαιδεύουμε το κρυφό επίπεδο του δικτύου χρησιμοποιώντας την μέθοδο προσαρμογής καμπύλης γνωστή ως οπίσθια διάδοση (back propagation). Η μέθοδος αυτή διατρέπει στοχαστική προσέγγιση και ταιριάζει σε μια μη γραμμική καμπύλη κατά τη φάση της εκπαίδευσης. Όπου για καθένα από τους κόμβους στο κρυφό στρώμα πρέπει να βρούμε t υποδοχείς και το r που είναι η ακτίνα ή αλλιώς και διακύμανση-διάδοση της συνάρτησης ακτινικής βάσης. Στην δεύτερη φάση της εκπαίδευσης, ενημερώνουμε τα διανύσματα των βαρών μεταξύ των κρυφών επιπέδων και των επιπέδων εξόδου. Στο κρυφό επίπεδο μπορεί να χρησιμοποιηθεί οποιαδήποτε συνάρτηση μετασχηματισμού ώστε να ικανοποιήσει τη μη γραμμική διαχωριστικότητα, καθώς ακόμα και συνδυασμός συναρτήσεων που αντιπροσωπεύουν έναν κόμβο. [18]

Τέλος η οπίσθια διάδοση (back propagation) είναι από τους πιο δημοφιλείς τρόπους εκπαίδευσης που χρησιμοποιούνται για την εκπαίδευση νευρωνικών δικτύων.

3.5 Χρήση Γενετικών Αλγορίθμων στην εκπαίδευση RBF

Συνοπτικά, υπάρχουν τρεις απαιτούμενες παράμετροι εκπαίδευσης για το νευρωνικό δίκτυο ακτινικής βάσης, το κέντρο του, το πλάτος του σε όλες τις κρυφές μονάδες και το βάρος σύνδεσης μεταξύ του κρυφού επιπέδου και της εξόδου. Ο καθορισμός κατάλληλων τιμών για τις τρεις παραμέτρους είναι το κλειδί για τη βελτίωση της μη γραμμικής προσέγγισης του νευρικού δικτύου ακτινικής βάσης. Η κύρια δυσκολία στη λειτουργία του δικτύου ακτινικής βάσης αφορά τη βελτιστοποίηση του κρυμμένου επιπέδου. Όταν μια δυναμική αρχιτεκτονική προτιμάται από μια προκαθορισμένη, η μέθοδος βελτιστοποίησης συχνά συνίσταται σε έναν αλγόριθμο κατάβασης κλίσης που μπορεί να παγιδευτεί σε τοπικά ελάχιστα. Επιπλέον, η λειτουργία ενεργοποίησης επηρεάζει την τελική κατάσταση της διαδικασίας βελτιστοποίησης.

Για την επίλυση των παραπάνω προβλημάτων υπάρχει μια προσέγγιση που συνδυάζει προσαρμογή και κατάρτιση του δικτύου. Η λύση έρχεται από τους γενετικούς αλγόριθμους που παρέχουν έναν ισχυρό κινητήρα για την εξερεύνηση λύσεων σε μη γραμμικά προβλήματα.

ΚΕΦΑΛΑΙΟ 4

Υλοποίηση

4.1 Αρχεία Εισόδου

Τα αρχεία εισόδου που χρησιμοποιήθηκαν είναι το iris.csv, seed.csv και prestige.csv.

	A	B	C	D	E
1	5.1	3.5	1.4	0.2	1
2	4.9	3	1.4	0.2	1
3	4.7	3.2	1.3	0.2	1
4	4.6	3.1	1.5	0.2	1
5	5	3.6	1.4	0.2	1
6	5.4	3.9	1.7	0.4	1
7	4.6	3.4	1.4	0.3	1
8	5	3.4	1.5	0.2	1
9	4.4	2.9	1.4	0.2	1
10	4.9	3.1	1.5	0.1	1
11	5.4	3.7	1.5	0.2	1
12	4.8	3.4	1.6	0.2	1
13	4.8	3	1.4	0.1	1
14	4.3	3	1.1	0.1	1
15	5.8	4	1.2	0.2	1
16	5.7	4.4	1.5	0.4	1
17	5.4	3.9	1.3	0.4	1
18	5.1	3.5	1.4	0.3	1
19	5.7	3.8	1.7	0.3	1
20	5.1	3.8	1.5	0.3	1
21	5.4	3.4	1.7	0.2	1
22	5.1	3.7	1.5	0.4	1
23	4.6	3.6	1	0.2	1
24	5.1	3.3	1.7	0.5	1
25	4.8	3.4	1.9	0.2	1
26	5	3	1.6	0.2	1
27	5	3.4	1.6	0.4	1
28	5.2	3.5	1.5	0.2	1
29	5.2	3.4	1.4	0.2	1
30	4.7	3.2	1.6	0.2	1
31	4.8	3.1	1.6	0.2	1

32	5.4	3.4	1.5	0.4	1
33	5.2	4.1	1.5	0.1	1
34	5.5	4.2	1.4	0.2	1
35	4.9	3.1	1.5	0.2	1
36	5	3.2	1.2	0.2	1
37	5.5	3.5	1.3	0.2	1
38	4.9	3.6	1.4	0.1	1
39	4.4	3	1.3	0.2	1
40	5.1	3.4	1.5	0.2	1
41	5	3.5	1.3	0.3	1
42	4.5	2.3	1.3	0.3	1
43	4.4	3.2	1.3	0.2	1
44	5	3.5	1.6	0.6	1
45	5.1	3.8	1.9	0.4	1
46	4.8	3	1.4	0.3	1
47	5.1	3.8	1.6	0.2	1
48	4.6	3.2	1.4	0.2	1
49	5.3	3.7	1.5	0.2	1
50	5	3.3	1.4	0.2	1
51	7	3.2	4.7	1.4	2
52	6.4	3.2	4.5	1.5	2
53	6.9	3.1	4.9	1.5	2
54	5.5	2.3	4	1.3	2
55	6.5	2.8	4.6	1.5	2
56	5.7	2.8	4.5	1.3	2
57	6.3	3.3	4.7	1.6	2
58	4.9	2.4	3.3	1	2
59	6.6	2.9	4.6	1.3	2
60	5.2	2.7	3.9	1.4	2
61	5	2	3.5	1	2
62	5.9	3	4.2	1.5	2

62	5.9	3	4.2	1.5	2
63	6	2.2	4	1	2
64	6.1	2.9	4.7	1.4	2
65	5.6	2.9	3.6	1.3	2
66	6.7	3.1	4.4	1.4	2
67	5.6	3	4.5	1.5	2
68	5.8	2.7	4.1	1	2
69	6.2	2.2	4.5	1.5	2
70	5.6	2.5	3.9	1.1	2
71	5.9	3.2	4.8	1.8	2
72	6.1	2.8	4	1.3	2
73	6.3	2.5	4.9	1.5	2
74	6.1	2.8	4.7	1.2	2
75	6.4	2.9	4.3	1.3	2
76	6.6	3	4.4	1.4	2
77	6.8	2.8	4.8	1.4	2
78	6.7	3	5	1.7	2
79	6	2.9	4.5	1.5	2
80	5.7	2.6	3.5	1	2
81	5.5	2.4	3.8	1.1	2
82	5.5	2.4	3.7	1	2
83	5.8	2.7	3.9	1.2	2
84	6	2.7	5.1	1.6	2
85	5.4	3	4.5	1.5	2
86	6	3.4	4.5	1.6	2
87	6.7	3.1	4.7	1.5	2
88	6.3	2.3	4.4	1.3	2
89	5.6	3	4.1	1.3	2
90	5.5	2.5	4	1.3	2
91	5.5	2.6	4.4	1.2	2
92	6.1	3	4.6	1.4	2

93	5.8	2.6	4	1.2	2
94	5	2.3	3.3	1	2
95	5.6	2.7	4.2	1.3	2
96	5.7	3	4.2	1.2	2
97	5.7	2.9	4.2	1.3	2
98	6.2	2.9	4.3	1.3	2
99	5.1	2.5	3	1.1	2
100	5.7	2.8	4.1	1.3	2
101	6.3	3.3	6	2.5	3
102	5.8	2.7	5.1	1.9	3
103	7.1	3	5.9	2.1	3
104	6.3	2.9	5.6	1.8	3
105	6.5	3	5.8	2.2	3
106	7.6	3	6.6	2.1	3
107	4.9	2.5	4.5	1.7	3
108	7.3	2.9	6.3	1.8	3
109	6.7	2.5	5.8	1.8	3
110	7.2	3.6	6.1	2.5	3
111	6.5	3.2	5.1	2	3
112	6.4	2.7	5.3	1.9	3
113	6.8	3	5.5	2.1	3
114	5.7	2.5	5	2	3
115	5.8	2.8	5.1	2.4	3
116	6.4	3.2	5.3	2.3	3
117	6.5	3	5.5	1.8	3
118	7.7	3.8	6.7	2.2	3
119	7.7	2.6	6.9	2.3	3
120	6	2.2	5	1.5	3
121	6.9	3.2	5.7	2.3	3
122	5.6	2.8	4.9	2	3
123	7.7	2.8	6.7	2	3

124	6.3	2.7	4.9	1.8	3
125	6.7	3.3	5.7	2.1	3
126	7.2	3.2	6	1.8	3
127	6.2	2.8	4.8	1.8	3
128	6.1	3	4.9	1.8	3
129	6.4	2.8	5.6	2.1	3
130	7.2	3	5.8	1.6	3
131	7.4	2.8	6.1	1.9	3
132	7.9	3.8	6.4	2	3
133	6.4	2.8	5.6	2.2	3
134	6.3	2.8	5.1	1.5	3
135	6.1	2.6	5.6	1.4	3
136	7.7	3	6.1	2.3	3
137	6.3	3.4	5.6	2.4	3
138	6.4	3.1	5.5	1.8	3
139	6	3	4.8	1.8	3
140	6.9	3.1	5.4	2.1	3
141	6.7	3.1	5.6	2.4	3
142	6.9	3.1	5.1	2.3	3
143	5.8	2.7	5.1	1.9	3
144	6.8	3.2	5.9	2.3	3
145	6.7	3.3	5.7	2.5	3
146	6.7	3	5.2	2.3	3
147	6.3	2.5	5	1.9	3
148	6.5	3	5.2	2	3
149	6.2	3.4	5.4	2.3	3
150	5.9	3	5.1	1.8	3

4.1 iris.csv

	A	B	C	D	E	F	G	H
1	15.26	14.84	0.871	5.763	3.312	2.221	5.22	1
2	14.88	14.57	0.8811	5.554	3.333	1.018	4.956	1
3	14.29	14.09	0.905	5.291	3.337	2.699	4.825	1
4	13.84	13.94	0.8955	5.324	3.379	2.259	4.805	1
5	16.14	14.99	0.9034	5.658	3.562	1.355	5.175	1
6	14.38	14.21	0.8951	5.386	3.312	2.462	4.956	1
7	14.69	14.49	0.8799	5.563	3.259	3.586	5.219	1
8	14.11	14.1	0.8911	5.42	3.302	2.7	5	1
9	16.63	15.46	0.8747	6.053	3.465	2.04	5.877	1
10	16.44	15.25	0.888	5.684	3.505	1.969	5.533	1
11	15.26	14.85	0.8696	5.714	3.242	4.543	5.314	1
12	14.03	14.16	0.8796	5.438	3.201	1.717	5.001	1
13	13.89	14.02	0.888	5.439	3.199	3.986	4.738	1
14	13.78	14.06	0.8759	5.479	3.156	3.136	4.872	1
15	13.74	14.05	0.8744	5.482	3.114	2.932	4.825	1
16	14.59	14.28	0.8993	5.351	3.333	4.185	4.781	1
17	13.99	13.83	0.9183	5.119	3.383	5.234	4.781	1
18	15.69	14.75	0.9058	5.527	3.514	1.599	5.046	1
19	14.7	14.21	0.9153	5.205	3.466	1.767	4.649	1
20	12.72	13.57	0.8686	5.226	3.049	4.102	4.914	1
21	14.16	14.4	0.8584	5.658	3.129	3.072	5.176	1
22	14.11	14.26	0.8722	5.52	3.168	2.688	5.219	1
23	15.88	14.9	0.8988	5.618	3.507	0.7651	5.091	1
24	12.08	13.23	0.8664	5.099	2.936	1.415	4.961	1
25	15.01	14.76	0.8657	5.789	3.245	1.791	5.001	1
26	16.19	15.16	0.8849	5.833	3.421	0.903	5.307	1
27	13.02	13.76	0.8641	5.395	3.026	3.373	4.825	1
28	12.74	13.67	0.8564	5.395	2.956	2.504	4.869	1
29	14.11	14.18	0.882	5.541	3.221	2.754	5.038	1
30	14.07	14.07	0.8824	5.541	3.221	2.754	5.038	1
32	15.49	14.94	0.8724	5.757	3.371	3.412	5.228	1
33	14.09	14.41	0.8529	5.717	3.186	3.32	5.299	1
34	13.94	14.17	0.8728	5.585	3.15	2.124	5.012	1
35	15.05	14.68	0.8779	5.712	3.328	2.129	5.36	1
36	16.12	15	0.9	5.709	3.485	2.27	5.443	1
37	16.2	15.27	0.8734	5.826	3.464	2.823	5.527	1
38	17.08	15.38	0.9079	5.832	3.683	2.956	5.484	1
39	14.8	14.52	0.8823	5.656	3.288	3.112	5.309	1
40	14.28	14.17	0.8944	5.397	3.298	6.685	5.001	1
41	13.54	13.85	0.8871	5.348	3.156	2.587	5.178	1
42	13.5	13.85	0.8852	5.351	3.158	2.249	5.176	1
43	13.16	13.55	0.9009	5.138	3.201	2.461	4.783	1
44	15.5	14.86	0.882	5.877	3.396	4.711	5.528	1
45	15.11	14.54	0.8986	5.579	3.462	3.128	5.18	1
46	13.8	14.04	0.8794	5.376	3.155	1.56	4.961	1
47	15.36	14.76	0.8861	5.701	3.393	1.367	5.132	1
48	14.99	14.56	0.8883	5.57	3.377	2.958	5.175	1
49	14.79	14.52	0.8819	5.545	3.291	2.704	5.111	1
50	14.86	14.67	0.8676	5.678	3.258	2.129	5.351	1
51	14.43	14.4	0.8751	5.585	3.272	3.975	5.144	1
52	15.78	14.91	0.8923	5.674	3.434	5.593	5.136	1
53	14.49	14.61	0.8538	5.715	3.113	4.116	5.396	1
54	14.33	14.28	0.8831	5.504	3.199	3.328	5.224	1
55	14.52	14.6	0.8557	5.741	3.113	1.481	5.487	1
56	15.03	14.77	0.8658	5.702	3.212	1.933	5.439	1
57	14.46	14.35	0.8818	5.388	3.377	2.802	5.044	1
58	14.92	14.43	0.9006	5.384	3.412	1.142	5.088	1
59	15.38	14.77	0.8857	5.662	3.419	1.999	5.222	1
60	12.11	13.47	0.8392	5.159	3.032	1.502	4.519	1
61	11.42	12.86	0.8683	5.008	2.85	2.7	4.607	1
62	11.23	12.63	0.884	4.902	2.879	2.269	4.703	1
63	12.36	13.19	0.8923	5.076	3.042	3.22	4.605	1
64	13.22	13.84	0.868	5.395	3.07	4.157	5.088	1
65	12.78	13.57	0.8716	5.262	3.026	1.176	4.782	1
66	12.88	13.5	0.8879	5.139	3.119	2.352	4.607	1
67	14.34	14.37	0.8726	5.63	3.19	1.313	5.15	1
68	14.01	14.29	0.8625	5.609	3.158	2.217	5.132	1
69	14.37	14.39	0.8726	5.569	3.153	1.464	5.3	1
70	12.73	13.75	0.8458	5.412	2.862	3.533	5.067	1
71	17.63	15.98	0.8673	6.191	3.561	4.076	6.06	2
72	16.84	15.67	0.8623	5.998	3.484	4.675	5.877	2
73	17.26	15.73	0.8763	5.978	3.594	4.539	5.791	2
74	19.11	16.26	0.9081	6.154	3.93	2.936	6.079	2
75	16.82	15.51	0.8786	6.017	3.486	4.004	5.841	2
76	16.77	15.62	0.8638	5.927	3.438	4.92	5.795	2
77	17.32	15.91	0.8599	6.064	3.403	3.824	5.922	2
78	20.71	17.23	0.8763	6.579	3.814	4.451	6.451	2
79	18.94	16.49	0.875	6.445	3.639	5.064	6.362	2
80	17.12	15.55	0.8892	5.85	3.566	2.858	5.746	2
81	16.53	15.34	0.8823	5.875	3.467	5.532	5.88	2
82	18.72	16.19	0.8977	6.006	3.857	5.324	5.879	2
83	20.2	16.89	0.8894	6.285	3.864	5.173	6.187	2
84	19.57	16.74	0.8779	6.384	3.772	1.472	6.273	2
85	19.51	16.71	0.878	6.366	3.801	2.962	6.185	2
86	18.27	16.09	0.887	6.173	3.651	2.443	6.197	2
87	18.88	16.26	0.8969	6.084	3.764	1.649	6.109	2
88	18.98	16.66	0.859	6.549	3.67	3.691	6.498	2
89	21.16	17.21	0.8989	6.573	4.033	5.78	6.231	2
90	20.88	17.05	0.9031	6.45	4.032	5.016	6.321	2
91	20.1	16.99	0.8746	6.581	3.785	1.955	6.449	2
92	18.76	16.2	0.8984	6.172	3.796	3.12	6.053	2
93	18.81	16.29	0.8906	6.272	3.693	3.237	6.053	2

94	18.59	16.05	0.9066	6.037	3.86	6.001	5.877	2
95	18.36	16.52	0.8452	6.666	3.485	4.933	6.448	2
96	16.87	15.65	0.8648	6.139	3.463	3.696	5.967	2
97	19.31	16.59	0.8815	6.341	3.81	3.477	6.238	2
98	18.98	16.57	0.8687	6.449	3.552	2.144	6.453	2
99	18.17	16.26	0.8637	6.271	3.512	2.853	6.273	2
100	18.72	16.34	0.881	6.219	3.684	2.188	6.097	2
101	16.41	15.25	0.8866	5.718	3.525	4.217	5.618	2
102	17.99	15.66	0.8932	5.89	3.694	2.068	5.837	2
103	19.46	16.5	0.8985	6.113	3.892	4.308	6.009	2
104	19.18	16.63	0.8717	6.369	3.681	3.357	6.229	2
105	18.95	16.42	0.8829	6.248	3.755	3.368	6.148	2
106	18.83	16.29	0.8917	6.037	3.786	2.553	5.879	2
107	18.85	16.17	0.9056	6.152	3.806	2.843	6.2	2
108	17.63	15.86	0.88	6.033	3.573	3.747	5.329	2
109	19.94	16.92	0.8752	6.675	3.763	3.252	6.55	2
110	18.55	16.22	0.8865	6.153	3.674	1.738	5.894	2
111	18.45	16.12	0.8921	6.107	3.769	2.235	5.794	2
112	19.38	16.72	0.8716	6.303	3.791	3.678	5.965	2
113	19.13	16.31	0.9035	6.183	3.902	2.109	5.324	2
114	19.14	16.61	0.8722	6.259	3.737	6.682	6.053	2
115	20.97	17.25	0.8859	6.563	3.991	4.677	6.316	2
116	19.06	16.45	0.8854	6.416	3.719	2.248	6.163	2
117	18.96	16.2	0.9077	6.051	3.897	4.334	5.75	2
118	19.15	16.45	0.889	6.245	3.815	3.084	6.185	2
119	18.89	16.23	0.9008	6.227	3.769	3.639	5.966	2
120	20.03	16.9	0.8811	6.493	3.857	3.063	6.32	2
121	20.24	16.91	0.8897	6.315	3.962	5.901	6.188	2
122	18.14	16.12	0.8772	6.059	3.563	3.619	6.011	2
123	16.17	15.38	0.8588	5.762	3.387	4.286	5.703	2
124	18.43	15.97	0.9077	5.98	3.771	2.984	5.905	2
125	15.99	14.89	0.9064	5.363	3.582	3.336	5.144	2
126	18.75	16.18	0.8999	6.111	3.869	4.188	5.992	2
127	18.65	16.41	0.8698	6.285	3.594	4.391	6.102	2
128	17.98	15.85	0.8993	5.979	3.687	2.257	5.919	2
129	20.16	17.03	0.8735	6.513	3.773	1.91	6.185	2
130	17.55	15.66	0.8991	5.791	3.69	5.366	5.661	2
131	18.3	15.89	0.9108	5.979	3.755	2.837	5.962	2
132	18.94	16.32	0.8942	6.144	3.825	2.908	5.949	2
133	15.38	14.9	0.8706	5.884	3.268	4.462	5.795	2
134	16.16	15.33	0.8644	5.845	3.395	4.266	5.795	2
135	15.56	14.89	0.8823	5.776	3.408	4.972	5.847	2
136	15.38	14.66	0.899	5.477	3.465	3.6	5.439	2
137	17.36	15.76	0.8785	6.145	3.574	3.526	5.971	2
138	15.57	15.15	0.8527	5.92	3.231	2.64	5.879	2
139	15.6	15.11	0.858	5.832	3.286	2.725	5.752	2
140	16.23	15.18	0.885	5.872	3.472	3.769	5.322	2
141	13.07	13.92	0.848	5.472	2.994	5.304	5.395	3
142	13.32	13.94	0.8613	5.541	3.073	7.035	5.44	3
143	13.34	13.95	0.862	5.389	3.074	5.995	5.307	3
144	12.22	13.32	0.8652	5.224	2.967	5.469	5.221	3
145	11.82	13.4	0.8274	5.314	2.777	4.471	5.178	3
146	11.21	13.13	0.8167	5.279	2.687	6.169	5.275	3
147	11.43	13.13	0.8335	5.176	2.719	2.221	5.132	3
148	12.49	13.46	0.8658	5.267	2.967	4.421	5.002	3
149	12.7	13.71	0.8491	5.386	2.911	3.26	5.316	3
150	10.79	12.93	0.8107	5.317	2.648	5.462	5.194	3
151	11.83	13.23	0.8496	5.263	2.84	5.195	5.307	3
152	12.01	13.52	0.8249	5.405	2.776	6.992	5.27	3
153	12.26	13.6	0.8333	5.408	2.833	4.756	5.36	3
154	11.18	13.04	0.8266	5.22	2.693	3.332	5.001	3
155	11.36	13.05	0.8382	5.175	2.755	4.048	5.263	3
156	11.19	13.05	0.8253	5.25	2.675	5.813	5.219	3
157	11.34	12.87	0.8596	5.053	2.849	3.347	5.003	3
158	12.13	13.73	0.8081	5.394	2.745	4.825	5.22	3
159	11.75	13.52	0.8082	5.444	2.678	4.378	5.31	3
160	11.49	13.22	0.8263	5.304	2.695	5.388	5.31	3
161	12.54	13.67	0.8425	5.451	2.879	3.082	5.491	3
162	12.02	13.33	0.8503	5.35	2.81	4.271	5.308	3
163	12.05	13.41	0.8416	5.267	2.847	4.988	5.046	3
164	12.55	13.57	0.8558	5.333	2.968	4.419	5.176	3
165	11.14	12.79	0.8558	5.011	2.794	6.388	5.049	3
166	12.1	13.15	0.8793	5.105	2.941	2.201	5.056	3
167	12.44	13.59	0.8462	5.319	2.897	4.924	5.27	3
168	12.15	13.45	0.8443	5.417	2.837	3.638	5.338	3
169	11.35	13.12	0.8291	5.176	2.668	4.337	5.132	3
170	11.24	13	0.8359	5.09	2.715	3.521	5.088	3
171	11.02	13	0.8189	5.325	2.701	6.735	5.163	3
172	11.55	13.1	0.8455	5.167	2.845	6.715	4.956	3
173	11.27	12.97	0.8419	5.088	2.763	4.309	5	3
174	11.4	13.08	0.8375	5.136	2.763	5.588	5.089	3
175	10.83	12.96	0.8099	5.278	2.641	5.182	5.185	3
176	10.8	12.57	0.859	4.981	2.821	4.773	5.063	3
177	11.26	13.01	0.8355	5.186	2.71	5.335	5.092	3
178	10.74	12.73	0.8329	5.145	2.642	4.702	4.963	3
179	11.48	13.05	0.8473	5.18	2.758	5.876	5.002	3
180	12.21	13.47	0.8453	5.357	2.893	1.661	5.178	3
181	11.41	12.95	0.856	5.09	2.775	4.957	4.825	3
182	12.46	13.41	0.8706	5.236	3.017	4.987	5.147	3
183	12.19	13.36	0.8579	5.24	2.909	4.857	5.158	3
184	11.65	13.07	0.8575	5.108	2.85	5.209	5.195	3
185	12.89	13.77	0.8541	5.495	3.026	6.185	5.316	3
186	11.56	13.31	0.8198	5.363	2.683	4.062	5.182	3

187	11.81	13.45	0.8198	5.413	2.716	4.898	5.352	3
188	10.91	12.8	0.8372	5.088	2.675	4.179	4.956	3
189	11.23	12.82	0.8594	5.089	2.821	7.524	4.957	3
190	10.59	12.41	0.8648	4.899	2.787	4.975	4.794	3
191	10.93	12.8	0.839	5.046	2.717	5.398	5.045	3
192	11.27	12.86	0.8563	5.091	2.804	3.985	5.001	3
193	11.87	13.02	0.8795	5.132	2.953	3.597	5.132	3
194	10.82	12.83	0.8256	5.18	2.63	4.853	5.089	3
195	12.11	13.27	0.8639	5.236	2.975	4.132	5.012	3
196	12.8	13.47	0.886	5.16	3.126	4.873	4.914	3
197	12.79	13.53	0.8786	5.224	3.054	5.483	4.958	3
198	13.37	13.78	0.8849	5.32	3.128	4.67	5.091	3
199	12.62	13.67	0.8481	5.41	2.911	3.306	5.231	3
200	12.76	13.38	0.8964	5.073	3.155	2.828	4.83	3
201	12.38	13.44	0.8609	5.219	2.989	5.472	5.045	3
202	12.67	13.32	0.8977	4.984	3.135	2.3	4.745	3
203	11.18	12.72	0.868	5.009	2.81	4.051	4.828	3
204	12.7	13.41	0.8874	5.183	3.091	8.456	5	3
205	12.37	13.47	0.8567	5.204	2.96	3.919	5.001	3
206	12.19	13.2	0.8783	5.137	2.981	3.631	4.87	3
207	11.23	12.88	0.8511	5.14	2.795	4.325	5.003	3
208	13.2	13.66	0.8883	5.236	3.232	8.315	5.056	3
209	11.84	13.21	0.8521	5.175	2.836	3.598	5.044	3
210	12.3	13.34	0.8684	5.243	2.974	5.637	5.063	3

4.2 seeds.csv

	A	B	C	D	E	F
1	13.11	12351	11.16	68.8	1113	1
2	12.26	25879	4.02	69.1	1130	1
3	12.77	9271	15.7	63.4	1171	1
4	11.42	8865	9.11	56.8	1175	1
5	14.62	8403	11.68	73.5	2111	1
6	15.64	11030	5.13	77.6	2113	1
7	15.09	8258	25.65	72.6	2133	1
8	15.44	14163	2.69	78.1	2141	1
9	14.52	11377	1.03	73.1	2143	1
10	14.64	11023	0.94	68.8	2153	1
11	12.39	5902	1.91	62	2161	1
12	12.3	7059	7.83	60	2163	1
13	13.83	8425	15.33	53.8	2183	1
14	14.44	8049	57.31	62.2	2311	1
15	14.36	7405	48.28	74.9	2315	1
16	14.21	6336	54.77	55.1	2331	1
17	15.77	19263	5.13	82.3	2343	1
18	14.15	6112	77.1	58.1	2351	1
19	15.22	9593	34.89	58.3	2391	1
20	14.5	4686	4.14	72.8	2511	1
21	15.97	12480	19.59	84.6	2711	1
22	13.62	5648	83.78	59.6	2731	1
23	15.08	8034	46.8	66.1	2733	1
24	15.96	25308	10.56	87.2	3111	1
25	15.94	14558	4.32	66.7	3115	1
26	14.71	17498	6.91	68.4	3117	1
27	12.46	4614	96.12	64.7	3131	1
28	9.45	3485	76.14	34.9	3135	3
29	13.62	5092	82.66	72.1	3137	1
30	15.21	10432	24.71	69.3	3151	1
31	12.79	5180	76.04	67.5	3156	2
32	11.09	6197	21.03	57.2	3314	1
33	12.71	7562	11.15	57.6	3337	2

35	11.49	3148	95.97	41.9	4113	2
36	11.32	4348	68.24	49.4	4131	2
37	10.64	2448	91.76	42.3	4133	2
38	11.36	4330	75.92	47.7	4143	2
39	9.17	4761	11.37	30.9	4153	2
40	12.09	3016	83.19	32.7	4161	2
41	11.04	2901	92.86	38.7	4171	2
42	9.22	5511	7.62	36.1	4172	2
43	10.07	3739	52.27	37.2	4173	2
44	10.51	3161	96.14	38.1	4175	2
45	11.2	4741	47.06	29.4	4191	2
46	11.13	5052	56.1	51.1	4192	2
47	11.43	6259	39.17	35.7	4193	2
48	11	4075	63.23	35.6	4197	2
49	9.84	7482	17.04	41.5	5130	2
50	11.13	8780	3.16	40.2	5133	2
51	10.05	2594	67.82	26.5	5137	2
52	9.93	2370	3.69	23.3	5145	3
53	11.6	8131	13.09	47.3	5171	2
54	11.09	6992	24.44	47.1	5172	2
55	11.03	7956	23.88	51.1	5191	2
56	9.47	8895	0	43.5	6111	3
57	10.93	8891	1.65	51.6	6112	3
58	7.74	3116	52	29.7	6121	3
59	8.5	3930	15.51	20.2	6123	3
60	10.57	7869	6.01	54.9	6141	3
61	7.33	3000	69.31	20.8	6162	3
62	7.11	3472	33.57	17.3	6191	3
63	7.58	3582	30.08	20.1	6193	3
64	8.6	1656	27.75	21.5	7182	3
65	8.88	6860	0	35.3	7711	3
66	7.54	4199	33.3	38.9	8213	3
67	7.64	5134	17.26	25.2	8215	3
68	7.64	5134	17.26	34.8	8215	3
69	7.42	1890	72.24	23.2	8221	3
70	6.69	4443	31.36	33.3	8267	3
71	6.74	3485	39.48	28.8	8278	3
72	10.09	8043	1.5	42.5	8311	3
73	8.81	6686	4.28	44.2	8313	3
74	8.4	6565	2.3	35.9	8333	3
75	7.92	6477	5.17	41.8	8335	3
76	8.43	5811	13.62	35.9	8513	3
77	8.78	6573	5.78	43.7	8515	3
78	8.76	3942	74.54	50.8	8534	3
79	10.29	5449	2.92	37.2	8537	3
80	6.38	2847	90.67	28.2	8563	3
81	8.1	5795	0.81	38.1	8581	3
82	10.1	7716	0.78	50.3	8582	3
83	6.67	4696	0	27.3	8715	3
84	9.05	8316	1.34	40.9	8731	3
85	9.93	7147	0.99	50.2	8733	3
86	8.24	8880	0.65	51.1	8780	3
87	6.92	5299	0.56	38.9	8781	3
88	6.6	5959	0.52	36.2	8782	3
89	7.81	4549	2.46	29.9	8785	3
90	8.33	6928	0.61	42.9	8791	3
91	7.52	3910	1.09	26.5	8798	3
92	12.27	14032	0.58	66.1	9111	1
93	8.49	8845	0	48.9	9131	3
94	7.58	5562	9.47	35.9	9171	3
95	7.93	4224	3.59	25.1	9173	3
96	8.37	4753	0	26.1	9313	3
97	10	6462	13.58	42.2	9511	3
98	8.55	3617	70.87	35.2	9517	3

4.3 prestige.csv

4.2 Κατηγορία GaussianNode

Στην Κατηγορία GaussianNode ορίζονται 2 μεταβλητές πεδία της κλάσης. Το ένα πεδίο είναι ένας πίνακας τύπου float με τα κέντρα και το άλλο πεδίο είναι μια στατική μεταβλητή τύπου float. Στην κλάση υπάρχουν τρεις συναρτήσεις και ο κατασκευαστής (constructor) χωρίς ορίσματα. Η συνάρτηση τύπου float output με όρισμα έναν πίνακα τύπου float, η τύπου void συνάρτηση set_centers με όρισμα έναν πίνακα τύπου float και η στατική συνάρτηση gaussian_function τύπου float με όρισμα μια τοπική μεταβλητή τύπου double.


```

83 class GaussianNode {
84     float[] centers;
85     protected static float C;
86
87     public GaussianNode() { }
88
89     public float output(float[] raw) {...}
96
97     public void set_centers(float[] centers) { this.centers = centers; }
100
101     private static float gaussian_function(double in) { return (float)Math.exp(- (in * in) / (2 * C * C)); }
104 }

```

4.4 Κατηγορία GaussianNode.

```

87     public GaussianNode() { }
88
89     public float output(float[] raw) {
90
91         // Gaussify the distance between input layer and the gauss centers
92         return gaussian_function(
93             RBFClassifier.euclidean_distance(raw, centers)
94         );
95     }
96
97     public void set_centers(float[] centers) {
98         this.centers = centers;
99     }
100
101     private static float gaussian_function(double in) {
102         return (float)Math.exp(- (in * in) / (2 * C * C));
103     }

```

4.5 Κατασκευαστής και Συναρτήσεις Κατηγορίας GaussianNode.

4.3 Κατηγορία OutputNode

Στην Κατηγορία OutputNode ορίζονται δύο πίνακες ίδιου τύπου float με ονομασία weights και last_input. Η κλάση OutputNode εμπεριέχει έναν κατασκευαστή constructor με όρισμα μια τοπική μεταβλητή τύπου int και μία συνάρτηση output τύπου float, με όρισμα έναν πίνακα τύπου float.

```

106 class OutputNode {
107     protected float[] weights;
108     protected float[] last_input;
109
110     public OutputNode(int num_inputs) {...}
118
119     @ public float output(float[] goutput) {...}
131 }

```

4.6 Κατηγορία OutputNode.

```

110     public OutputNode(int num_inputs) {
111         weights = new float[num_inputs];
112         for (int i = 0; i < num_inputs; i++) {
113
114             // Keep the weights between -5 and 5.
115             weights[i] = (float)(Math.random() - .5);
116         }
117     }
118
119     @ public float output(float[] goutput) {
120         last_input = goutput;
121
122         float sum = 0f;
123
124         // Dot product weights*goutput
125         for (int i = 0; i < goutput.length; i++) {
126             sum += weights[i] * goutput[i];
127         }
128
129         return sum;
130     }

```

4.7 Κατασκευαστής και Συνάρτηση Κατηγορίας OutputNode.

4.4 Κατηγορία RBFNetwork

Στην Κατηγορία RBFNetwork ορίζονται τρία πεδία. Ένας πίνακας gnodes τύπου GaussianNode, ένας πίνακας onodes τύπου OutputNode και ένα πεδίο learning_rate τύπου float. Επίσης, η κλάση εμπεριέχει δύο συναρτήσεις και κατασκευαστή με ορίσματα. Ο κατασκευαστής έχει σαν ορίσματα έναν πίνακα num_inputs τύπου float, δύο τοπικές μεταβλητές τύπου int με ονόματα num_gaussian και num_outputs και ακόμη δύο τοπικές μεταβλητές τύπου float με ονόματα learning_rate και gaussian_width. Η μία συνάρτηση get_output είναι τύπου πίνακα float με όρισμα έναν πίνακα input τύπου float και η άλλη συνάρτηση back_propagate είναι τύπου void με ορίσματα δύο πίνακες τύπου float με ονόματα expected και outcome.

```

3     public class RBFNetwork {
4         protected GaussianNode[] gnodes;
5         protected OutputNode[] onodes;
6         protected float learning_rate;
7
8         /** Constructs an actual Radial-Basis Function network ...*/
14        public RBFNetwork(float[] num_inputs, int num_gaussian, int num_outputs, float learning_rate, float gaussian_width) {...}
33
34        /** Gets the trained outputs of the network given input ...*/
39        public float[] get_output(float[] input) {...}
55
56        /** Updates the weights of the hidden layer (the heights of the ...*/
62        public void back_propagate(float[] expected, float[] outcome) {...}
81    }

```

4.8 Κατηγορία RBFNetwork.

```
14 public RBFNetwork(float[] num_inputs, int num_gaussian, int num_outputs, float learning_rate, float gaussian_width) {
15     this.learning_rate = learning_rate;
16     GaussianNode.C = gaussian_width;
17
18     // Save the kmeans data to initialize the gnodes
19     gnodes = new GaussianNode[num_gaussian];
20     for (int i = 0; i < num_gaussian; i++) {
21         // Inputs = input space
22         gnodes[i] = new GaussianNode();
23     }
24
25     // Initialize the output nodes (random weights will be updated)
26     onodes = new OutputNode[num_outputs];
27     for (int i = 0; i < num_outputs; i++) {
28
29         // Inputs = number of hidden nodes
30         onodes[i] = new OutputNode(num_gaussian);
31     }
32 }
```

4.9 Κατασκευαστής Κατηγορίας RBFNetwork.

```
39 public float[] get_output(float[] input) {
40
41     // Feed input layer into gaussian layer
42     float gaussian[] = new float[gnodes.length];
43     for (int i = 0; i < gaussian.length; i++) {
44         gaussian[i] = gnodes[i].output(input);
45     }
46
47     // Feed gaussian layer into output layer
48     float output[] = new float[onodes.length];
49     for (int i = 0; i < output.length; i++) {
50         output[i] = onodes[i].output(gaussian);
51     }
52
53     return output;
54 }
```

4.10 Συνάρτηση get_output της Κατηγορίας RBFNetwork.

```
62 public void back_propogate(float[] expected, float[] outcome) {
63
64     // Calculate the error for each node of the output layer
65     for (int i = 0; i < onodes.length; i++) {
66
67         // Get the error of this classification
68         float error = expected[i] - outcome[i];
69
70         // Update the weights
71         for (int j = 0; j < onodes[i].weights.length; j++){
72
73             // The weight update function
74             onodes[i].weights[j] +=
75                 learning_rate *
76                 error *
77                 onodes[i].last_input[j]; // * onodes[i].weights[j];
78         }
79     }
80 }
```

4.5 Κατηγορία `RBFClassifier`

Η Κατηγορία `RBFClassifier` αποτελείται από μία τοπική μεταβλήτη `network` τύπου `RBFNetwork`, έναν κατασκευαστή με ορίσματα και έξι συναρτήσεις. Ο κατασκευαστής έχει ως πρώτο όρισμα έναν πίνακα `num_inputs` τύπου `float`, δύο τοπικές μεταβλητές τύπου `int` με ονόματα `num_gaussian` και `num_outputs` και ακόμη δύο τοπικές μεταβλητές τύπου `float` με ονόματα `learning_rate` και `gaussian_width`. Επίσης, έχει μία συνάρτηση `classify` τύπου `int` με όρισμα έναν πίνακα `input` τύπου `float`. Η δεύτερη συνάρτηση στη σειρά είναι η `set_weights_w` τύπου `void` με ορίσματα δύο τοπικές μεταβλητές που η μία είναι τύπου `String` και η άλλη τύπου `int` με ονόματα `filename` και `num_gaussian`, καθώς και έναν πίνακα `data` τύπου `float`. Η τρίτη κατά σειρά συνάρτηση είναι η `set_weights_r` τύπου `void` με όρισμα `filename` τύπου `String` ως τοπική μεταβλητή. Η τέταρτη συνάρτηση είναι η `kmeans` τύπου `float` δισδιάστατου πίνακα, με ορίσματα έναν δισδιάστατο πίνακα τύπου `float` και μια τοπική μεταβλητή τύπου `int` την `num_gaussian`. Τέλος, η συνάρτηση `euclidean_distance` τύπου `double` με ορίσματα δύο πίνακες με ονόματα `a` και `b` τύπου `float`.

```

5   public class RBFClassifier {
6
7       protected RBFNetwork network;
8
9       /** Builds a manager for a Radial-Basis Function network which will classify ...*/
16      public RBFClassifier(float[] num_inputs, int num_gaussian, int num_output, float learning_rate, float gaussian_width) {...}
19
20      /** Classifies (tests) the input ...*/
25      public int classify(float[] input) {...}
41
42
43      public void learn(float[] input, float[] outcome) {...}
57
58      /** Sets the weights of the hidden layer based on the data. Also writes ...*/
65      public void set_weights_w(String filename, float[][] data, int num_gaussian) {...}
77
78      /** Sets the weights of the hidden layer based on the data. Also writes ...*/
83      public void set_weights_r(String filename) {...}
91
92      /** Applies k-means clustering on a data file into a given number of ...*/
180 @ private static float[][] kmeans(float[][] data, int num_gaussian) {...}
174
175      /** Gets the Euclidean distance between two vectors ...*/
181 @ public static double euclidean_distance(float[] a, float[] b) {...}
198 }

```

4.12 Κατηγορία `RBFClassifier`.

```

16      public RBFClassifier(float[] num_inputs, int num_gaussian, int num_output, float learning_rate, float gaussian_width) {
17          network = new RBFNetwork(num_inputs, num_gaussian, num_output, learning_rate, gaussian_width);
18      }
19
20      /** Classifies (tests) the input ...*/
25      public int classify(float[] input) {
26
27          // First read from the the RBF network
28          float[] class_vals = network.get_output(input);
29
30          // Find the argmax value (the most likely category)
31          int best = 0;
32          for (int i = 0; i < class_vals.length; i++) {
33              if (class_vals[best] < class_vals[i]) {
34                  best = i;
35              }
36          }
37      }
38
39      return best;
48 }

```

4.13 Κατασκευαστής και Συνάρτηση classify της Κατηγορίας RBFClassifier.

```
43 public void learn(float[] input, float[] outcome) {
44     float[] out = network.get_output(input);
45
46
47     // Learning requires "backpropagation"
48     // swapping these parameters fixed the world somehow.
49     network.back_propogate(
50
51         // The expected is what the network thinks will happen
52         outcome,
53
54         // The actual is given based on the data
55         out);
56 }
```

4.14 Συνάρτηση learn της Κατηγορίας RBFClassifier.

```
65 public void set_weights_w(String filename, float[][] data, int num_gaussian) {
66
67     // make k-means clusters
68     float[][] kmeans = kmeans(data, num_gaussian);
69
70     Main.float_to_csv(filename, kmeans);
71
72     // Simply these clusters to the hidden layer
73     for (int i = 0; i < kmeans.length; i++) {
74         network.gnodes[i].set_centers(kmeans[i]);
75     }
76 }
77
78 /** Sets the weights of the hidden layer based on the data. Also writes ...*/
83 public void set_weights_r(String filename) {
84     // float[][] centers = Main.csv_to_float("centers.csv");
85     float[][] centers = Main.csv_to_float(filename);
86
87     for (int i = 0; i < centers.length; i++) {
88         network.gnodes[i].set_centers(centers[i]);
89     }
90 }
```

4.15 Συναρτήσεις set_weights_w και set_weights_w της Κατηγορίας RBFClassifier.

```

100 @ private static float[][] kmeans(float[][] data, int num_gaussian) {
101
102     // Make the list of centers- num_gaussian lists, each at the
103     // length of the network input
104     float[][] centers = new float[num_gaussian][data[0].length];
105
106     // Initialize the first centers- force the initial clusters
107     System.arraycopy(data, srcPos: 0, centers, destPos: 0, centers.length);
108
109     // Prepare the loop
110     HashMap<float[], Integer> old_assignments = null;
111     boolean changed = true;
112
113     // Run the loop. Terminate when there is no change because
114     // this function is proven to terminate without requiring
115     // an infinite convergence.

```

4.16 Αρχή της Συνάρτησης kmeans της Κατηγορίας RBFClassifier.

```

116     while (changed) {
117
118         changed = false;
119         HashMap<float[], Integer> assignments = new HashMap<>();
120         float[][] new_centers = new float[num_gaussian][data[0].length];
121
122         // Assignments
123         int[] center_count = new int[num_gaussian];
124         for (float[] f : data) {
125
126             // Find the closest center
127             int min_index = 0;
128             double min_dis = euclidean_distance(centers[0], f);
129             for (int i = 1; i < centers.length; i++) {
130                 double distance = euclidean_distance(centers[i], f);
131                 if (distance < min_dis) {
132                     min_dis = distance;
133                     min_index = i;
134                 }
135             }
136
137             // Save the this data point's closest center
138             assignments.put(f, min_index);
139
140             // Did the point change closest cluster since last time?
141             if (old_assignments == null || old_assignments.get(f) != min_index) {
142                 changed = true;
143             }
144
145             // Make up new centers
146             for (int i = 0; i < f.length; i++) {
147                 new_centers[min_index][i] += f[i];
148             }
149
150             // Count the number in this center (good for average later)
151             center_count[min_index]++;
152
153         }

```

4.17 Μέση της Συνάρτησης kmeans της Κατηγορίας RBFClassifier.

```
156     for (int i = 0; i < num_gaussian; i++) {
157         for (int j = 0; j < new_centers[i].length; j++) {
158
159             // Average the centers
160             if (center_count[i] != 0) {
161                 new_centers[i][j] /= center_count[i];
162             }
163         }
164     }
165
166     // store this iteration for next time
167     old_assignments = assignments;
168     centers = new_centers;
169 }
170
171 // Return the best
172 return centers;
173 }
```

4.18 Τέλος της Συνάρτησης kmeans της Κατηγορίας RBFClassifier.

```
181 @ public static double euclidean_distance(float[] a, float[] b) {
182     double sum = 0.0;
183     for (int i = 0; i < a.length; i++) {
184         if(a[i]!=0f && b[i]!=0f) {
185             sum += Math.pow(a[i] - b[i], 2);
186         }
187     }
188     return Math.sqrt(sum);
189 }
```

4.19 Συνάρτηση euclidean_distance της Κατηγορίας RBFClassifier.

4.6 Κατηγορία Main

Η Κατηγορία Main περιέχει τρεις στατικές συναρτήσεις. Η μία συνάρτηση `csv_to_float` είναι τύπου διδιάστατου πίνακα με όρισμα μια τοπική μεταβλητή `filename` τύπου `String`, η άλλη συνάρτηση `float_to_csv` τύπου `void` με όρισμα μια τοπική μεταβλητή `filename` τύπου `String` και έναν διδιάστατο πίνακα με όνομα `data`. Τέλος, η κύρια συνάρτηση με όνομα `main` τύπου `void` με όρισμα ένα πίνακα τύπου `String` με όνομα `args`.

```
9 ▶ public class Main {
10 ▶     public static void main(String[] args) {...}
105
106 @ ▶     public static float[][] csv_to_float(String filename) {...}
132
133 @ ▶     public static void float_to_csv(String filename, float[][] data) {...}
162 }
```

4.20 Κατηγορίας main.

4.21

```
106 @ ▶     public static float[][] csv_to_float(String filename) {
107         BufferedReader file = null;
108         String raw = "";
109         try {
110             file = new BufferedReader(new FileReader(filename));
111             while (file.ready()) {
112                 raw += file.readLine() + "\n";
113             }
114             file.close();
115         }
116         catch (IOException e) {
117             return null;
118         }
119
120         String[] lines = raw.split( regex: "\n");
121         float[][] data = new float[lines.length][];
122         for (int i = 0; i < lines.length; i++) {
123             String[] line = lines[i].split( regex: "");
124             data[i] = new float[line.length];
125             for (int j = 0; j < line.length; j++) {
126                 data[i][j] = Float.parseFloat(line[j]);
127             }
128         }
129
130         return data;
131     }
```

Συνάρτηση `csv_to_float` της Κατηγορίας main.


```

133 @ public static void float_to_csv(String filename, float[][] data) {
134     BufferedWriter writer = null;
135     try {
136         writer = new BufferedWriter(new OutputStreamWriter(
137             new FileOutputStream(filename), "utf-8"));
138
139         // Clear the file
140         writer.write("");
141         for (int i = 0; i < data.length; i++){
142
143             // Dirty implode (Java so lame)
144             String line = "";
145             for (int j = 0; j < data[i].length; j++) {
146                 line += data[i][j] + ",";
147             }
148             writer.append(line.substring(0, line.length() - 1));
149             if (i < data.length - 1) {
150                 writer.append("\n");
151             }
152         }
153     }
154     catch (IOException ex) {}
155     finally {
156         try {
157             writer.close();
158         }
159         catch (Exception ex) {}
160     }
161 }

```

4.22 Συνάρτηση float_to_csv της Κατηγορίας main.

```

10 public static void main(String[] args) {
11     JFrame f;
12     f=new JFrame();
13     f.setSize( width: 300, height: 300);
14     f.setLayout(null);
15     f.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
16     int num_input=Integer.parseInt(JOptionPane.showInputDialog(f, "message: "Enter number of inputs"));
17     int num_gaussian=Integer.parseInt(JOptionPane.showInputDialog(f, "message: "Give number of centers/hidden layers."));
18     int num_output=Integer.parseInt(JOptionPane.showInputDialog(f, "Enter number of output"));
19     //
20     //
21     float learning_rate=Float.parseFloat(JOptionPane.showInputDialog(f, "Give Learning Rate"));
22     float gaussian_width=Float.parseFloat(JOptionPane.showInputDialog(f, "Give Gaussian width"));
23     String input_filename=JOptionPane.showInputDialog(f, "message: "Enter input filename");
24     String output_filename=JOptionPane.showInputDialog(f, "Enter output filename");
25     //
26     //
27     int num_input = 2;
28     //
29     //
30     int num_gaussian = 50; // number of centers // hidden layers
31     //
32     //
33     int num_output = 4;
34     //
35     //
36     float learning_rate = 0.1f;
37     //
38     //
39     float gaussian_width = 0.02f;
40
41     //Implementation until 26_10
42     LocalDate localDate = LocalDate.now();
43     LocalTime localTime = LocalTime.now();
44     String line="";
45     String csvSplitBy=",";
46     List<Integer> target=new ArrayList<Integer>();
47     Integer counter=0;
48     ArrayList<ArrayList> file_input=new ArrayList<>();

```

```

39     int k=0;
40     try(BufferedReader br=new BufferedReader(new FileReader(input_filename))){
41         while ((line=br.readLine())!=null) {
42             String[] data = line.split(csvSplitBy);
43             for (int i = 0; i < num_input; i++) {
44                 if(i==num_input-1)
45                     target.add(Integer.parseInt(data[i]));
46             }
47         }
48     }catch (Exception e){
49         e.printStackTrace();
50     }

```

```

51     try(BufferedReader br=new BufferedReader(new FileReader(input_filename))){
52         BufferedWriter writer = new BufferedWriter(new FileWriter("classified_data.csv"));
53         writer.write("Classified Data for" + " " + input_filename + " " +localDate+ " " +localTime);
54         writer.newLine();
55         while ((line=br.readLine())!=null) {
56             String[] data = line.split(csvSplitBy);
57             ArrayList<Float> input=new ArrayList<>();
58
59             for (int i = 0; i < num_input - 1; i++) {
60                 input.add(Float.parseFloat(data[i]));
61             }
62             System.out.println("input is : " + input);
63             file_input.add(input);
64
65             Float[] sample2=new Float[input.size()];
66             input.toArray(sample2);
67             float[] sample= new float[sample2.length];
68             for (int i = 0; i < sample2.length; i++) {
69                 sample[i] = Float.parseFloat(sample2[i].toString());
70             }
71
72             RBFClassifier classifier = new RBFClassifier(sample, num_gaussian, num_output, 3, learning_rate: 0.1f, gaussian_width: 0.02f);
73             classifier.set_weights_r(input_filename);
74
75             float[] actual=new float[target.size()];
76             for(int h=0;h< target.size();h++){
77                 actual[h]=target.get(h);
78             }
79
80             classifier.learn(sample,actual);
81
82             int output = classifier.classify(sample);
83             if (output == 0) {
84                 output = (int)actual[k];
85             } else if (output == 1) {
86                 output = (int)actual[k];
87             } else {
88                 output = (int)actual[k];
89             }
90             if(output==target.get(k)){
91                 counter++;
92             }
93             System.out.println("Classified ( " + input + " ) as " + output + ", Actual was " + target.get(k));
94
95             writer.write("Classified ( " + input + " ) as " + output + ", Actual was " + target.get(k));
96             writer.flush();
97             writer.newLine();
98             k++;
99         }
100     }catch (Exception e){
101         e.printStackTrace();
102     }
103     System.out.println("Σωστά ταξινομημένα είναι : " + counter + " και τα συνολικά είναι : " + num_gaussian + ".");
104 }

```

4.23 Κύρια Συνάρτηση της Κατηγορίας main.

4.7 Παράδειγμα εκτέλεσης και αποτελέσματα

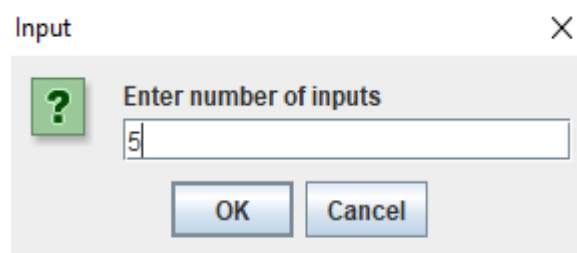
Εφόσον προσθέσουμε τον κώδικα σε ένα περιβάλλον ανάπτυξης λογισμικού για Java (Eclipse, IntelliJ) και πατήσουμε για εκτέλεση του κώδικα.

Βήμα 1) Θα μας ζητήσει να πληκτρολογήσουμε τον αριθμό των εισόδων συμπεριλαμβανομένης της εξόδου (Εικόνα 4.24) . Ο αριθμός που θα εισάγουμε πρέπει να συμπίπτει με τον αριθμό εισόδων συμπεριλαμβανομένης της εξόδου που υπάρχουν στο αρχείο. Στο παράδειγμα αυτό ο αριθμός είναι 5.

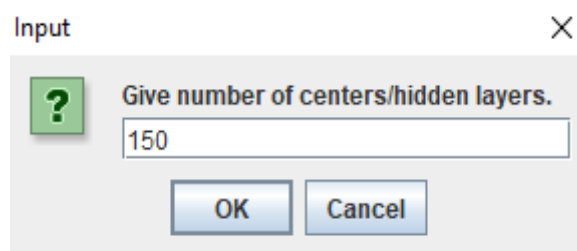
Βήμα 2) Θα μας ζητήσει να πληκτρολογήσουμε τα κέντρα ή αλλιώς τους κόμβους του κρυφού επιπέδου (Εικόνα 4.25). Ο αριθμός που θα εισάγουμε είναι το πλήθος όλων των κέντρων που έχει το αρχείο, δηλαδή το πλήθος των γραμμών. Στο παράδειγμα αυτό ο αριθμός είναι 150.

Βήμα 3) Θα μας ζητήσει να πληκτρολογήσουμε το όνομα του αρχείου που θέλουμε ώστε να εκτελεστεί το πρόγραμμα (Εικόνα 4.26). Το όνομα του αρχείου που θα επιλέξουμε πρέπει να συμπίπτει με τα προηγούμενα δεδομένα και να συμπεριλαμβάνει την κατάληξη του. Στο παράδειγμα αυτό θα είναι το αρχείο iris.csv.

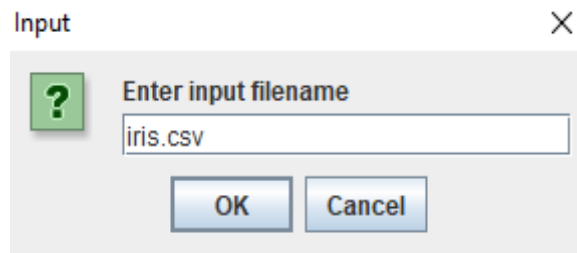
Τέλος εκτελείται το πρόγραμμα και εμφανίζει τα αποτελέσματα στην γραμμή εντολών (Εικόνα 4.27) μαζί με το πλήθος των σωστά ταξινομημένων προτύπων. Αυτά αποθηκεύονται σε αρχείο με όνομα classified_data.csv μαζί με το όνομα του αρχείου που εισάχθηκε, την ημερομηνία και την ώρα εκτέλεσης. (Εικόνα 4.28)



4.24 Εισαγωγή των συνολικών εισόδων.



4.25 Εισαγωγή των κέντρων/κρυφών κόμβων.



4.26 Εισαγωγή αρχείου.

Σωστά ταξινομημένα είναι 150 και τα συνολικά είναι 150.

4.27 Εμφάνιση πλήθους σωστά ταξινομημένων προτύπων.

```
1 Classified Data for iris.csv 2020-11-16 20:34:11.307
2 Classified ( [5.1, 3.5, 1.4, 0.2] ) as 1. Actual was 1
3 Classified ( [4.9, 3.0, 1.4, 0.2] ) as 1. Actual was 1
4 Classified ( [4.7, 3.2, 1.3, 0.2] ) as 1. Actual was 1
5 Classified ( [4.6, 3.1, 1.5, 0.2] ) as 1. Actual was 1
6 Classified ( [5.0, 3.6, 1.4, 0.2] ) as 1. Actual was 1
7 Classified ( [5.4, 3.9, 1.7, 0.4] ) as 1. Actual was 1
8 Classified ( [4.6, 3.4, 1.4, 0.3] ) as 1. Actual was 1
9 Classified ( [5.0, 3.4, 1.5, 0.2] ) as 1. Actual was 1
10 Classified ( [4.4, 2.9, 1.4, 0.2] ) as 1. Actual was 1
11 Classified ( [4.9, 3.1, 1.5, 0.1] ) as 1. Actual was 1
12 Classified ( [5.4, 3.7, 1.5, 0.2] ) as 1. Actual was 1
13 Classified ( [4.8, 3.4, 1.6, 0.2] ) as 1. Actual was 1
14 Classified ( [4.8, 3.0, 1.4, 0.1] ) as 1. Actual was 1
15 Classified ( [4.3, 3.0, 1.1, 0.1] ) as 1. Actual was 1
16 Classified ( [5.8, 4.0, 1.2, 0.2] ) as 1. Actual was 1
17 Classified ( [5.7, 4.4, 1.5, 0.4] ) as 1. Actual was 1
18 Classified ( [5.4, 3.9, 1.3, 0.4] ) as 1. Actual was 1
19 Classified ( [5.1, 3.5, 1.4, 0.3] ) as 1. Actual was 1
20 Classified ( [5.7, 3.8, 1.7, 0.3] ) as 1. Actual was 1
21 Classified ( [5.1, 3.8, 1.5, 0.3] ) as 1. Actual was 1
22 Classified ( [5.4, 3.4, 1.7, 0.2] ) as 1. Actual was 1
23 Classified ( [5.1, 3.7, 1.5, 0.4] ) as 1. Actual was 1
24 Classified ( [4.6, 3.6, 1.0, 0.2] ) as 1. Actual was 1
25 Classified ( [5.1, 3.3, 1.7, 0.5] ) as 1. Actual was 1
```

```
26 Classified ( [4.8, 3.4, 1.9, 0.2] ) as 1. Actual was 1
27 Classified ( [5.0, 3.0, 1.6, 0.2] ) as 1. Actual was 1
28 Classified ( [5.0, 3.4, 1.6, 0.4] ) as 1. Actual was 1
29 Classified ( [5.2, 3.5, 1.5, 0.2] ) as 1. Actual was 1
30 Classified ( [5.2, 3.4, 1.4, 0.2] ) as 1. Actual was 1
31 Classified ( [4.7, 3.2, 1.6, 0.2] ) as 1. Actual was 1
32 Classified ( [4.8, 3.1, 1.6, 0.2] ) as 1. Actual was 1
33 Classified ( [5.4, 3.4, 1.5, 0.4] ) as 1. Actual was 1
34 Classified ( [5.2, 4.1, 1.5, 0.1] ) as 1. Actual was 1
35 Classified ( [5.5, 4.2, 1.4, 0.2] ) as 1. Actual was 1
36 Classified ( [4.9, 3.1, 1.5, 0.2] ) as 1. Actual was 1
37 Classified ( [5.0, 3.2, 1.2, 0.2] ) as 1. Actual was 1
38 Classified ( [5.5, 3.5, 1.3, 0.2] ) as 1. Actual was 1
39 Classified ( [4.9, 3.6, 1.4, 0.1] ) as 1. Actual was 1
40 Classified ( [4.4, 3.0, 1.3, 0.2] ) as 1. Actual was 1
41 Classified ( [5.1, 3.4, 1.5, 0.2] ) as 1. Actual was 1
42 Classified ( [5.0, 3.5, 1.3, 0.3] ) as 1. Actual was 1
43 Classified ( [4.5, 2.3, 1.3, 0.3] ) as 1. Actual was 1
44 Classified ( [4.4, 3.2, 1.3, 0.2] ) as 1. Actual was 1
45 Classified ( [5.0, 3.5, 1.6, 0.6] ) as 1. Actual was 1
46 Classified ( [5.1, 3.8, 1.9, 0.4] ) as 1. Actual was 1
47 Classified ( [4.8, 3.0, 1.4, 0.3] ) as 1. Actual was 1
48 Classified ( [5.1, 3.8, 1.6, 0.2] ) as 1. Actual was 1
49 Classified ( [4.6, 3.2, 1.4, 0.2] ) as 1. Actual was 1
50 Classified ( [5.3, 3.7, 1.5, 0.2] ) as 1. Actual was 1
```

```
51 Classified ( [5.0, 3.3, 1.4, 0.2] ) as 1. Actual was 1
52 Classified ( [7.0, 3.2, 4.7, 1.4] ) as 2. Actual was 2
53 Classified ( [6.4, 3.2, 4.5, 1.5] ) as 2. Actual was 2
54 Classified ( [6.9, 3.1, 4.9, 1.5] ) as 2. Actual was 2
55 Classified ( [5.5, 2.3, 4.0, 1.3] ) as 2. Actual was 2
56 Classified ( [6.5, 2.8, 4.6, 1.5] ) as 2. Actual was 2
57 Classified ( [5.7, 2.8, 4.5, 1.3] ) as 2. Actual was 2
58 Classified ( [6.3, 3.3, 4.7, 1.6] ) as 2. Actual was 2
59 Classified ( [4.9, 2.4, 3.3, 1.0] ) as 2. Actual was 2
60 Classified ( [6.6, 2.9, 4.6, 1.3] ) as 2. Actual was 2
61 Classified ( [5.2, 2.7, 3.9, 1.4] ) as 2. Actual was 2
62 Classified ( [5.0, 2.0, 3.5, 1.0] ) as 2. Actual was 2
63 Classified ( [5.9, 3.0, 4.2, 1.5] ) as 2. Actual was 2
64 Classified ( [6.0, 2.2, 4.0, 1.0] ) as 2. Actual was 2
65 Classified ( [6.1, 2.9, 4.7, 1.4] ) as 2. Actual was 2
66 Classified ( [5.6, 2.9, 3.6, 1.3] ) as 2. Actual was 2
67 Classified ( [6.7, 3.1, 4.4, 1.4] ) as 2. Actual was 2
68 Classified ( [5.6, 3.0, 4.5, 1.5] ) as 2. Actual was 2
69 Classified ( [5.8, 2.7, 4.1, 1.0] ) as 2. Actual was 2
70 Classified ( [6.2, 2.2, 4.5, 1.5] ) as 2. Actual was 2
71 Classified ( [5.6, 2.5, 3.9, 1.1] ) as 2. Actual was 2
72 Classified ( [5.9, 3.2, 4.8, 1.8] ) as 2. Actual was 2
73 Classified ( [6.1, 2.8, 4.0, 1.3] ) as 2. Actual was 2
74 Classified ( [6.3, 2.5, 4.9, 1.5] ) as 2. Actual was 2
75 Classified ( [6.1, 2.8, 4.7, 1.2] ) as 2. Actual was 2
```

```
76 Classified ( [6.4, 2.9, 4.3, 1.3] ) as 2. Actual was 2
77 Classified ( [6.6, 3.0, 4.4, 1.4] ) as 2. Actual was 2
78 Classified ( [6.8, 2.8, 4.8, 1.4] ) as 2. Actual was 2
79 Classified ( [6.7, 3.0, 5.0, 1.7] ) as 2. Actual was 2
80 Classified ( [6.0, 2.9, 4.5, 1.5] ) as 2. Actual was 2
81 Classified ( [5.7, 2.6, 3.5, 1.0] ) as 2. Actual was 2
82 Classified ( [5.5, 2.4, 3.8, 1.1] ) as 2. Actual was 2
83 Classified ( [5.5, 2.4, 3.7, 1.0] ) as 2. Actual was 2
84 Classified ( [5.8, 2.7, 3.9, 1.2] ) as 2. Actual was 2
85 Classified ( [6.0, 2.7, 5.1, 1.6] ) as 2. Actual was 2
86 Classified ( [5.4, 3.0, 4.5, 1.5] ) as 2. Actual was 2
87 Classified ( [6.0, 3.4, 4.5, 1.6] ) as 2. Actual was 2
88 Classified ( [6.7, 3.1, 4.7, 1.5] ) as 2. Actual was 2
89 Classified ( [6.3, 2.3, 4.4, 1.3] ) as 2. Actual was 2
90 Classified ( [5.6, 3.0, 4.1, 1.3] ) as 2. Actual was 2
91 Classified ( [5.5, 2.5, 4.0, 1.3] ) as 2. Actual was 2
92 Classified ( [5.5, 2.6, 4.4, 1.2] ) as 2. Actual was 2
93 Classified ( [6.1, 3.0, 4.6, 1.4] ) as 2. Actual was 2
94 Classified ( [5.8, 2.6, 4.0, 1.2] ) as 2. Actual was 2
95 Classified ( [5.0, 2.3, 3.3, 1.0] ) as 2. Actual was 2
96 Classified ( [5.6, 2.7, 4.2, 1.3] ) as 2. Actual was 2
97 Classified ( [5.7, 3.0, 4.2, 1.2] ) as 2. Actual was 2
98 Classified ( [5.7, 2.9, 4.2, 1.3] ) as 2. Actual was 2
99 Classified ( [6.2, 2.9, 4.3, 1.3] ) as 2. Actual was 2
100 Classified ( [5.1, 2.5, 3.0, 1.1] ) as 2. Actual was 2
```

```
101 Classified ( [5.7, 2.8, 4.1, 1.3] ) as 2. Actual was 2
102 Classified ( [6.3, 3.3, 6.0, 2.5] ) as 3. Actual was 3
103 Classified ( [5.8, 2.7, 5.1, 1.9] ) as 3. Actual was 3
104 Classified ( [7.1, 3.0, 5.9, 2.1] ) as 3. Actual was 3
105 Classified ( [6.3, 2.9, 5.6, 1.8] ) as 3. Actual was 3
106 Classified ( [6.5, 3.0, 5.8, 2.2] ) as 3. Actual was 3
107 Classified ( [7.6, 3.0, 6.6, 2.1] ) as 3. Actual was 3
108 Classified ( [4.9, 2.5, 4.5, 1.7] ) as 3. Actual was 3
109 Classified ( [7.3, 2.9, 6.3, 1.8] ) as 3. Actual was 3
110 Classified ( [6.7, 2.5, 5.8, 1.8] ) as 3. Actual was 3
111 Classified ( [7.2, 3.6, 6.1, 2.5] ) as 3. Actual was 3
112 Classified ( [6.5, 3.2, 5.1, 2.0] ) as 3. Actual was 3
113 Classified ( [6.4, 2.7, 5.3, 1.9] ) as 3. Actual was 3
114 Classified ( [6.8, 3.0, 5.5, 2.1] ) as 3. Actual was 3
115 Classified ( [5.7, 2.5, 5.0, 2.0] ) as 3. Actual was 3
116 Classified ( [5.8, 2.8, 5.1, 2.4] ) as 3. Actual was 3
117 Classified ( [6.4, 3.2, 5.3, 2.3] ) as 3. Actual was 3
118 Classified ( [6.5, 3.0, 5.5, 1.8] ) as 3. Actual was 3
119 Classified ( [7.7, 3.8, 6.7, 2.2] ) as 3. Actual was 3
120 Classified ( [7.7, 2.6, 6.9, 2.3] ) as 3. Actual was 3
121 Classified ( [6.0, 2.2, 5.0, 1.5] ) as 3. Actual was 3
122 Classified ( [6.9, 3.2, 5.7, 2.3] ) as 3. Actual was 3
123 Classified ( [5.6, 2.8, 4.9, 2.0] ) as 3. Actual was 3
124 Classified ( [7.7, 2.8, 6.7, 2.0] ) as 3. Actual was 3
125 Classified ( [6.3, 2.7, 4.9, 1.8] ) as 3. Actual was 3
```

4.28 Αρχείο

```
126 Classified ( [6.7, 3.3, 5.7, 2.1] ) as 3. Actual was 3
127 Classified ( [7.2, 3.2, 6.0, 1.8] ) as 3. Actual was 3
128 Classified ( [6.2, 2.8, 4.8, 1.8] ) as 3. Actual was 3
129 Classified ( [6.1, 3.0, 4.9, 1.8] ) as 3. Actual was 3
130 Classified ( [6.4, 2.8, 5.6, 2.1] ) as 3. Actual was 3
131 Classified ( [7.2, 3.0, 5.8, 1.6] ) as 3. Actual was 3
132 Classified ( [7.4, 2.8, 6.1, 1.9] ) as 3. Actual was 3
133 Classified ( [7.9, 3.8, 6.4, 2.0] ) as 3. Actual was 3
134 Classified ( [6.4, 2.8, 5.6, 2.2] ) as 3. Actual was 3
135 Classified ( [6.3, 2.8, 5.1, 1.5] ) as 3. Actual was 3
136 Classified ( [6.1, 2.6, 5.6, 1.4] ) as 3. Actual was 3
137 Classified ( [7.7, 3.0, 6.1, 2.3] ) as 3. Actual was 3
138 Classified ( [6.3, 3.4, 5.6, 2.4] ) as 3. Actual was 3
139 Classified ( [6.4, 3.1, 5.5, 1.8] ) as 3. Actual was 3
140 Classified ( [6.0, 3.0, 4.8, 1.8] ) as 3. Actual was 3
141 Classified ( [6.9, 3.1, 5.4, 2.1] ) as 3. Actual was 3
142 Classified ( [6.7, 3.1, 5.6, 2.4] ) as 3. Actual was 3
143 Classified ( [6.9, 3.1, 5.1, 2.3] ) as 3. Actual was 3
144 Classified ( [5.8, 2.7, 5.1, 1.9] ) as 3. Actual was 3
145 Classified ( [6.8, 3.2, 5.9, 2.3] ) as 3. Actual was 3
146 Classified ( [6.7, 3.3, 5.7, 2.5] ) as 3. Actual was 3
147 Classified ( [6.7, 3.0, 5.2, 2.3] ) as 3. Actual was 3
148 Classified ( [6.3, 2.5, 5.0, 1.9] ) as 3. Actual was 3
149 Classified ( [6.5, 3.0, 5.2, 2.0] ) as 3. Actual was 3
150 Classified ( [6.2, 3.4, 5.4, 2.3] ) as 3. Actual was 3
151 Classified ( [5.9, 3.0, 5.1, 1.8] ) as 3. Actual was 3
```

αποθήκευσης αποτελεσμάτων.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Sidath Asiri, <<Machine Learning Classifiers,>> [Ηλεκτρονικό]. Available: <https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623> . [Πρόσβαση 11 2020].
- [2] G. Ou and Y. L. Murphey, «Multi-class pattern classification using neural networks,» Science Direct, τόμ. 40, 2007.
- [3] D. Reby, S. Lek, I. Dimopoulos, J. Joachim, J. Lauga and S. Aulagnier, «Artificial neural networks as a classification method in the behavioural sciences,» Behavioural Processes, τόμ. 40, 1997.
- [4] Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας, Η. Σακελλαρίου, <<Τεχνητή Νοημοσύνη,>> Β' Έκδοση, [Ηλεκτρονικό]. Available: <http://aibook.csd.auth.gr/include/slides/Chap18.pdf>. [Πρόσβαση 11 2020].
- [5] Κ. Διαμαντάρας, <<Μηχανική Μάθηση,>> Κλειδάριθμος 2019.

- [6] DataRobot, [Ηλεκτρονικό]. Available: <https://www.datarobot.com/wiki/unsupervised-machine-learning/>. [Πρόσβαση 11 2020].
- [7] Raul Rojas, << Neural Networks,>> [Ηλεκτρονικό]. Available: <http://page.mi.fu-berlin.de/rojas/neural/neuron.pdf>. [Πρόσβαση 10 2020].
- [8] Prateek Brajaj, <<Reinforcement learning,>> [Ηλεκτρονικό]. Available: <https://www.geeksforgeeks.org/what-is-reinforcement-learning/>. [Πρόσβαση 10 2020].
- [9] Prashant Gupta, <<Decision Trees in Machine Learning,>> [Ηλεκτρονικό]. Available: <https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052> [Πρόσβαση 10 2020].
- [10] Κ. Διαμαντάρας, <<Τεχνητά Νευρωνικά Δίκτυα,>> Κλειδάριθμος 2007.
- [11] C.G. Papanicolaou, I.C. Papantoniou, <<Textile Fibre Composites in Civil Engineering,>> [Ηλεκτρονικό]. Available: <https://www.sciencedirect.com/topics/engineering/genetic-algorithm>. [Πρόσβαση 11 2020].
- [12] Michael Affenzeller, Stephan Winkler, Stefan Wagner, Andreas Beham (2009). <<Genetic Algorithms and Genetic Programming,>> by CRC Press.
- [13] Shimon Ullman, Tomaso Poggio Danny Harari, Daneil Zysman, Darren Seibert <<Unsupervised learning Clustering,>> [Ηλεκτρονικό]. Available: <http://www.mit.edu/~9.54/fall14/slides/Class13.pdf>. [Πρόσβαση 11 2020].
- [14] Imad Dabbura, <<K-Means Clustering,>> [Ηλεκτρονικό]. Available: <https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a>. [Πρόσβαση 11 2020].
- [15] Junjie Wu, <<Advances in K-Means Clustering (A Data Mining Thinking),>> Springer-Verlag Berlin Heidelberg 2012.
- [16] Amir Sharif Ahmadian, <<Numerical Models of Submerged Breakwaters,>> Coastal Hydrodynamics and Morphodynamics 2016, Pages 109-126.
- [17] Mohammad-Taghi Vakil-Baghmisheh, Nikola Pavešić, <<Neurocomputing,>> Volume 62, December 2004, Pages 39-64.