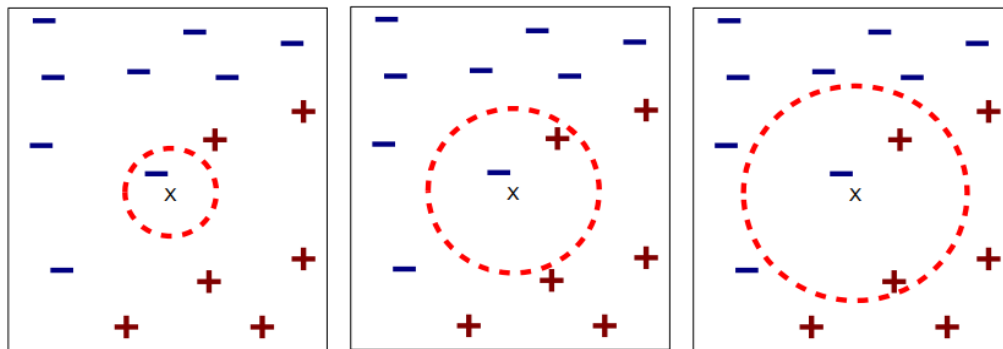


Η αναφορά της πρώτης προγραμματιστικής άσκησης στο μάθημα της *Μηχανικής Μάθησης (Machine Learning)* αφορά στη μελέτη αλγορίθμων ταξινόμησης δεδομένων, όπου αυτοί οι αλγόριθμοι είναι οι εξής:

- ***k-NN Nearest Neighbor με Ευκλείδεια απόσταση:***



(a) 1-nearest neighbor      (b) 2-nearest neighbor      (c) 3-nearest neighbor

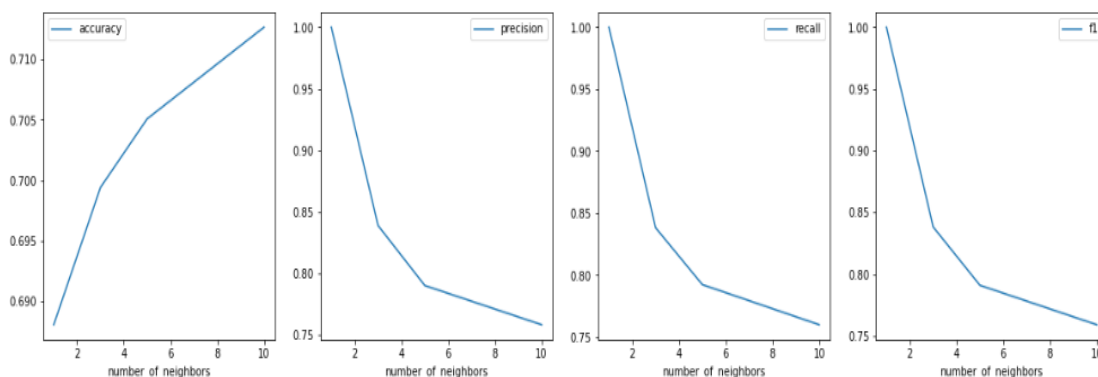
Ο συγκεκριμένος αλγόριθμος χρησιμοποιεί την τεχνική της εποπτευόμενης μάθησης για την ταξινόμηση και παλινδρόμηση των δεδομένων. Και στις δύο περιπτώσεις, η είσοδος αποτελείται από τα  $k$  πιο κοντινά παραδείγματα εκπαίδευσης (με την μικρότερη συναρτησιακή τιμή στην μετρική της απόστασης, η οποία καθορίζεται από τον χρήστη) του συνόλου δεδομένων. Πιο συγκεκριμένα, στην πρώτη περίπτωση, η οποία είναι και ζήτημα μελέτης της άσκησης, η έξοδος είναι μια ιδιότητα μέλους μιας κλάσης, δηλαδή μία ετικέτα, η οποία υποδηλώνει την κλάση στην οποία ανήκει το αντικείμενο. Δηλαδή το αντικείμενο ταξινομείται σύμφωνα με το πλήθος ψηφοφοριών των γειτόνων του, με το αντικείμενο να εκχωρείται στην κατηγορία που είναι πιο κοινή μεταξύ των  $k$  πλησιέστερων γειτόνων του ( $k$  είναι ένας θετικός ακέραιος, συνήθως μικρός). Αν το  $k=1$ , τότε το αντικείμενο απλώς εκχωρείται στην κλάση αυτού του απλού πλησιέστερου γείτονα. Η υλοποίηση του συγκεκριμένου αλγορίθμου έγινε με την χρήση της βιβλιοθήκης `sklearn`. Όμως, λόγω του γεγονότος ότι δεν ήταν εξ αρχής γνωστές οι πραγματικές ετικέτες από το σύνολο των test δεδομένων, ο υπολογισμός των μετρικών **precision**, **recall** και **f1 score** πραγματοποιήθηκε στα train δεδομένα, ενώ στην πραγματικότητα οι παραπάνω μετρικές υπολογίζονται ανάμεσα στην πρόβλεψη των ετικετών του ταξινομητή και των πραγματικών ετικετών, δηλαδή γνωρίζοντας ποιες είναι οι πραγματικές ετικέτες ελέγχουμε την ευστοχία των προβλέψεων του ταξινομητή με την χρήση των παραπάνω μετρικών.

Όνομα: Κοσμάς Αποστολίδης  
 ΑΜ: 4259  
 Εξάμηνο: 10  
 Ομάδα: mlteam35@cse-uoi

Τα αναλυτικά αποτελέσματα των προηγούμενων μετρικών απεικονίζονται στο Data Frame αλλά και στις γραφικές παραστάσεις:

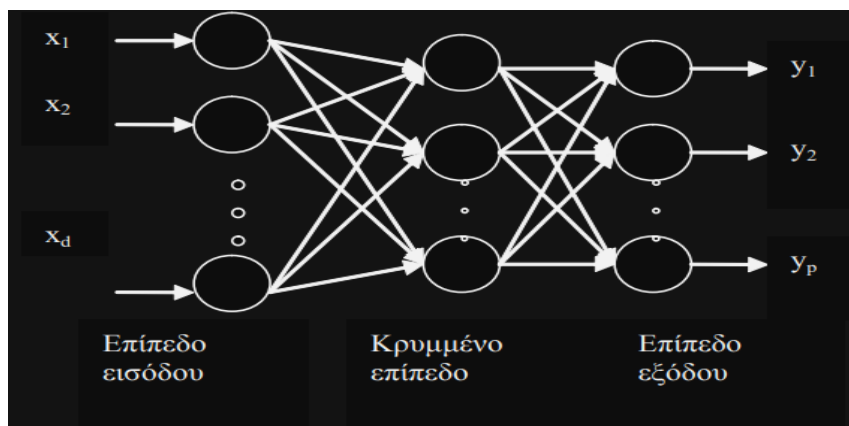
	accuracy	precision	recall	f1
number_of_neighbors				
1	0.68809	1.000000	1.000000	1.000000
3	0.69943	0.839029	0.838275	0.838315
5	0.70510	0.790334	0.792453	0.791139
10	0.71266	0.758496	0.760108	0.759209

<AxesSubplot: xlabel='number\_of\_neighbors'>



Στο συμπέρασμα που απορρέει από τις γραφικές παραστάσεις είναι ότι με την αύξηση του πλήθους των  $k$  κοντινότερων γειτόνων, μειώνονται και οι τιμές των αντίστοιχων μετρικών.

- **Multiplayer Perceptron:**



Είναι ένα τεχνητό νευρωνικό δίκτυο εισόδου-εξόδου το οποίο αποτελείται από **εσωτερικές κρυμμένες(hidden) μονάδες perceptrons**, όπου κάθε κρυμμένη μονάδα είναι διασυνδεδεμένη με κάθε άλλη μονάδα του προηγούμενου και του επόμενου επιπέδου, δημιουργώντας έτσι ένα **full mesh** δίκτυο. Η εκμάθηση ενός perceptron(νευρώνα) πραγματοποιείται διαφοροποιώντας τα βάρη της σύνδεσης κάθε φορά όπου

Όνομα: Κοσμάς Αποστολίδης  
 ΑΜ: 4259  
 Εξάμηνο: 10  
 Ομάδα: mlteam35@cse-uoi

πραγματοποιείται η επεξεργασία ενός τεμαχίου δεδομένων, σύμφωνα με το μέγεθος του σφάλματος στην έξοδο, σε σύγκριση με το αναμενόμενο αποτέλεσμα. Αυτό έχει ως αποτέλεσμα τα νευρωνικά δίκτυα MLP να είναι παραδείγματα εποπτευόμενων αλγορίθμων μάθησης. Επίσης, η αποτελεσματικότητα της μάθησης πραγματοποιείται με την χρήση της τεχνικής της **οπισθοδιάδοσης**, μιας γενίκευσης του αλγορίθμου των ελαχίστων τετραγώνων στο γραμμικό perceptron. Επιπλέον, η συνάρτηση ενεργοποίησης που κατέχει κάθε νευρώνας στο/α κρυμμένο/α επίπεδο/α αποφασίζει εάν θα πρέπει να ενεργοποιηθεί ή όχι ο αντίστοιχος νευρώνας του αντίστοιχου επιπέδου. Δηλαδή, αυτό σημαίνει ότι αποφασίζει εάν η είσοδος του νευρώνα στο δίκτυο είναι σημαντική ή όχι στη διαδικασία πρόβλεψης. Στα πλαίσια της άσκησης, ζητήθηκε να υλοποιηθεί ένα τεχνητό νευρωνικό δίκτυο ενός επιπέδου με **50,100,200 κρυμμένες μονάδες** επεξεργασίας και με δύο επίπεδα με **(50,25)**, **(100,50)**, **(200,100) κρυφές μονάδες επεξεργασίας** για κάθε επίπεδο αντίστοιχα. Στην πρώτη περίπτωση υπολογίστηκαν οι μετρικές χρησιμοποιώντας τα train δεδομένα:

```
Multi-layer Perceptron with 1 hidden layer and 50 hidden units
The number of iterations the Stochastic Gradient Descent has run: 1641
Name of the output activation function: softmax
```

```
Multi-layer Perceptron with 1 hidden layer and 100 hidden units
The number of iterations the Stochastic Gradient Descent has run: 1530
Name of the output activation function: softmax
```

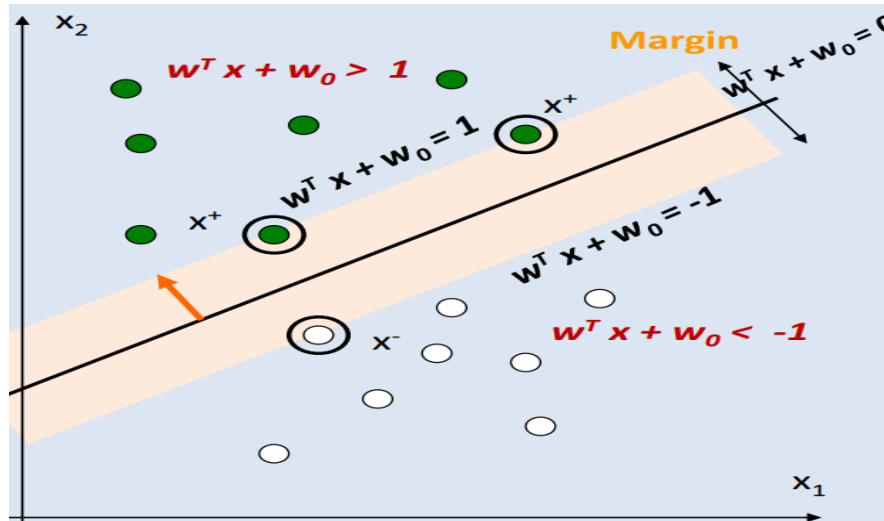
```
Multi-layer Perceptron with 1 hidden layer and 200 hidden units
The number of iterations the Stochastic Gradient Descent has run: 1656
Name of the output activation function: softmax
```

```
Multi-layer Perceptron classification metrics for 1 hidden layer and fixed amount of hidden units at that layer
```

	accuracy	precision	recall	f1
number_of_hidden_units				
50	0.71266	0.719924	0.727763	0.718003
100	0.72211	0.717270	0.725067	0.714553
200	0.71833	0.728662	0.735849	0.726710

Στην παραπάνω εικόνα φαίνονται ο αριθμός των επαναλήψεων που έτρεξε η μέθοδος βελτιστοποίησης **Stochastic Gradient Descent** και ποια **μέθοδος ενεργοποίησης** εφαρμόστηκε στις εξόδους του δικτύου, η οποία είναι **softmax**, όπου χρησιμοποιείται συχνότερα ως η τελευταία συνάρτηση ενεργοποίησης ενός νευρωνικού δικτύου για την κανονικοποίηση των εξόδων του σε μια κατανομή πιθανότητας για τις προβλεπόμενες κλάσεις εξόδου.

- Μηχανές Διανυσμάτων Υποστήριξης(Support Vector Machine , SVM)



Οι αλγόριθμοι αυτού του είδους χαρακτηρίζονται ως εποπτευόμενα μοντέλα μάθησης όπου αναλύοντας τα δεδομένα που παίρνουν ως είσοδο, αποφασίζουν για την κατηγορία τους λαμβάνοντας υπόψιν μία διαχωριστική γραμμή (όταν αναφερόμαστε σε ένα γραμμικό μοντέλο, αλλιώς σε μη γραμμικό). Έτσι, για ένα σύνολο δεδομένων παραδειγμάτων εκπαίδευσης, το καθένα επισημαίνεται ότι ανήκει σε μία από τις δύο κατηγορίες, δηλαδή όταν ένας αλγόριθμος εκπαίδευσης SVM δημιουργεί ένα μοντέλο που εκχωρεί νέα παραδείγματα στη μία ή στην άλλη κατηγορία, καθιστώντας το έναν μη πιθανολογικό δυαδικό γραμμικό ταξινομητή. Επιπλέον, αντιστοιχίζει τα παραδείγματα εκπαίδευσης σε σημεία στο χώρο, ώστε να μεγιστοποιεί το πλάτος του περιθωρίου(ή αλλιώς **margin**) μεταξύ των δύο αυτών κατηγοριών. Στη συνέχεια, τα νέα παραδείγματα εντάσσονται στον ίδιο χώρο και προβλέπεται ότι ανήκουν σε μια κατηγορία με βάση την εκάστοτε πλευρά του κενού που υπάρχει ανάμεσα στα δεδομένα. Στην παραπάνω εικόνα είναι ορατή η περίπτωση της γραμμικής SVM, στην οποία εφαρμόστηκε η στρατηγική **one-versus-all**, επειδή πραγματοποιείται ταξινόμηση πολλών κατηγοριών. Για το συγκεκριμένο ερώτημα της άσκησης υπολογίστηκαν, όπως και προηγουμένως, οι μετρικές precision, recall ,f1 score για την αξιολόγηση των προβλέψεων του ταξινομητή.

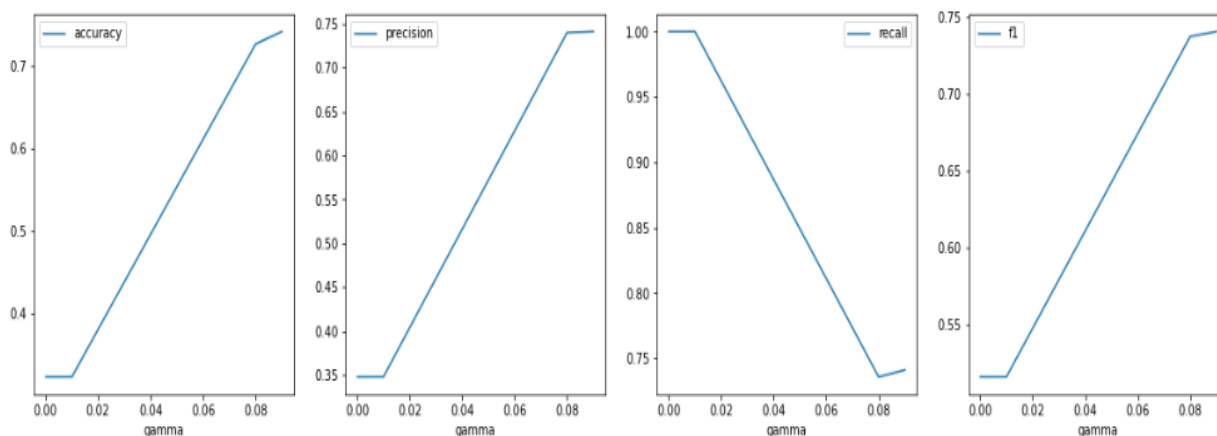
```
Support Vector Machine with linear kernel with one-versus-all decision_function_shape
accuracy precision recall f1
0 0.73724 0.742147 0.746631 0.743403
```

Το accuracy που υπολογίστηκε ισούται με **0.73724**. Από την άλλη, στην περίπτωση όπου η συνάρτηση πυρήνα για την SVM είναι η **Radial Basis Function(RBF)** :  $\exp(-\gamma \|x - x'\|^2)$ , όπου  $\gamma$  είναι μία θετική παράμετρος, τα αποτελέσματα των μετρήσεων για διάφορες τιμές της παραμέτρου  $\gamma$  είναι τα ακόλουθα:

Support Vector Machine with Radial Basis Function (RBF) kernel with one-versus-all decision\_function\_shape

gamma	accuracy	precision	recall	f1
0.0001	0.32325	0.347709	1.000000	0.516000
0.0010	0.32325	0.347709	1.000000	0.516000
0.0100	0.32325	0.347709	1.000000	0.516000
0.0800	0.72589	0.739936	0.735849	0.737304
0.0900	0.74102	0.741048	0.741240	0.740555

<AxesSubplot:xlabel='gamma'>



Κρίνεται αναγκαίο να σημειωθεί ότι για  $\gamma = 0.09$  επιτυγχάνεται μία αρκετά υψηλή τιμή για το accuracy του ταξινομητή. Επίσης και σε αυτήν την περίπτωση εφαρμόστηκε η στρατηγική one-versus-all

- **Naïve Bayes classifier**

Είναι μια τεχνική ταξινόμησης που βασίζεται στο θεώρημα του **Bayes**, με την υπόθεση ότι τα χαρακτηριστικά είναι μεταξύ τους **ανεξάρτητα**. Με άλλα λόγια, ο ταξινομητής Naïve Bayes υποθέτει ότι η παρουσία ενός συγκεκριμένου χαρακτηριστικού σε μια κλάση δεν σχετίζεται με την παρουσία οποιουδήποτε άλλου χαρακτηριστικού. Επίσης υποθέτουμε ότι όλα τα χαρακτηριστικά του συνόλου δεδομένων ακολουθούν την **κανονική κατανομή** ή **Gaussian** κατανομή με συνάρτηση πυκνότητας πιθανότητας

$$P(x_i | y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

όπου  $\sigma^2$  είναι η **διακύμανση** και  $\mu$  η **μέση τιμή**. Στη συνέχεια, για την εύρεση της κλάσης ενός δεδομένου έπρεπε να υπολογιστούν οι μετρικές της διακύμανσης και της μέσης τιμής για κάθε χαρακτηριστικό της κάθε κλάσης, όπως και οι πιθανότητες εμφάνισης της κάθε κλάσης. Αναλυτικότερα:

Όνομα: Κοσμάς Αποστολίδης

ΑΜ: 4259

Εξάμηνο: 10

Ομάδα: mlteam35@cse-uoi

Total number of records in test dataset: 371

Number of records with class Ghoul: 129

Number of records with class Goblin: 125

Number of records with class Ghost: 117

P(ghoul): 0.3477088948787062

P(goblin): 0.33692722371967654

P(ghost): 0.31536388140161725

Ghoul metrics

	bone_length	rotting_flesh	hair_length	has_soul	color
mean	0.517159	0.490515	0.655550	0.601634	0.470284
variance	0.014544	0.013307	0.019079	0.018175	0.062261

Goblin metrics

	bone_length	rotting_flesh	hair_length	has_soul	color
mean	0.429650	0.442357	0.544581	0.478663	0.442667
variance	0.010387	0.019680	0.013680	0.016203	0.060978

Ghost metrics

	bone_length	rotting_flesh	hair_length	has_soul	color
mean	0.347466	0.593759	0.373186	0.320023	0.492877
variance	0.013865	0.020207	0.013840	0.019541	0.063407

Έτσι, γνωρίζοντας πλέον τις διακυμάνσεις και τις μέσες τιμές κάθε χαρακτηριστικού της κλάσης, κρίνεται εύκολη η ταξινόμηση των δεδομένων σε μία από τις τρεις κλάσεις του προβλήματος. Στην παρακάτω εικόνα παρουσιάζεται ο τρόπος με τον οποίο πραγματοποιείται ο υπολογισμός της κλάσης ενός δεδομένου:

$$X: \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{bmatrix}$$

$x_1: \text{bone\_length}$   
 $x_2: \text{rotting\_flesh}$   
 $x_3: \text{hair\_length}$   
 $x_4: \text{has\_soul}$   
 $x_5: \text{color}$   
 $x_6: \text{type}$

$$P(X | \text{class} = \text{"Ghoul"}) = P(\text{"Ghoul"}) \times P(x_1 | \text{"Ghoul"}) \times P(x_2 | \text{"Ghoul"}) \times P(x_3 | \text{"Ghoul"}) \times P(x_4 | \text{"Ghoul"}) \times P(x_5 | \text{"Ghoul"})$$

$$P(x_1 | \text{"Ghoul"}) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_1 - \mu_{\text{Ghoul bone\_length}})^2}{2\sigma^2}}$$

Ο υπολογισμός των υπόλοιπων πιθανοτήτων είναι αντίστοιχος με τον προηγούμενο, με τη διαφορά ότι σε  $\sigma^2$  και  $\mu$  αντικαθίσταται με τις αντίστοιχες αμέσως των χαρακτηριστικών της κλάσης στην οποία ανήκει.

Όνομα: Κοσμάς Αποστολίδης

AM: 4259

Εξάμηνο: 10

Ομάδα: mlteam35@cse-uoi

Στο τέλος, έχοντας υπολογίσει πλέον τις τρεις πιθανότητες ένα συγκεκριμένο δεδομένο να ανήκει σε μία από τις τρεις δεδομένες κλάσεις Ghoul, Goblin, Ghost, πραγματοποιείται έλεγχος για το ποια από τις τρεις πιθανότητες είναι η μεγαλύτερη, δηλαδή  $\max\{P(x|\text{class} = \text{"Ghoul"}), P(x|\text{class} = \text{"Goblin"}), P(x|\text{class} = \text{"Ghost"})\}$ . Η μεγαλύτερη τιμή πιθανότητας υποδηλώνει τον τύπο της κλάσης όπου ανήκει ένα τυχαίο δεδομένο του συνόλου με τα test δεδομένα. Τέλος, για τον υπολογισμό των μετρικών precision, recall και f1, υλοποιήθηκε ο ίδιος ταξινομητής naïve bayes, αλλά αυτή τη φορά με την χρήση της βιβλιοθήκης sklearn. Τα αποτελέσματα των παραπάνω μετρικών απεικονίζονται στην ακόλουθη εικόνα.

### Metrics for Gaussian Naive Bayes Classifier

	accuracy	precision	recall	f1
0	0.73534	0.759417	0.760108	0.759732

Στις παρακάτω εικόνες παρουσιάζονται τα accuracies που παρουσιάστηκαν σε όλες τις προηγούμενες εικόνες τα ταξινομητών και τα οποία λήφθηκαν από τον ιστότοπο του kaggle

Όνομα: Κοσμάς Αποστολίδης

AM: 4259

Εξάμηνο: 10

Ομάδα: mlteam35@cse-uoi

Submission and Description	Private Score	Public Score	Use for Final Score
<a href="#">neural_network_predictions_for_2_hiddenlayers_and_(200100)_hiddenu...</a> 7 days ago by <a href="#">Kosmas Apostolidis</a> Message	0.71644	0.71644	<input type="checkbox"/>
<a href="#">neural_network_predictions_for_2_hiddenlayers_and_(10050)_hiddenuni...</a> 7 days ago by <a href="#">Kosmas Apostolidis</a> Message	0.72589	0.72589	<input type="checkbox"/>
<a href="#">neural_network_redictions_for_2_hiddenlayer_and_(5025)_hiddenunits....</a> 7 days ago by <a href="#">Kosmas Apostolidis</a> Message	0.72022	0.72022	<input type="checkbox"/>
<a href="#">neural_network_redictions_for_1_hiddenlayer_and_200_hiddenunits.csv</a> 7 days ago by <a href="#">Kosmas Apostolidis</a> Message	0.71833	0.71833	<input type="checkbox"/>
<a href="#">neural_network_redictions_for_1_hiddenlayer_and_100_hiddenunits.csv</a> 7 days ago by <a href="#">Kosmas Apostolidis</a> Message	0.72211	0.72211	<input type="checkbox"/>
<a href="#">neural_network_redictions_for_1_hiddenlayer_and_50_hiddenunits.csv</a> 7 days ago by <a href="#">Kosmas Apostolidis</a> Message	0.71266	0.71266	<input type="checkbox"/>
<a href="#">naive_bayes_clf_predictions.csv</a> 14 days ago by <a href="#">Kosmas Apostolidis</a> Naive Bayes Classifier(hardcoded)	0.73534	0.73534	<input type="checkbox"/>
<a href="#">naive_bayes_clf_predictions.csv</a> 14 days ago by <a href="#">Kosmas Apostolidis</a> Naive Bayes Classifier(hardcoded)	0.73534	0.73534	<input type="checkbox"/>
<a href="#">naive_bayes_clf_predictions.csv</a> 14 days ago by <a href="#">Kosmas Apostolidis</a> Naive Bayes Classifier(hardcoded)	0.73534	0.73534	<input type="checkbox"/>
<a href="#">naive_bayes_clf_predictions.csv</a> 16 days ago by <a href="#">Kosmas Apostolidis</a> Naive Bayes Classifier(hardcoded)	0.73534	0.73534	<input type="checkbox"/>
<a href="#">naive_bayes_clf_predictions.csv</a> 16 days ago by <a href="#">Kosmas Apostolidis</a> Naive Bayes Classifier(hardcoded)	0.73534	0.73534	<input type="checkbox"/>
<a href="#">0.095_gamma_rbf_svc_predictions.csv</a> 18 days ago by <a href="#">Kosmas Apostolidis</a> RBF kernel SVC Predictions for gamma = 0.095.	0.73724	0.73724	<input type="checkbox"/>



Όνομα: Κοσμάς Αποστολίδης

ΑΜ: 4259

Εξάμηνο: 10

Ομάδα: mlteam35@cse-uoi

<a href="#">0.09_gamma_rbf_svc_predictions.csv</a> 18 days ago by <a href="#">Kosmas Apostolidis</a> RBF kernel SVC Predictions for gamma = 0.09.	0.74102	0.74102	<input type="checkbox"/>
<a href="#">0.08_gamma_rbf_svc_predictions.csv</a> 18 days ago by <a href="#">Kosmas Apostolidis</a> RBF kernel SVC Predictions for gamma = 0.08.	0.72589	0.72589	<input type="checkbox"/>
<a href="#">1_gamma_rbf_svc_predictions.csv</a> 18 days ago by <a href="#">Kosmas Apostolidis</a> RBF kernel SVC Predictions for gamma = 1.	0.72967	0.72967	<input type="checkbox"/>
<a href="#">0.1_gamma_rbf_svc_predictions.csv</a> 18 days ago by <a href="#">Kosmas Apostolidis</a> RBF kernel SVC Predictions for gamma = 0.1.	0.73534	0.73534	<input type="checkbox"/>
<a href="#">0.01_gamma_rbf_svc_predictions.csv</a> 18 days ago by <a href="#">Kosmas Apostolidis</a> RBF kernel SVC Predictions for gamma = 0.01.	0.32325	0.32325	<input type="checkbox"/>
<a href="#">0.001_gamma_rbf_svc_predictions.csv</a> 18 days ago by <a href="#">Kosmas Apostolidis</a> RBF kernel SVC Predictions for gamma = 0.001.	0.32325	0.32325	<input type="checkbox"/>
<a href="#">0.0001_gamma_rbf_svc_predictions.csv</a> 18 days ago by <a href="#">Kosmas Apostolidis</a> RBF kernel SVC Predictions for gamma = 0.0001.	0.32325	0.32325	<input type="checkbox"/>
<a href="#">linear_svc_predictions.csv</a> 18 days ago by <a href="#">Kosmas Apostolidis</a> Linear SVC Predictions.	0.73724	0.73724	<input type="checkbox"/>
<a href="#">10_knn_predictions.csv</a> 18 days ago by <a href="#">Kosmas Apostolidis</a> KNearestNeighbors ML Algorithm with n_neighbors = 10.	0.71266	0.71266	<input type="checkbox"/>
<a href="#">10_knn_predictions.csv</a> 18 days ago by <a href="#">Kosmas Apostolidis</a> KNearestNeighbors ML Algorithm Predictions with n_neighbors = 10.	0.71266	0.71266	<input type="checkbox"/>
<a href="#">5_knn_predictions.csv</a> 18 days ago by <a href="#">Kosmas Apostolidis</a> KNearestNeighbors ML Algorithm Predictions with n_neighbors = 5.	0.70510	0.70510	<input type="checkbox"/>
<a href="#">3_knn_predictions.csv</a> 18 days ago by <a href="#">Kosmas Apostolidis</a> KNearestNeighbors ML Algorithm Predictions with n_neighbors = 3.	0.69943	0.69943	<input type="checkbox"/>
<a href="#">1_knn_predictions.csv</a> 18 days ago by <a href="#">Kosmas Apostolidis</a> KNearestNeighbors ML Algorithm Predictions with n_neighbors = 1.	0.68809	0.68809	<input type="checkbox"/>