#### Ομάδα 23

Νοταράς Ευάγγελος 10084 <u>enotaras@ece.auth.gr</u>

Καλλιμάνης Ιωάννης 10007 <u>ikallima@ece.auth.gr</u> Εργασία - Αναγνώριση <u>Προτύπων & Μηχανική Μάθηση</u> Περιεχόμενα

Μέρος Α: Ταξινομητής Bayes

Μέρος Β: Ταξινομητής k - NN

Μέρος Γ: Ταξινομητής SVM

Μέρος Δ: Διερεύνηση τρόπου ταξινόμησης

### Μέρος Α: Περίληψη

- Στο μέρος Α εξετάστηκε ο ταξινομητής bayes για την επίλυση του συγκεκριμένου προβλήματος ταξινόμησης.
- Ο ταξινομητής Bayes είναι μια στατιστική μέθοδος που έχει τις ρίζες της στο Θεώρημα του Bayes.
- Ο ταξινομητής είναι εύκολος ως προς της κατανόηση και την εφαρμογή του με γρήγορη ταχύτητα εκπαίδευσης μιας και περιλαμβάνει τον υπολογισμό απλών πιθανοτήτων.
- Εξετάστηκαν δύο διαφορετικές περιπτώσεις:
  - Χρησιμοποιώντας ίδιο πίνακα συνδιασποράς για όλες τις κλάσεις
  - Χρησιμοποιώντας διαφορετικό πίνακα συνδιασποράς για κάθε κλάση

### Μέρος Α: Περιγραφή Κώδικα

- Για την ταξινόμηση με την χρήση του ταξινομητή Bayes δημιουργήθηκε η κλάση BayesianClassifier
- Για την χρήση της κλάσεις, πρέπει πρώτα να δημιουργηθεί ένας ταξινομητής και να εκπαιδευτεί για ένα σύνολο δεδομένων. Έπειτα, δίνοντας δεδομένα, ο ταξινομητής προβλέπει την έξοδο.
- Ορίστηκαν οι συναρτήσεις:
  - ο fit(X, y): Εκπαιδεύει τον ταξινομητή με δεδομένα εκπαίδευσης X και αντίστοιχες ετικέτες y. Υπολογίζει τις apriori πιθανότητες, τους μέσους όρους των κλάσεων και τους πίνακες συνδιασποράς με βάση τον επιλεγμένο τρόπο.
  - ο predict(X): Προβλέπει τις ετικέτες κλάσης για τα δεδομένα εισόδου X με τη Maximum Likelihood τεχνική.
  - o multivariate\_normal\_pdf(x, mean, cov): Υπολογίζει την πολυμεταβλητή κανονική συνάρτηση πυκνότητας πιθανότητας για ένα δεδομένο σημείο δεδομένων.

### Μέρος Α: Πίνακας Συνδιασποράς

- Η κλάση BayesianClassifier δέχεται σαν όρισμα μια παράμετρο mode με την οποία επιλέγεται αν οι κλάσεις θα έχουν διαφορετικό ή κοινό πίνακα συνδιασποράς.
- Στην περίπτωση του διαφορετικού πίνακα συνδιασποράς, ο πίνακας συνδιασποράς κάθε κλάσης επιλέγεται χρησιμοποιώντας την συνάρτηση cov της βιβλιοθήκης numpy. Έτσι, ο κάθε πίνακας ορίζεται ως:

$$\sigma_i = \sum_{j=1}^N \frac{\left(x_j - \mu_i\right)^2}{N-1} \quad , \ \forall x_j \in \text{στην κλάση } i$$

# Μέρος Α: Κοινός πίνακας συνδιασποράς

 Στην περίπτωση του κοινού πίνακα συνδιασποράς, ο πίνακας συνδιασποράς ορίζεται ως:

$$\sigma = \sum_{i=1}^{N} \left(x_i - \mu_j\right)^2 /_N \quad \text{όπου N είναι το σύνολο όλων των δειγμάτων}\,.$$

Υποθέτουμε επίσης ότι τα πρώτα  $N_{\rm A}$  δείγματα ανήκουν στην κλάση 1, τα επόμενα

 $N_{_B}$  στην κλάση 2 και τα τελευταία  $N_{_\Gamma}$  στην κλάση 3,  $\left($ ισχύει  $N_{_A}+N_{_B}+N_{_\Gamma}=N\right)$ .

Οπότε, έχουμε:

$$\sigma = \frac{\sum_{i=1}^{N} (x_i - \mu_j)^2}{N} = \frac{\sum_{i=1}^{N_A} (x_i + \mu_1)^2 + \sum_{i=N_A+1}^{N_A+N_B} (x_i + \mu_2)^2 + \sum_{i=N_A+N_B+1}^{N} (x_i + \mu_3)^2}{N}$$

$$= \frac{N_A}{N} \cdot \frac{\sum_{i=1}^{N_A} (x_i + \mu_1)^2}{N_A} + \frac{N_B}{N} \cdot \frac{\sum_{i=N_A+1}^{N_A+N_B} (x_i + \mu_2)}{N_B} + \frac{N_\Gamma}{N} \cdot \frac{\sum_{i=N_A+N_B+1}^{N} (x_i + \mu_3)}{N_\Gamma}$$

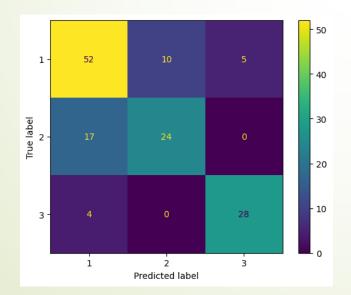
όπου αν θέσουμε ως apriori, την apriori πιθανότητα της κλάσης j.

$$\sigma = \sum \left( apriori_i \cdot \sigma_i \right) \mu \epsilon \ \sigma_i \ \alpha \pi \delta \ \tau \eta \nu \ \pi \rho o \eta \gamma o \delta \mu \epsilon \nu \eta \ \sigma \chi \acute{\epsilon} \sigma \eta$$
 
$$i = 1$$

### Μέρος Α: Αποτελέσματα

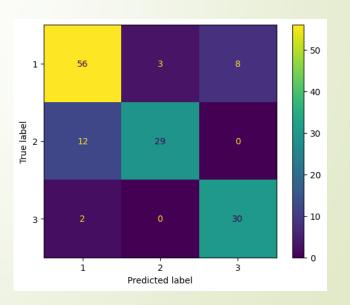
## Κοινός πίνακας συνδιασποράς

- Accuracy = 74,29 %
- ► F1 score = 73,99 %
- Elapsed Time = 0.042 seconds

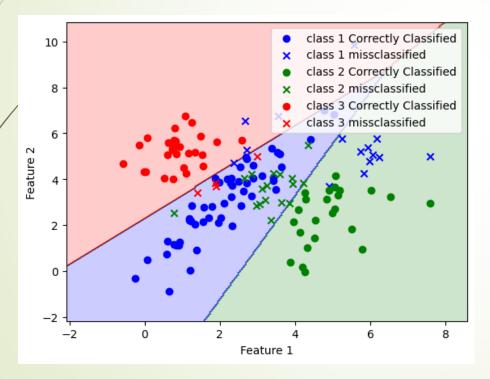


#### Διαφορετικός πίνακας συνδιασποράς

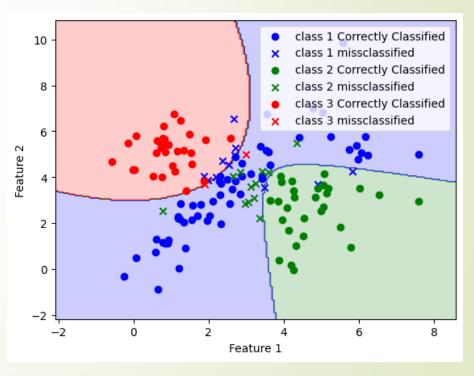
- Accuracy = 82,14 %
- F1 score = 81,98 %
- Elapsed Time = 0.028 seconds



Κοινός πίνακας συνδιασποράς



#### Διαφορετικός πίνακας συνδιασποράς



- Παρατηρούμε ότι στην πρώτη περίπτωση (με κοινό πίνακα συνδιασποράς) οι περιοχές απόφασης χωρίζονται γραμμικά, ενώ στην άλλη περίπτωση ο διαχωρισμός των πεδίων γίνεται έντονα μη γραμμικώς. Αυτό συμβαίνει, καθώς στην πρώτη περίπτωση τα όρια απόφασης καθορίζονται από γραμμικές εξισώσεις στο feature space, ενώ στην δεύτερη περίπτωση επιτρέπεται μεγαλύτερη ευελιξία, διότι κάθε κλάση έχει το δικό της σύνολο αποκλίσεων και συνδιασπορών.
- Η πρώτη περίπτωση φαίνεται να είναι μια πιο γενικευμένη μορφή της αρχιτεκτονικής, ενώ η δεύτερη φαίνεται να παρουσιάζει καλύτερη προσαρμογή στα δεδομένα εκπαίδευσης, γεγονός που την καθιστά κάπως πιο επιρρεπή στο overfit.

### Μέρος Α: Συμπεράσματα

- Στη περίπτωση που οι κλάσεις έχουν κοινό πίνακα συνδιασποράς, το πρόβλημα απλοποιείται και μειώνεται η υπολογιστική πολυπλοκότητα. Για αυτό και ο αλγόριθμος εκτελείται γρηγορότερα. Θα γινόταν πιο εμφανές σε πιο πολύπλοκα προβλήματα με μεγαλύτερο αριθμό feature.
- Επιπλέον, στην περίπτωση αυτή μειώνεται το ρίσκο για overfit καθώς το μοντέλο δεν υπερπροσαρμόζεται στα δεδομένα εκπαίδευσης.
- Στην άλλη περίπτωση, με διαφορετικό πίνακα συνδιασποράς για κάθε κλάση, το μοντέλο εμφανίζει μεγαλύτερη ευελιξία. Για αυτό τον λόγο δίνει καλύτερες αποδόσεις και φαίνεται να επιλύει το συγκεκριμένο πρόβλημα αποδοτικότερα.

### Μέρος Β: Περίληψη

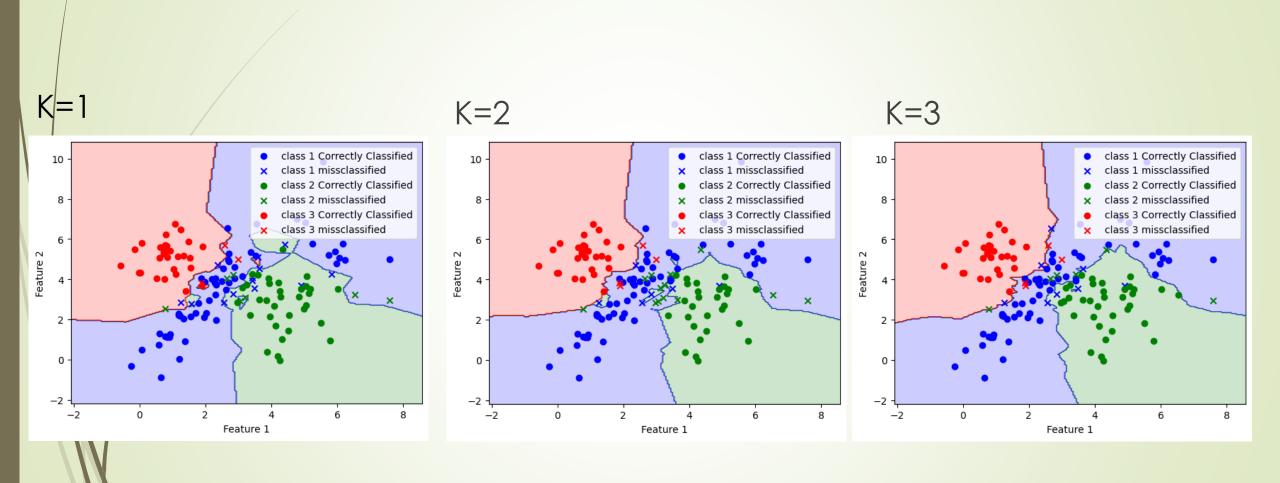
- Στο Β μέρος εξετάστηκε ο ταξινομητής των K-Nearest Neighbors για τιμές του Κ από 1 έως 10
- Ο ΚΝΝ αποτελεί έναν απλό αλγόριθμο ταξινόμησης. Αποφασίζει σε ποια κλάση ανήκει κάποιο δεδομένο εκτίμησης ανάλογα με την πλειοψηφία των κλάσεων που ανήκουν οι Κ κοντινότεροι γείτονες.
- Λειτουργεί με την λογική ότι κάποιο καινούργιο δείγμα θα ανήκει στην ίδια κλάση που ανήκουν τα όμοια του, όπου η ομοιότητα καθορίζεται από την απόσταση μεταξύ του καινούργιου δείγματος και των δειγμάτων εκπαίδευσης.

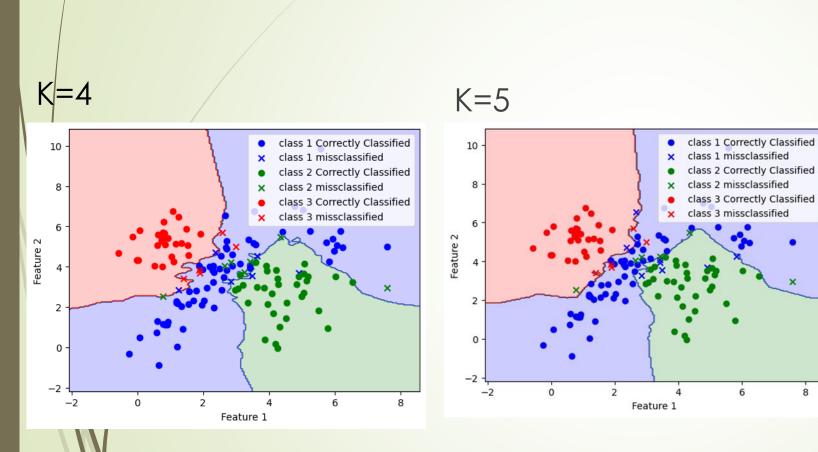
### Μέρος Β: Περιγραφή Κώδικα

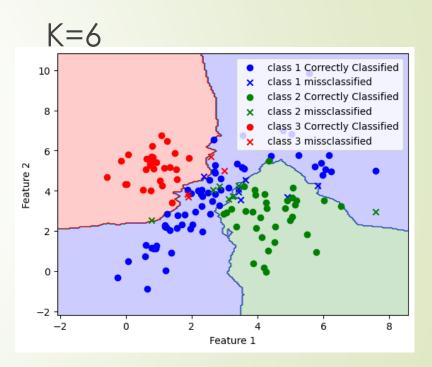
- ► Για την ταξινόμηση των δειγμάτων με την χρήση του ΚΝΝ αλγόριθμου, υλοποιήθηκε μια for loop ώστε το Κ να λαμβάνει τιμές από 1 έως 10.
- Στην συνέχεια χρησιμοποιήθηκε η έτοιμη συνάρτηση KNeighborsClassifier από την βιβλιοθήκη sklearn.neighbors με τις default ρυθμίσεις. Αυτό σημαίνει ότι υπήρξε ομοιόμορφη κατανομή των βαρών, δηλαδή όλα τα σημεία σε μια γειτονιά είναι εξίσου σημαντικά παρουσιάζοντας ίσα βάρη. Επίσης, η συνάρτηση αυτή χρησιμοποιεί την απόσταση minkowski που θεωρείται ως μια γενίκευση τόσο της ευκλείδειας, όσο και της manhattan. Η default περίπτωση έχει p=2 όπου η απόσταση αυτή ταυτίζεται με την ευκλείδεια.

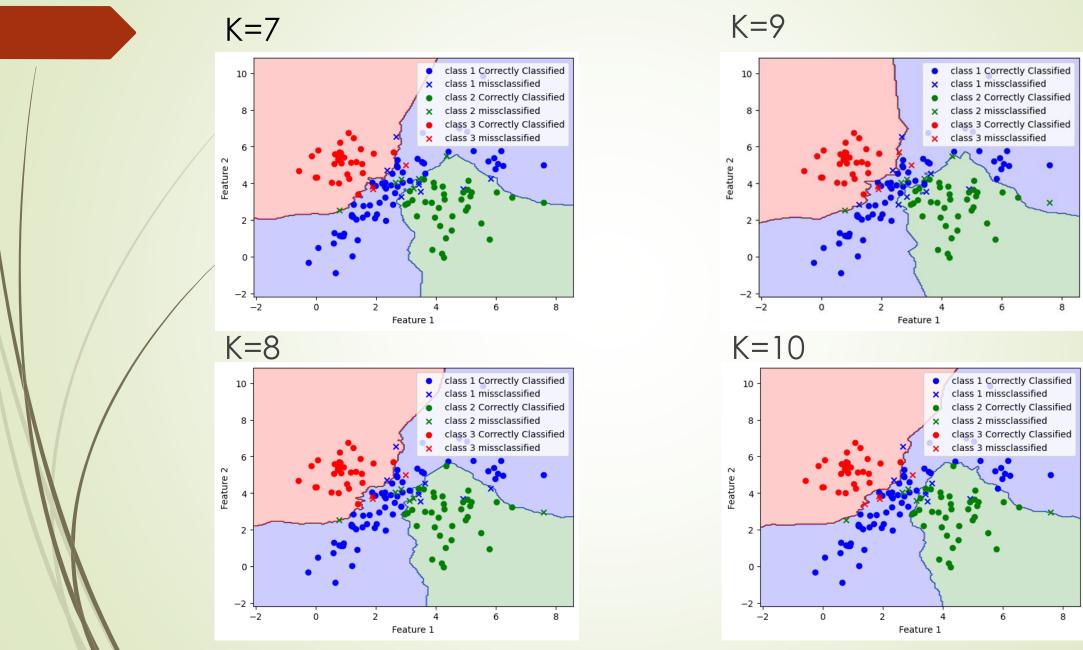
### Μέρος Β: Αποτελέσματα

	K=1	K=2	K=3	K=4	K=5	K=6	K=7	K=8	K=9	K=10
Accuracy	83.571%	83.571%	87.143%	87.142%	85.714%	87.143%	87.857%	87.857%	88.571%	88.571%
F1 score	83.582%	83.412%	87.139%	87.12%	85.714%	87.149%	87.857%	87.857%	88.587%	88.586%
Time Elapsed	0.022 s	0.043 s	0.030 s	0.024 s	0.025 s	0.023 s	0.026 s	0.024 s	0.030 s	0.040 s









- Παρατηρούμε ότι για μικρές τιμές του k οι περιοχές απόφασης είναι σαφώς πιο πολύπλοκες. Αυτό συμβαίνει διότι όσο κινούμαστε σε μικρότερες τιμές του k το μοντέλο εμφανίζει αυξημένη επιρροή από συγκεκριμένα δείγματα εκπαίδευσης, με αποτέλεσμα να οδηγείται σε overfitting.
- Αντίστοιχα, παρατηρούμε ότι όσο αυξάνεται το k οι καμπύλες απόφασης γίνονται πιο ομαλές, κάνοντας το μοντέλο μας πιο γενικευμένο. Παρόλα αυτά, υπάρχει κίνδυνος, για μεγάλες τιμές του k, να μην αναγνωρίζει αποδοτικά πρότυπα.

### Μέρος Β: Συμπεράσματα

- Παρατηρούμε ότι ο Knn αλγόριθμος, αποτελεί έναν πολύ απλό στην υλοποίηση αλγόριθμο, καθώς και πολύ αποδοτικό.
- Για το συγκεκριμένο πρόβλημα, επιλέγεται η τιμή k=10, διότι από την μία έχει την μεγαλύτερη ακρίβεια (όπως και ο k=9), ενώ επίσης καθιστά το μοντέλο μας πιο γενικευμένο.
- Συγκρίνοντας τα αποτελέσματα με τους ταξινομητές bayes, βλέπουμε από το παρακάτω πινακάκι, ότι ο ταξινομητής knn με k=10, είναι και πιο γρήγορος αλλά και αρκετά πιο αποτελεσματικός.

	Bayes_shared_cov	Bayes_diff_cov	K=10
Accuracy	74,29 %	82,14%	88.571%
F1 score	79,31%	79,31%	88.586%
Time elapsed	0.049 s	0.063 s	0.040 s

### Μέρος Γ: Περίληψη

- Στο τρίτο στάδιο του κώδικα μας πραγματοποιούνται μοντέλα εκτίμησης με την χρήση Support Vector Machines (SVM).
- Στόχος των SVM είναι να μεγιστοποιήσουν το περιθώριο margin που αυξάνει την απόσταση μεταξύ των κοντινότερων δειγμάτων δύο κλάσεων, έτσι ώστε αν εισαχθεί κάποιο δείγμα με περεταίρω θόρυβο να μην ξεπεράσει το διαχωριστικό επίπεδο των δύο κλάσεων.
- Εξετάστηκε περίπτωση με γραμμική συνάρτηση πυρήνα (kernel), έτσι ώστε να διαπιστωθεί αν το πρόβλημα είναι γραμμικώς διαχωρίσιμο στο επίπεδο των χαρακτηριστικών του.
- Εξετάσθηκε και η περίπτωση με συναρτήσεις πυρήνα τύπου Radial Basis Function (RBF), οι οποίες είναι αποτελεσματικότερες στο να αναγνωρίζουν μη γραμμικά μοτίβα. Μέσω των συναρτήσεων αυτών, τα δεδομένα ανάγονται σε μεγαλύτερες διαστάσεις, στις οποίες είναι γραμμικώς διαχωρίσιμα.

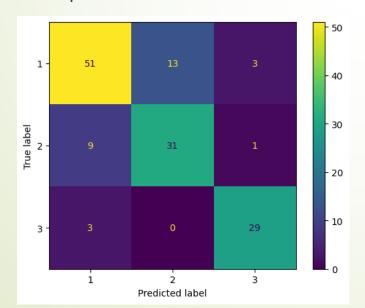
### Μέρος Γ: Περιγραφή κώδικα

- Για την υλοποίηση των SVM χρησιμοποιήθηκε από την βιβλιοθήκη sklearn.svm η κλάση SVC, ενώ για την επιλογή των βέλτιστων υπερπαραμέτρων χρησιμοποιήθηκε από την βιβλιοθήκη sklearn.model\_selection η συνάρτηση GridSearchCV, που επιλέγει το βέλτιστο μοντέλο με τεχνική cross validation.
- Για το γραμμικό SVM ελέγχθηκαν διάφορες τιμές για την παράμετρο C και την παράμετρο class\_weight. Μικρότερες τιμές του C ενθαρρύνουν ένα υπερεπίπεδο με μεγαλύτερο περιθώριο αλλά μπορεί να ταξινομήσουν λανθασμένα ορισμένα σημεία. Μεγαλύτερα C τιμωρούν περισσότερο τις λανθασμένες ταξινομήσεις, οδηγώντας σε ένα επίπεδο μικρότερου περιθωρίου. Η παράμετρος class\_weight επηρεάζει τον συντελεστή C κάθε κλάσης και τον αλλάζει σε class\_weight[i]\*C.
- Για τον RBF ελέγχθηκαν διάφορες διαφορές τιμές για τα C και για τα gamma. Ένα μικρό gamma επιτρέπει στα σημεία να έχουν μια πιο σφαιρική επιρροή, ενώ ένα μεγάλο gamma καθιστά το όριο απόφασης περισσότερο εξαρτώμενο από τα σημεία που βρίσκονται κοντά σε αυτό.

### Μέρος Γ: Αποτελέσματα

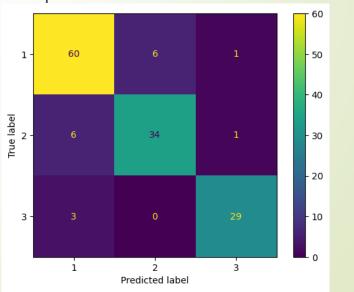
#### Kernel = 'linear'

- Το μοντέλο που επιλέχθηκε έχει υπερπαραμέτρους: C=0.57, class\_weight='balanced'
- Accuracy = 79.286%
- ► F1 score = 79.306%
- Elapsed Time = 0.061 seconds



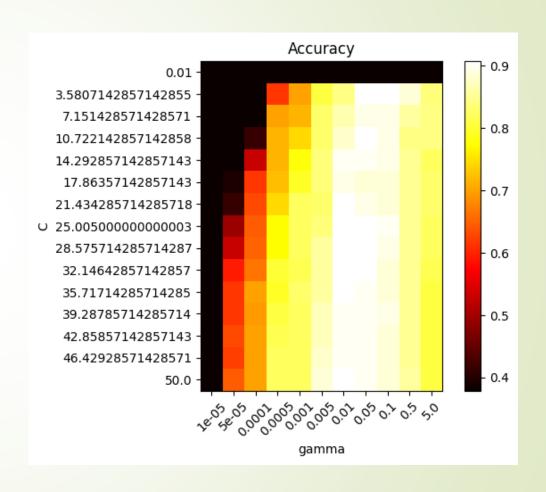
#### Kernel = 'rbf' (default)

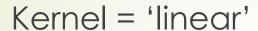
- Το μοντέλο που επιλέχθηκε έχει υπερπαραμέτρους: C=13.9, class\_weight=None, gamma='scale'
- Accuracy = 87.857%
- ► F1 score = 87.856%
- Elapsed Time = 0.010 seconds

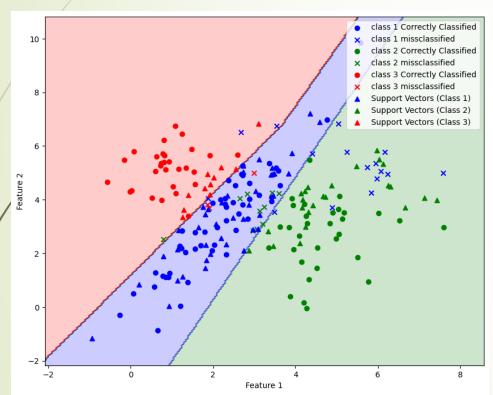


### Μέρος Γ: Επίδραση Υπερπαραμέτρων

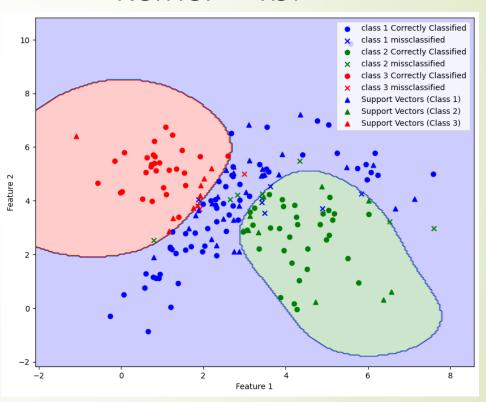
- Προκειμένου να ελέγξουμε τι επίδραση έχουν οι υπερπαράμετροι στο RBF SVM δημιουργήθηκε ένα heatmap του test\_accuracy ως συνάρτηση του C και του gamma.
- Παρατηρούμε για πολύ μικρές του C η απόδοση είναι πολύ χαμηλή, ενώ όταν η τιμή του ξεπεράσει ένα κατώφλι δεν υπάρχει μεγάλη επίδραση στην ακρίβεια.
- Παρατηρείται, επίσης, ότι το test\_accuracy συναρτήσει του gamma έχει μια παραβολική μορφή. Μεγιστοποιείται σε τιμές κοντά στο 0.01- 0.05, ενώ όσο τις ξεπερνά μειώνεται η ακρίβεια.







#### Kernel = 'rbf'



- Όπως ήταν αναμενόμενο τα δεδομένα με την χρήση γραμμικών SVM διαχωρίζονται με γραμμικά επίπεδα στο επίπεδο που βρίσκονται, ενώ με την χρήση rbf ως συνάρτησή πυρήνα τα δεδομένα διαχωρίζονται και μη γραμμικά.
- Από τα αποτελέσματα των δύο μοντέλων παρατηρούμε ότι τα δεδομένα μας δεν είναι γραμμικώς διαχωρίσιμα στον χώρο των διαστάσεων, που ορίζουν τα χαρακτηριστικά τους.

### Μέρος Γ: Συμπεράσματα

- Είναι εμφανές ότι ο γραμμικώς διαχωρισμός δεν αποτελεί αποδοτική λύση στο συγκεκριμένο πρόβλημα.
- Τα SVM με συνάρτηση πυρήνα RBF είναι τόσο ταχύτερα όσο και αποδοτικότερα από τα γραμμικά SVM.
- Επιπλέον συγκρίνοντας τα αποτελέσματα του SVM με τις προηγούμενες μεθόδους, σύμφωνα με το παρακάτω πινακάκι, για την λύση του συγκεκριμένου προβλήματος, εξακολουθεί ο KNN με k=10 να είναι οριακά ο πιο αποδοτικός, παρόλα αυτά το SVM με συνάρτηση πυρήνα RBF δεν απέχει τόσο πολύ σε απόδοση καθώς επιπλέον είναι και 4 φορές πιο γρήγορο:

	SVM (rbf)	Bayes_diff_cov	K=10
Accuracy	87,857%	82,14%	88.571%
F1 score	87.856%	79,31%	88.586%
Time Elapsed	0.010 s	0.063 s	0.040 s

### Μέρος Δ: Εισαγωγή

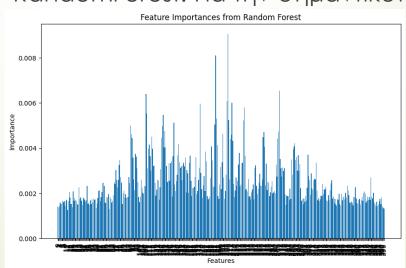
- Στο τέταρτο κομμάτι της εργασίας μας, μας δόθηκε ένα άγνωστο dataset. Σκοπός ήτανε να το μελετήσουμε, να το προεπεξεργαστούμε και να δημιουργήσουμε ένα μοντέλο, το οποίο θα έκανε την καλύτερη δυνατή πρόβλεψη ετικετών.
- Στην συνέχεια μας δόθηκε ένα dataset με άγνωστες ετικέτες και μας ζητήθηκε η πρόβλεψη τους ώστε να γίνει η αξιολόγηση του μοντέλου μας.

### Μέρος Δ: Διερεύνηση μοντέλων

- Για την επιλογή του καταλληλότερου μοντέλου, εξετάσθηκαν διάφορες αρχιτεκτονικές με σκοπό την εύρεση εκείνης που ταιριάζει περισσότερο στα δεδομένα του προβλήματος μας.
- Αρχικά χωρίσαμε τα δεδομένα μας με αναλογία 70-30 σε δεδομένα εκπαίδευσης και εκτίμησης αντίστοιχα, ώστε να έχουμε μια μετρική ώστε να συγκρίνουμε τα διαφορετικά μοντέλα που επρόκειτο να εξετάσουμε.
- Δοκιμάστηκε η χρήση SVM με συνάρτηση πυρήνα RBF. Έγινε ένα tuning για την εξαγωγή των βέλτιστων υπερπαραμέτρων για τις ανάγκες του προβλήματος μας και δοκιμάστηκε το βέλτιστο μοντέλο για την πρόβλεψη των ετικετών στα δεδομένα εκτίμησης.
- Στην συνέχεια, εξετάστηκε η ακρίβεια του RandomForestClassifier. Έγινα και σε αυτήν την περίπτωση ένα tuning για τον υπολογισμό των ιδανικότερων υπερπαραμέτρων του.

### Μέρος Δ: Διερεύνηση μοντέλων

- Στην συνέχεια δοκιμάστηκε και η απόδοση το KNearestNeighbors (KNN), αφού υλοποίηθηκε σε προηγούμενο ερώτημα. Εξετάστηκε ο αλγόριθμος για τιμές του k από 1 έως 30. Παρόλα αυτά, δεν παρατηρήθηκε κάποια βελτίωση στα αποτελέσματα μας.
- Έπειτα, δοκιμάστηκε να μειωθεί η διάσταση του προβλήματος. Για αρχή, αφαιρέθηκαν κάποια χαρακτηριστικά, με βάση την σημαντικότητα της σύμφωνα με την RandomForest. Για την σημαντικότητα τους:



### Μέρος Δ: Διερεύνηση μοντέλων

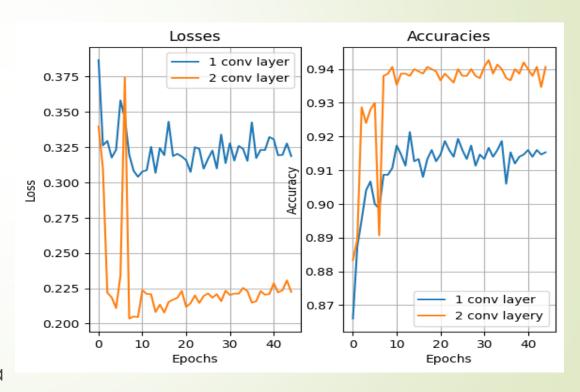
- Η μείωση διάστασης δοκιμάστηκε και με χρήση Principal Component Analysis
   (PCA) για διάφορες τιμές του αριθμού των νέων χαρακτηριστικών ώστε να δούμε ποια έχει τα καλύτερα αποτελέσματα.
- Τα νέα δεδομένα μετά την μείωση της διάστασης με τους παραπάνω τρόπους χρησιμοποιήθηκαν στην αρχιτεκτονική των Support Vector Machines. Αν και αύξησαν κάπως την απόδοση, δεν ήταν ακόμα σε ικανοποιητικό επίπεδο.
- Ακολούθως, εφαρμόστηκε η αρχιτεκτονική των πλήρως συνεκτικών νευρωνικών δικτύων (MLP). Έγινε μια διερεύνηση πάνω στην αρχιτεκτονική του μοντέλου και στην επιλογή του βέλτιστων υπερπαραμέτρων του. Όπως ήταν αναμενόμενο η χρήση MLP παρουσίασε αυξημένη απόδοση, σε σχέση με τις προηγούμενες υλοποιήσεις.
- Επιπλέον, δοκιμάστηκε και η αποθορυβοποίηση των δεδομένα μέσω
   ΑυτοEncoder, χωρίς όμως να υπάρχει σημαντική βελτίωση στα αποτελέσματα μας.

### Μέρος Δ: Συνελικτικό Δίκτυο

- Τα συνελικτικά νευρωνικά δίκτυα, αποτελούν ένα είδος πολυστρωματικών νευρωνικών δικτύων, τα οποία περιλαμβάνουν συνελικτικά επίπεδα. Το είδος των δικτύων αυτών ενδείκνυται όταν τα δεδομένα μας είναι εικόνες και βίντεο.
- Η βασική ιδιότητά τους είναι ότι μπορούν να εντοπίζουν χαρακτηριστικά εικόνων όπως φωτεινά ή σκοτεινά σημεία (ή συγκεκριμένο χρώμα), τις ακμές σε διάφορους προσανατολισμούς, πρότυπα, και ούτω καθεξής.
- Στην πορεία της διερεύνησης για την επιλογή της καταλληλότερης αρχιτεκτονικής, δοκιμάστηκε δίκτυο και αυτού του είδους.
- ► Για να συμβεί αυτό, τα 400 χαρακτηριστικά της εισόδου έπρεπε να μετασχηματιστούν σε πίνακα 1x20x20, θεωρώντας ότι έχουμε ένα κανάλι χρώματος και από 20 pixels πλάτους και ύψους αντίστοιχα, έτσι ώστε να μπορεί να συμβεί συνέλιξη δύο διαστάσεων.
- Κατά την χρήση αυτής της αρχιτεκτονικής παρατηρήθηκε μεγάλη βελτίωση της απόδοσης του μοντέλου μας. Για αυτό και επιλέχθηκε να χρησιμοποιηθεί αυτή η αρχιτεκτονική.

### Μέρος Δ: Αρχιτεκτονική Συνελκτικού Δικτύου

- Προκειμένου να επιλεχθεί ο καταλληλότερος αριθμός συνελικτικών στρωμάτων, έγινε διερεύνηση πάνω στον αριθμό των συνελίξεων, καθώς και των διαστάσεων τις συνέλιξης.
- Λόγω της μικρής διάστασης των εικόνων μας (20x20), δεν ήταν δυνατή η προσθήκη πολλών στρωμάτων MaxPooling.
- Σαν βάση για την αρχιτεκτονική του μοντέλου χρησιμοποιήθηκε αυτή του VGG. Δίπλα, παρουσιάζονται τα αποτελέσματα του δικτύου με 1 στρώμα (2 συνελικτικά) και 2 στρωμάτων (2 συνελικτικά στο ένα και 3 στο άλλο).
- Είναι προφανές, ότι το δίκτυο με 2 συνελικτικά στρώματα έχει αρκετά καλύτερες επιδόσεις.

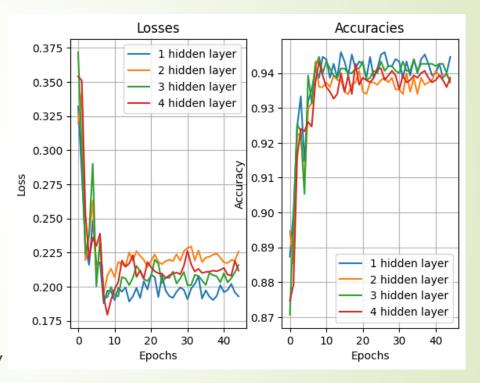


### Μέρος Δ: Αρχιτεκτονική Συνελκτικού Δικτύου

- Έχοντας επιλέξει τον αριθμό των νευρωνικών στρωμάτων στο δίκτυο, έγινε στην συνέχεια μια περεταίρω διερεύνηση όσον αφορά την αρχιτεκτονική των dense στρωμάτων.
- Ως συνάρτηση ενεργοποίησης επιλέχτηκε η ReLU,η οποία είναι ιδιαίτερα δημοφιλής σε προβλήματα κατηγοριοποίησης με την χρήση CNN καθώς βοηθά να καταπολεμιστεί το πρόβλημα του Vanishing Gradient.
- Τα dense στρώματα αποτελούνται από fully connected κρυφά στρώματα νευρώνων, ενώ ανάμεσα τους προστίθενται στρώματα batch normalization καθώς και dropout.
- Πιο συγκεκριμένα, τα στρώματα batch normalization, είναι στρώματα τα οποία κανονικοποιούν τα δεδομένα στην είσοδο του κάθε στρώματος. Από την άλλη, το Dropout είναι μια τεχνική κατά την οποία μερικοί έξοδοι ενός στρώματος αγνοούνται με τυχαίο τρόπο με αποτέλεσμα να υπάρχει περισσότερος θόρυβος κατά τη διαδικασία της εκπαίδευσης και κάποιοι κόμβοι να αναγκάζονται να λειτουργούν περισσότερο. Έτσι, το μοντέλο δεν βασίζεται τόσο πολύ σε συγκεκριμένους νευρώνες και αποφεύγεται η υπερεκπαίδευση.

### Μέρος Δ: Αρχιτεκτονική Συνελκτικού Δικτύου

- Όσον αφορά την διερεύνηση, δοκιμάστηκαν αρχιτεκτονικές CNN με ένα, δύο, τρία και τέσσερα κρυφά στρώματα νευρώνων. Τα αποτελέσματα φαίνονται δίπλα:
- Παρατηρούμε ότι και οι 4 υλοποιήσεις, έχουν πανομοιότυπα αποτελέσματα στην απόδοση του μοντέλου. Από τα τέσσερα, επιλέγεται αυτό με τα τρία κρυφά στρώματα, μιας και τα αποτελέσματα του ήταν ικανοποιητικά και το βάθος του μπορεί να επιτρέψει την αναγνώριση και πιο πολύπλοκων μοτίβων.



### Μέρος Δ: Επιλογή Υπερπαραμέτρων

- Εφόσον επιλέχθηκε η αρχιτεκτονική του μοντέλου, το επόμενο στάδιο είναι η επιλογή των υπερπαραμέτρων του μέσω ενός fine tuning.
- Επιλέχθηκαν οι βέλτιστες τιμές για τα μεγέθη hidden\_size1, hidden\_size2 και hidden\_size3, τα οποία αποτελούν των αριθμών των νευρώνων σε κάθε ένα από τα κρυφά στρώματα του dense layer.
- Έγινε διερεύνηση και για τις τιμές dropout1 και dropout2 που συμβολίζουν το dropout rate του κάθε dense layer.
- To fine tuning έγινε με την χρήση cross validation και εξετάστηκαν οι παρακάτω τιμές:

hidden1=[128,256, 512, 1024] hidden2=[128,256, 512, 1024]

hidden3=[128, 256, 512, 1024]

dropout1=[0.2, 0.3, 0.4]

dropout2=[0.0, 0.1, 0.2, 0.3]

Οι υπερπαράμετροι που επιλέχθηκαν είναι: hidden\_size1=256, hidden\_size2=512, hidden\_size3=128, drop1=0.2, drop2=0.2.

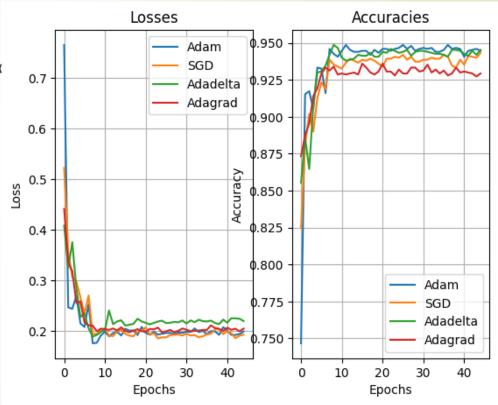
### Μέρος Δ: Τελικό Μοντέλο

 Το τελικό μοντέλο συνελικτικού νευρωνικού δικτύου που επιλέχτηκε φαίνεται παρακάτω:



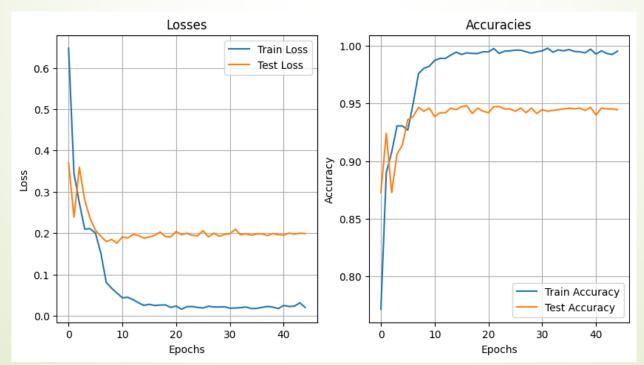
### Μέρος Δ: Επιλογή Βελτιστοποιητή

- Χρησιμοποιώντας το μοντέλο με την αρχιτεκτονική και τις υπερπαραμέτρους που βρέθηκε παραπάνω, δοκιμάστηκε ένας αριθμός από βελτιστοποιητές έτσι ώστε να βρεθεί εκείνος που ταιριάζει καλύτερα στις ανάγκες του συγκεκριμένου προβλήματος. Οι βελτιστοποιητές που δοκιμάστηκαν δημιουργήθηκαν μέσω της βιβλιοθήκης torch.optim.
- Πιο σύγκεκριμένα, εξετάστηκαν οι Adam, SGD, Adagrad, Adadelta, που αποτελούν μερικές από τις πιο δημοφιλή επιλογές. Η επίδραση τους φαίνεται δίπλα:
- Από τα διαγράμματα, βλέπουμε ότι με εξαίρεση τον Adagrad, που έχει την χειρότερη απόδοση, όλοι οι υπόλοιποι βελτιστοποιητές έχουν παρόμοια αποτελέσματα.
- Επιλέγεται ο Adam, μιας και φαίνεται να έχει την καλύτερη επίδοση και αποτελεί μια δημοφιλή επιλογή για προβλήματα κατηγοριοποίησης



### Μέρος Δ: Αποτελέσματα

- Προκειμενου να ελεγχθεί η απόδοση του μοντέλου, χωρίστηκε το datasetC
   σε train και test set με αναλογία 70-30.
- ► Accuracy στο συγκεκριμένο split: 94.47%
- 20 fold Cross-Validation Accuracy: 94.22%
- Elapsed Time: 30.033 sec

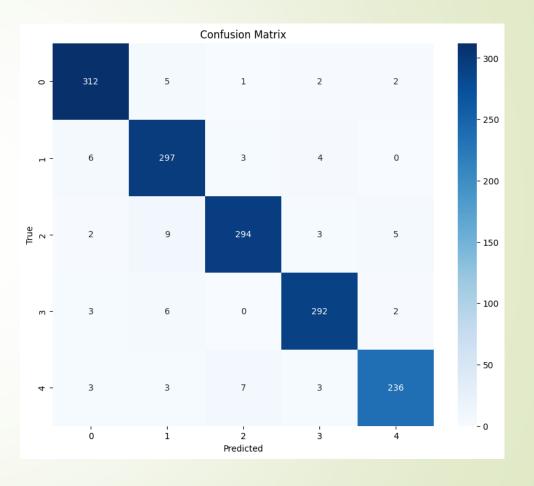


### Μέρος Δ: Συνένωση Μοντέλων

- Καθώς επιλέχθηκε η τελική αρχιτεκτονική ενός βέλτιστου συνελικτικού δικτύου, στην συνέχεια δημιουργήθηκαν 30 όμοια μοντέλα. Τα μοντέλα αυτά εκπαιδεύτηκαν στα ίδια δεδομένα. Το κάθε ένα από αυτά έδωσε τις προβλέψεις του στα δεδομένα εκτίμησής.
- Η τελική πρόβλεψη δεν λαμβάνεται επομένως από ένα μοντέλο, άλλα επιλέγεται η ετικέτα αυτή με τις περισσότερες ψήφους από τα 30 εκπαιδευμένα μοντέλα.
- Ο λόγος που υλοποιήθηκε αυτή η αρχιτεκτονική είναι:
- i. Για την αύξηση της ευρωστίας των αποτελεσμάτων, μιας και το σύνολο των προβλέψεων είναι λιγότερο ευαίσθητο σε ακραίες τιμές ή θόρυβο.
- ii. Βελτίωση της ακρίβειας.
- iii. Καταγραφή διαφορετικών αναπαραστάσεων, αφού διαφορετικά μοντέλα μπορούν να αποκτήσουν μια πιο ολοκληρωμένη κατανόηση των δεδομένων.

### Μέρος Δ: Συνένωση Μοντέλων

- Για τον έλεγχο των αποτελεσμάτων του συνενωμένου μοντέλου, χωρίστηκε το datasetC σε train και test set με αγαλογία 70-30.
- ► Accuracy στο συγκεκριμένο split: 95.40%
- Elapsed Time: 853.304 seconds



### Μέρος Δ: Συμπεράσματα

- Το γεγονός ότι η ακρίβεια του CNN μοντέλου είχε μεγάλη διαφορά σε σχέση με όλα τα υπόλοιπα που δοκιμάστηκαν, υποδεικνύει ότι το πρόβλημα αφορά κατηγοριοποίηση εικόνων 20x20.
- Οι υποθέσεις ότι η συνένωση μοντέλων θα οδηγούσε σε αύξηση της ακρίβειας επιβεβαιώθηκαν.
- Η συνένωση μοντέλων επιφέρει ένα μεγάλο trade-off μεταξύ ταχύτητα εκπαίδευσης και ακρίβειας του αλγορίθμου. Όπως είναι λογικό, η εκπαίδευση 30 μοντέλων αντί ενός, 30πλασιάζει και τον χρόνο εκπαίδευσης, κάνοντας το μοντέλο μη πρακτικό από άποψη χρόνου. Επειδή όμως η παρούσα εργασία επικεντρώνεται στην επίτευξη του υψηλότερου δυνατού accuracy, το πρόβλημα αυτό αγνοήθηκε.