

# Αναγνώριση Προτύπων

Ομάδα 28

Μαχμουτάι Έλενα

Τσουκαλά Ναταλία

# Μέρος Α

Ο στοχος του Μερους Α είναι:

- χρήση της τεχνική **Maximum Likelihood** για εκπαίδευση του ταξινομητή Bayes στα 2 παρακάτω σενάρια
  - ίδιο πίνακας συνδιασποράς για όλες τις κλάσεις
  - διαφορετικός πίνακας συνδιασποράς για όλες τις κλάσεις
- σχεδιασμός δεδομένων δοκιμής (testing)
- απεικόνιση εσφαλμένων και σωστών ταξινομήσεων
- οπτική αναπαράσταση περιοχών απόφασης για κάθε κλάση

# Τεχνική Maximum Likelihood με κοινό πίνακα συνδιασποράς για όλες τις κλάσεις

Προκειμένου να υλοποιήσουμε την τεχνική στον ταξινομητή Bayes τα βήματα που ακολουθήσαμε είναι τα εξής:

- Εισαγωγή δεδομένων από το αρχείο CSV και διαχώρηση τους σε training και testing με πιθανότητα 50/50
- Ορισμός της συνάρτησης `fit_bayes(X, y)` που είναι υπέυθυνη για τον υπολογισμό του μέσου όρου κάθε στήλης και τον κοινό πίνακα συνδιασποράς για το σύνολο X

$$\text{class\_means}_c = \frac{1}{N_c} \sum_{i=1}^{N_c} X_i$$

$$\text{common\_covariance} = \text{cov}(X)$$

- Ορισμός της συνάρτησης `predict_bayes(X, class_means, common_covariance, class_priors)` που είναι υπέυθυνη για την ταξινόμηση των data point

Υπολογισμός Πιθανότητας

Υπολογισμός εκ των υστέρων πιθανοτήτων:

Επιλογή κλάσης με την υψηλότερη εκ των υστέρων πιθανότητα

$$\text{likelihood}_k(l) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp \left( -\frac{1}{2} (l - \mu_k)^T \Sigma^{-1} (l - \mu_k) \right)$$

$$\text{posterior}_k(l) = \frac{\text{likelihood}_k(l) \times \text{prior}_k}{\sum_{j=1}^K \text{likelihood}_j(l) \times \text{prior}_j}$$

$$\text{predicted class for } l = \underset{k \in \{1, 2, \dots, K\}}{\operatorname{argmax}} \text{posterior}_k(l)$$

# Εκπαίδευση και αξιολόγηση ενός ταξινομητή Gaussian Bayes

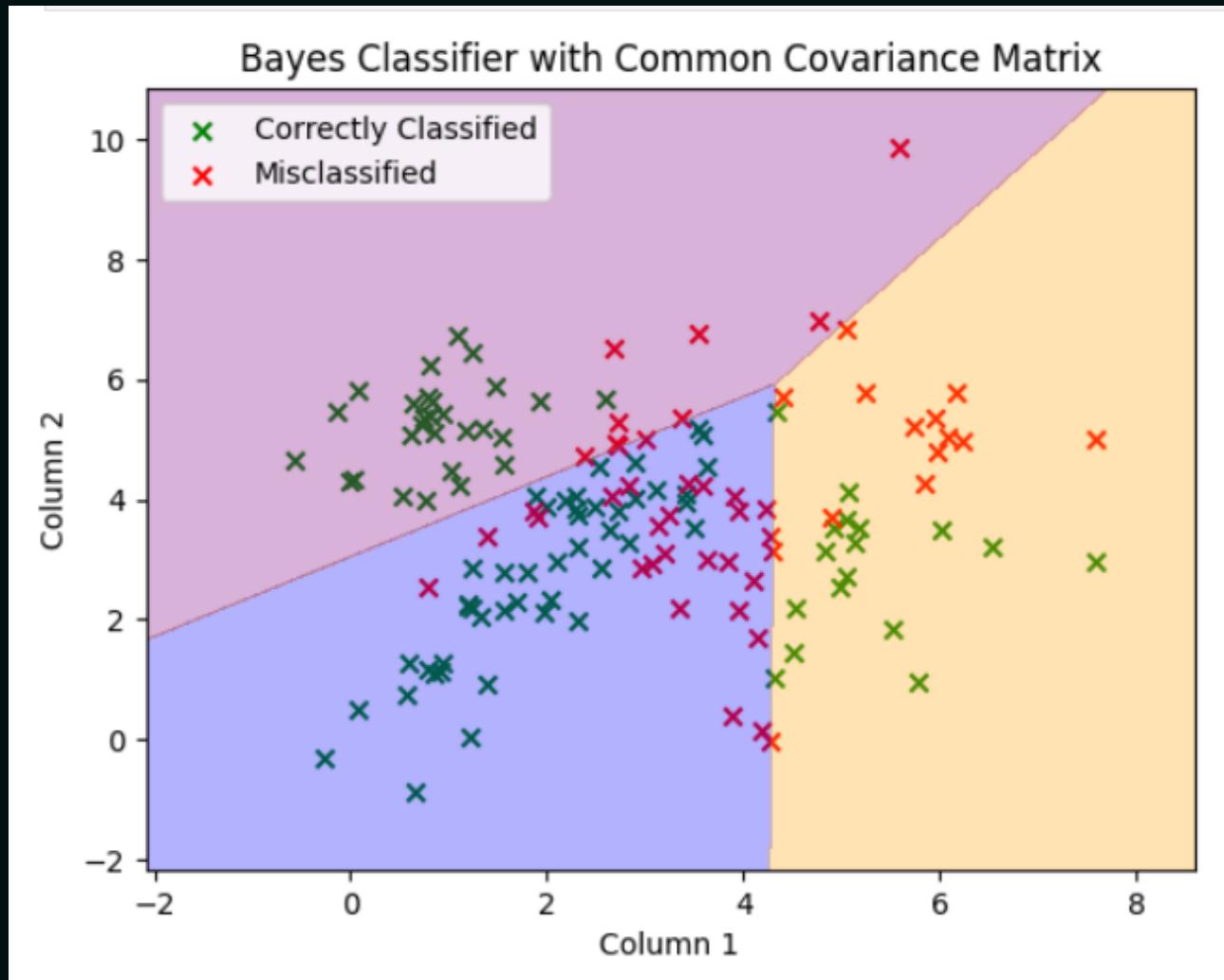
Προετοιμασία δεδομένων και εκπαίδευση μοντέλων

- Διαχωρισμός δεδομένων:
  - $X_{\text{train}}$ ,  $Y_{\text{train}}$
- Διαδικασία πρόβλεψης:
  - Συνάρτηση πρόβλεψης χρησιμοποιώντας την κατανομή Gauss και το θεώρημα του Bayes:

$$\text{likelihood}_k(x) = \mathcal{N}(x; \mu_k, \Sigma)$$

$$\text{posterior}_k(x) = \frac{\text{likelihood}_k(x) \times P(Y=k)}{\sum_j \text{likelihood}_j(x) \times P(Y=j)}$$

# Οπτικοποίηση των δεδομένων δοκιμής και των ορίων απόφασης



- Το γράφημα απεικονίζει τα σημεία δεδομένων δοκιμής από το test μαζί με τις οριακές περιοχές απόφασης που καθορίζονται από τον ταξινομητή Gaussian Bayes.
  - Κόκκινο X: λάθος ταξινομημένα δεδομένα
  - Πράσινο X: σωστά ταξινομημένα δεδομένα
- Περιοχές απόφασης και όρια:
  - Τα όρια απόφασης είναι περιοχές στο χώρο των χαρακτηριστικών όπου ο ταξινομητής μεταβαίνει από την πρόβλεψη μιας κλάσης σε μια άλλη.
  - μπλε, πορτοκαλί, μωβ
  - δημιουργούνται από τη συνάρτηση `plt.contourf`
- Υπολογισμός και ερμηνεία του σφάλματος ταξινόμησης
  - Εξίσωση για το σφάλμα ταξινόμησης:

$$\text{Classification Error} = \frac{\text{Number of Misclassified Samples}}{\text{Total Number of Samples}}$$

# Τεχνική μέγιστης πιθανοφάνειας με διαφορετικό πίνακα συνδιακύμανσης

Για την τεχνική αυτή εφαρμόσαμε έναν Γκαουσιανό ταξινομητή Bayes με διαφορετικούς πίνακες συνδιακύμανσης για κάθε κλάση. Χρησιμοποιήσαμε την συνάρτηση `def fit_bayes_different_covariance(X, y)`

- Class Means

$$\mu_c = \frac{1}{N_c} \sum_{i:y_i=c} X_i$$

- Class Covariances

$$\Sigma_c = \frac{1}{N_c-1} \sum_{i:y_i=c} (X_i - \mu_c)(X_i - \mu_c)^T$$

- Class Priors

$$P(Y = c) = \frac{N_c}{N}$$

# Πρόβλεψη με χρήση διαφορετικών πινάκων συνδιακύμανσης

Η συνάρτηση `def** predict_bayes_different_covariance` χρησιμοποιείται για την πραγματοποίηση προβλέψεων χρησιμοποιώντας έναν ταξινομητή Gaussian Bayes με διαφορετικούς πίνακες συνδιακύμανσης για κάθε κλάση

- Επαναλαμβάνει πάνω από τις περιπτώσεις εισόδου και τις αληθείς ετικέτες

- Υπολογίζει πιθανότητες για κάθε κλάση  $L_c(x) = \frac{1}{\sqrt{(2\pi)^k |\Sigma_c|}} \exp\left(-\frac{1}{2}(x - \mu_c)^T \Sigma_c^{-1} (x - \mu_c)\right)$

- Υπολογίζει εκ των υστέρων πιθανότητες:  $P(c|x) = L_c(x) \times P(c)$

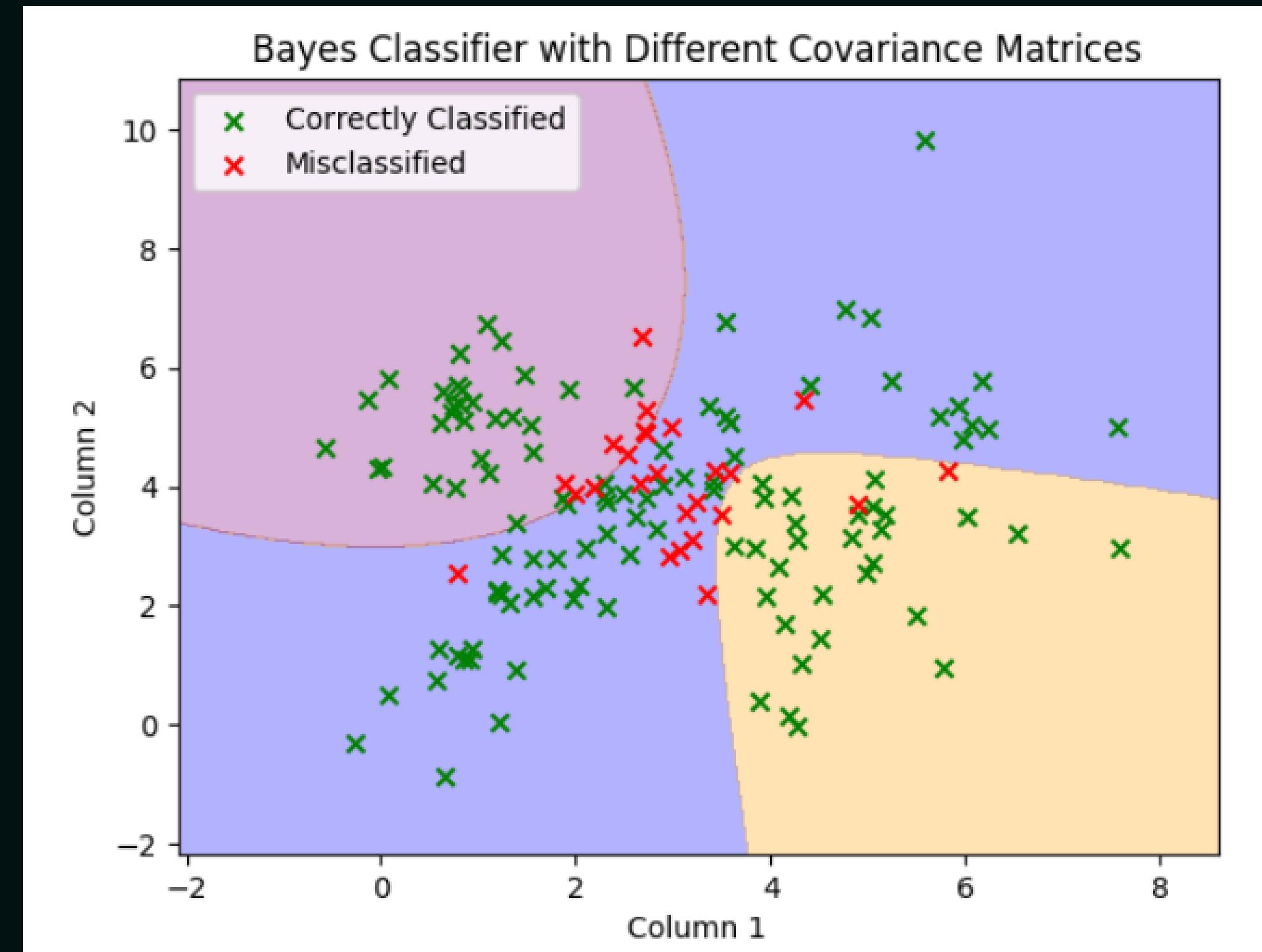
- Προβλέπει την κλάση με τη μεγαλύτερη εκ των υστέρων πιθανότητα  $\hat{y} = \underset{c}{\operatorname{argmax}} P(c|x)$
- Εντοπίζει εσφαλμένα ταξινομημένες περιπτώσεις

Αυτή η συνάρτηση επιστρέφει δύο πίνακες: ένας με τις προβλεπόμενες ετικέτες κλάσης για κάθε δείγμα στο  $X$  και μια άλλη με τους δείκτες/ετικέτες των λανθασμένα ταξινομημένων δειγμάτων.

- Εκπαιδευούμε τον ταξινομητή Naive Bayes με διαφορετικούς πίνακες συνδιακύμανσης
- Κάνουμε προβλέψεις στο σύνολο των δοκιμών και υπολογίζουμε το **μέσο σφάλμα ταξιμόνησης: 17.86%**

# Οπτικοποίηση των δεδομένων δοκιμής και των ορίων απόφασης

Τα όρια απόφασης ενός ταξινομητή Naive Bayes με διαφορετικούς πίνακες συνδιακύμανσης



## Συμπεράσματα - Μέρος Α

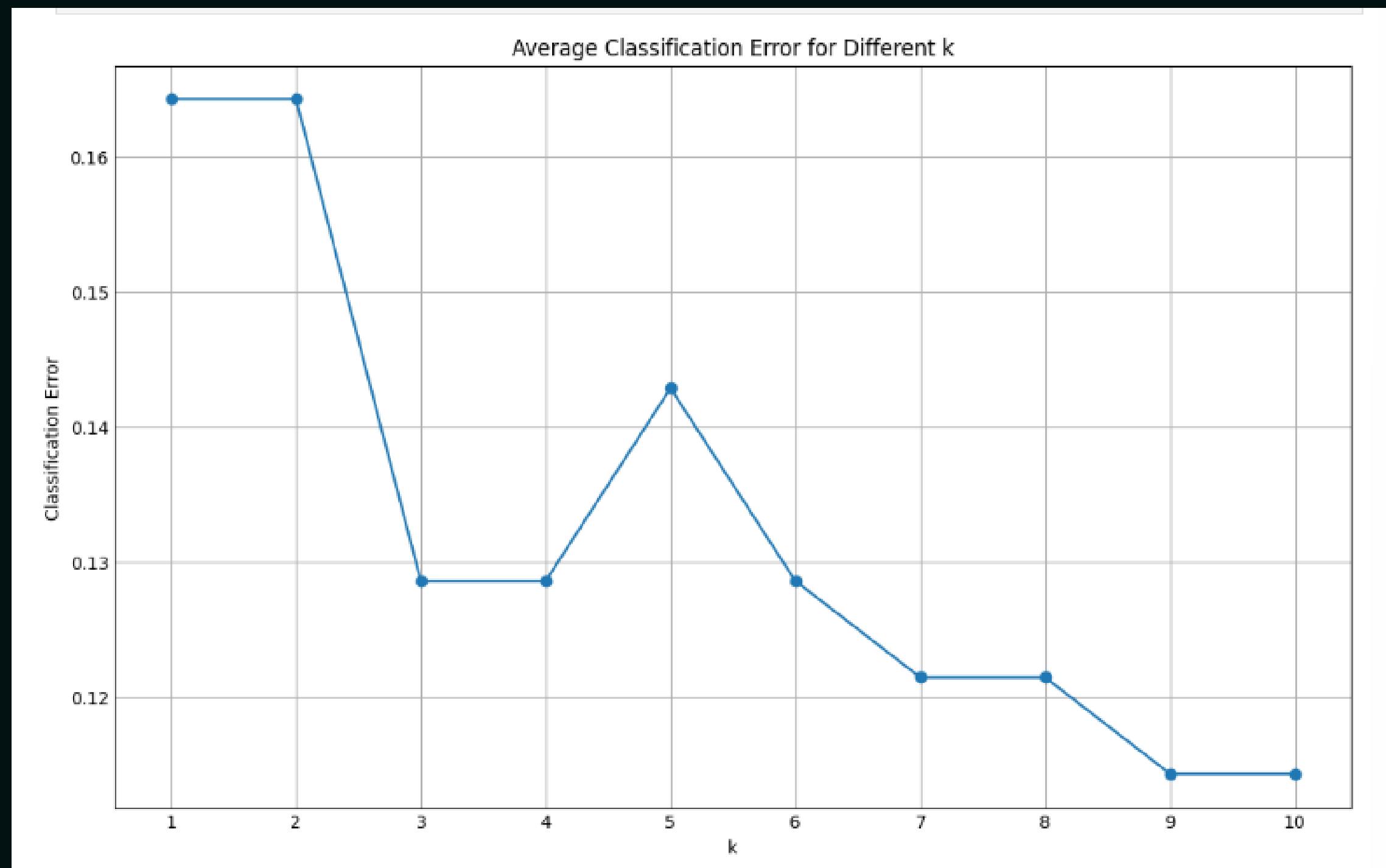
- Η τεχνική Μέγιστης Πιθανότητας με κοινό πίνακα συνδιακύμανσης οδήγησε σε μέσο σφάλμα ταξινόμησης 35,00%
- Η τεχνική Μέγιστης Πιθανότητας με διαφορετικούς πίνακες συνδιακύμανσης πέτυχε σημαντικά χαμηλότερο μέσο σφάλμα ταξινόμησης 17,86%.
- Συγκρίνοντας τις 2 προσεγγίσεις παρατηρούμε ότι η τεχνική με διαφορετικές μήτρες συνδιακύμανσης υπερτερεί της μεθόδου με κοινή μήτρα συνδιακύμανσης σε αυτή την περίπτωση και επιδεικνύει καλύτερη κατανόηση των λεπτομερειών των μοτίβων των δεδομένων.

## Μέρος Β.

Στόχος του δεύτερου μέρους της εργασίας είναι η εκπαίδευση ενός KNN ταξινομητή χρησιμοποιώντας το ίδιο dataset με το Μέρος Α και να συγκρίνουμε τα δεδομένα.

- Κάνουμε import τα δεδομένα και τα χωρίζουμε σε test / train με 50%-50% αναλογία
- Ορίζουμε τη συνάρτηση `train_and_evaluate_knn(k, X_train, X_test, y_train, y_test)` για την εκπαίδευση ενός ταξινομητή K-NN και τον υπολογισμό του μέσου σφάλματος στα δεδομένα δοκιμής.
- Και στη συνέχεια σχεδιάζουμε τα μέσα σφάλματα ταξινόμησης για  $k = 1, \dots, 10$  στο ακόλουθο γράφημα με βάση την έξοδο της συνάρτησης `train_and_evaluate`.

# Οπτικοποίηση των δεδομένων δοκιμής και των ορίων απόφασης



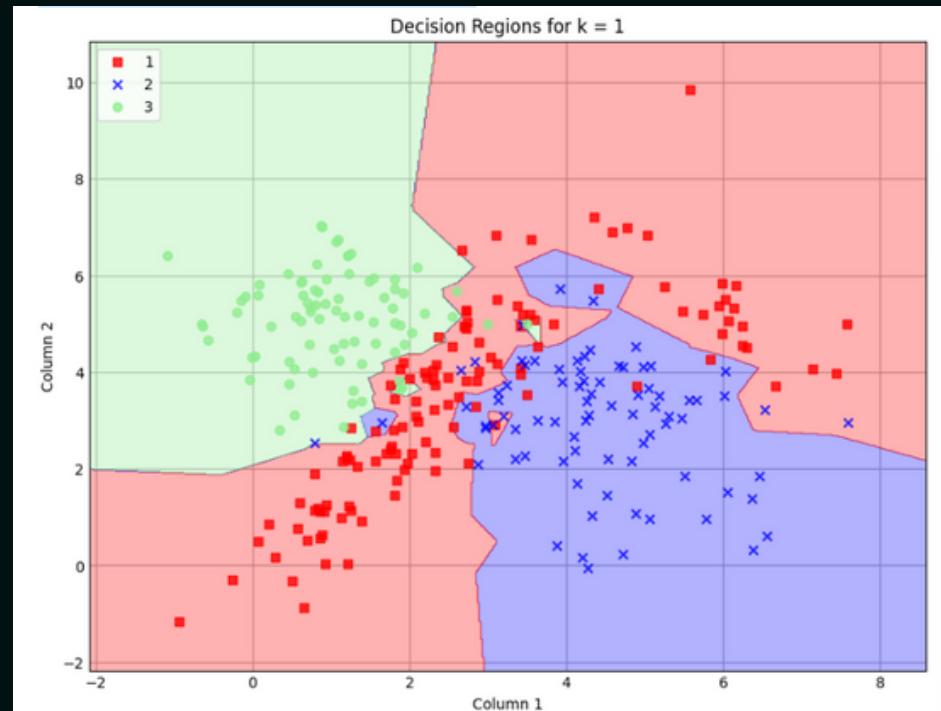
- Το μέσο σφάλμα μειώνεται καθώς αυξάνεται η τιμή  $k$  μέχρι την τιμή  $k = 9$
- Για  $k=9$  το μέσο σφάλμα παραμένει σταθερό για λίγο.
- Η μεγαλύτερη ανατροπή της τιμής συμβαίνει για  $k = 2$  έως  $k = 3$ .

Ορίζουμε τη λειτουργία `knn_plot_decision_regions` που είναι υπεύθυνη για την απεικόνιση των περιοχών απόφασης για τις διάφορες κατηγορίες δεδομένων μας.

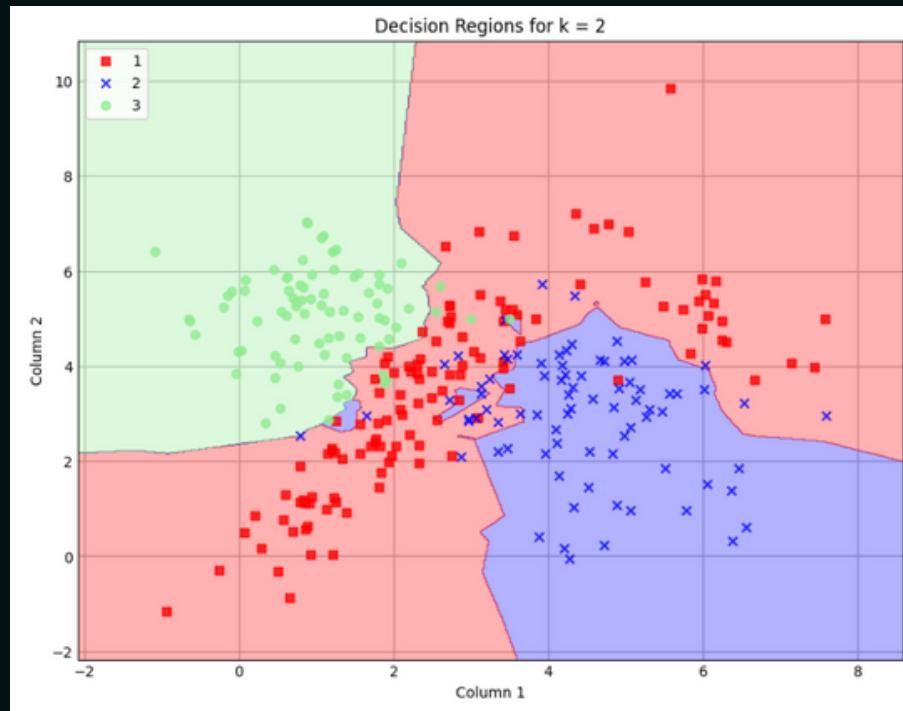
- Εμφάνιση των περιοχών απόφασης των k-κοντινότερων γειτόνων (kNN)
- `knn_plot_decision_regions` :
  - Οπτικοποιεί τις περιοχές απόφασης για την ταξινόμηση kNN.
  - Δημιουργεί ένα πλέγμα πλέγματος με χρωματική κωδικοποίηση με βάση τις προβλέψεις kNN.
  - Επικαλύπτει τα σημεία δεδομένων με ξεχωριστά χρώματα και δείκτες για κάθε κλάση.
- `plot_data_into_regions` :
  - Εμφανίζει το μέσο σφάλμα ταξινόμησης για ένα καθορισμένο k.
  - Καλεί την `knn_plot_decision_regions` για να σχεδιάσει τις περιοχές απόφασης.
  - Προσθέτει λεπτομέρειες γραφικής παράστασης, όπως τίτλο, ετικέτες αξόνων και υπόμνημα.
- **Χρήση στην ανάλυση:**
  - Εφαρμόζεται επαναληπτικά για τιμές k από 1 έως 10.
  - Επιτρέπει τη σύγκριση των ορίων απόφασης και της ακρίβειας σε διαφορετικές τιμές k.
  - Παρέχει οπτική και ποσοτική εικόνα της απόδοσης του ταξινομητή kNN.

# Γραφίματα από k=1 έως k=5

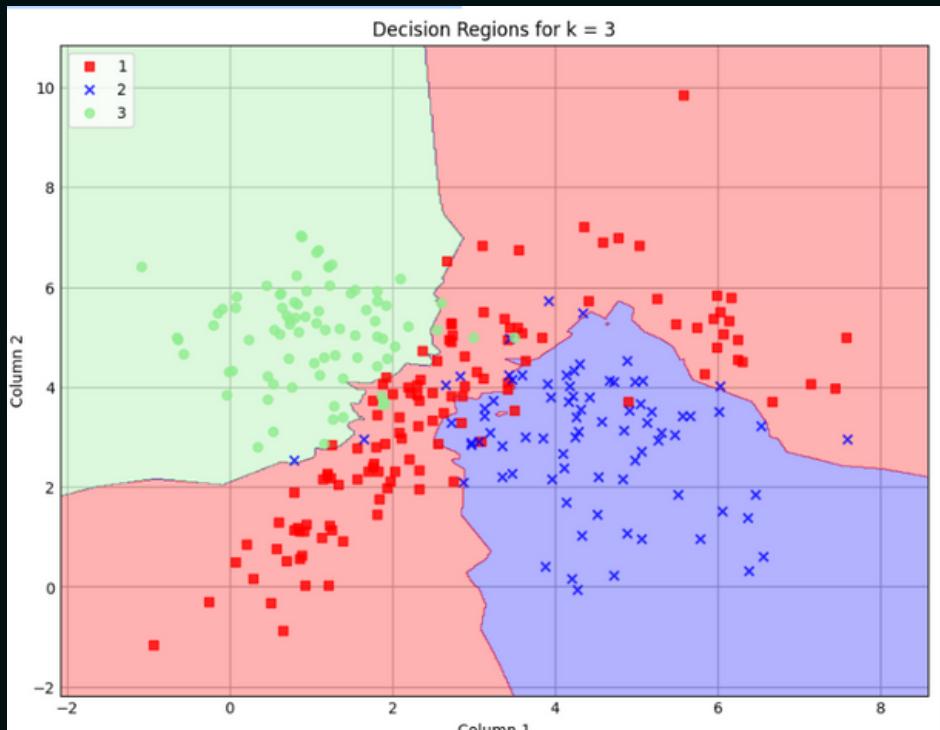
Average Classification Error for k = 1: **16.43%**



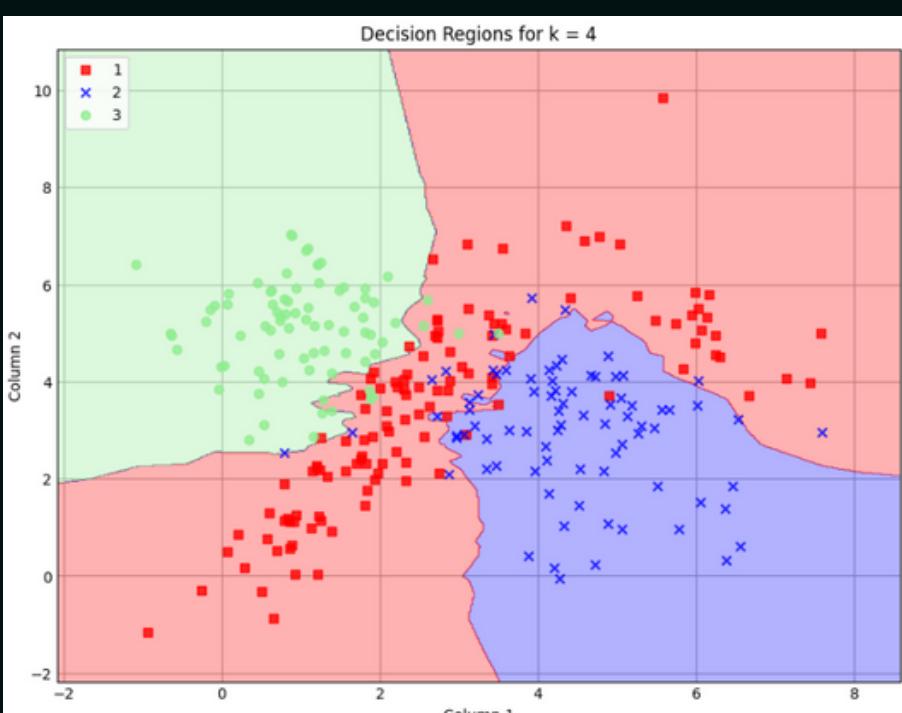
Average Classification Error for k = 2: **16.43%**



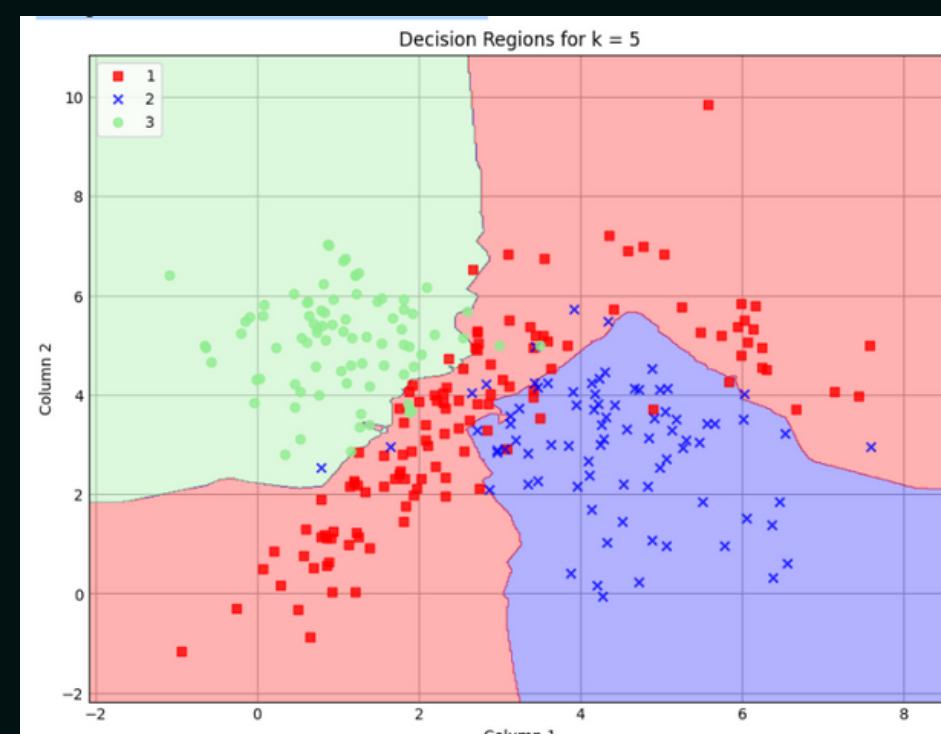
Average Classification Error for k = 3: **12.86%**



Average Classification Error for k = 4: **12.86%**

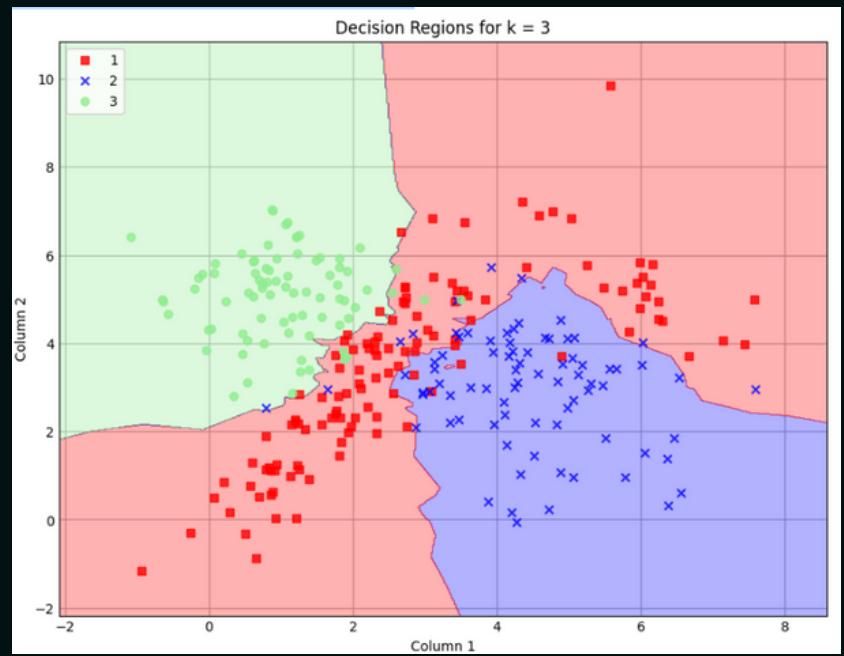


Average Classification Error for k = 5: **14.29%**

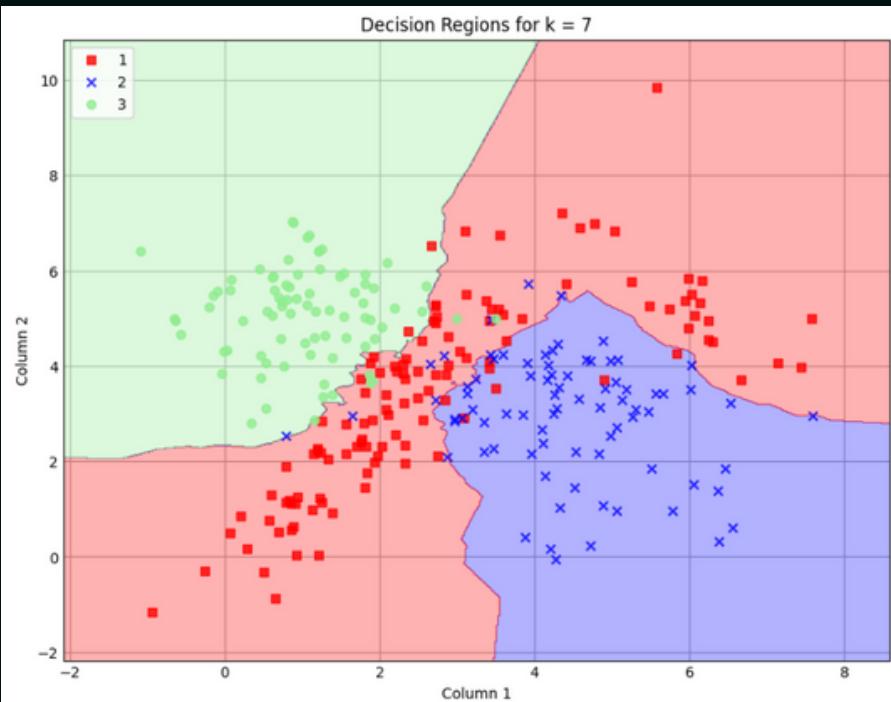


# Γραφίματα από k=6 έως k=10

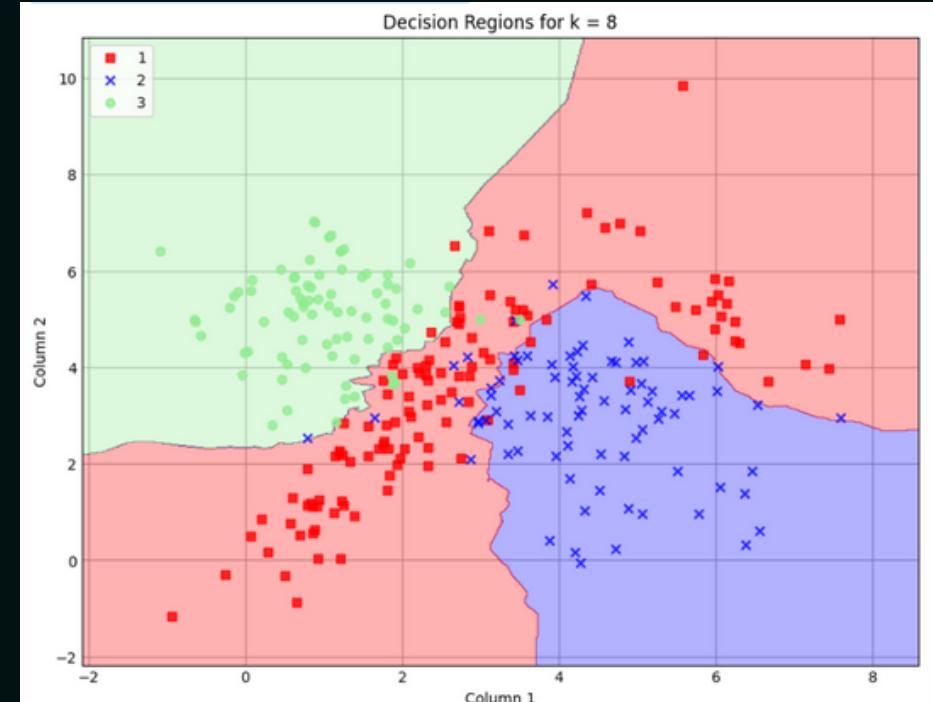
Average Classification Error for k = 6: **12.86%**



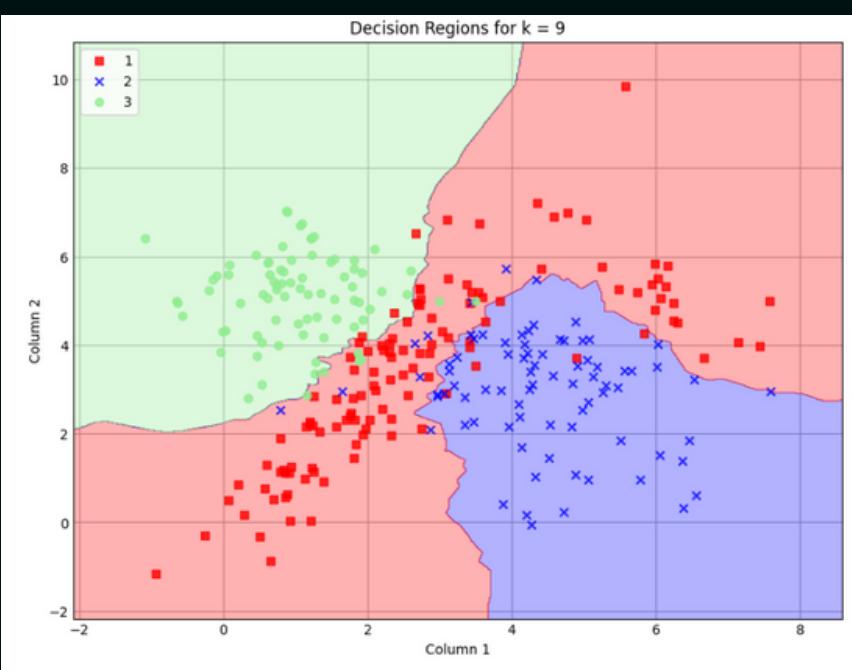
Average Classification Error for k = 7: **12.14%**



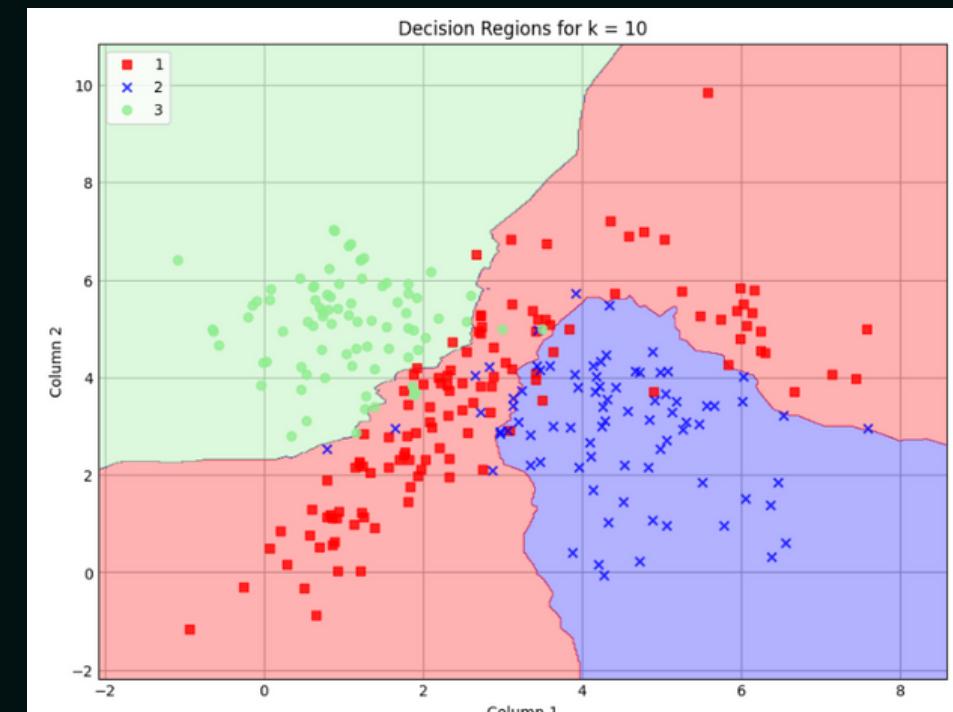
Average Classification Error for k = 8: **12.14%**



Average Classification Error for k = 9: **11.43%**



Average Classification Error for k = 10: **11.43%**



# Συμπεράσματα:

- **Τάσεις σφαλμάτων:** Το μέσο σφάλμα ταξινόμησης γενικά μειώνεται καθώς αυξάνεται το  $k$  (αριθμός γειτόνων), υποδεικνύοντας βελτιωμένη ακρίβεια με περισσότερους γείτονες.
- **Πλάτωμα σφαλμάτων:** Το ποσοστό σφάλματος φτάνει στο πλατώ  $k = 9$ , υποδηλώνοντας περιορισμένη αύξηση της ακρίβειας με υψηλότερες τιμές  $k$ .
- **Αξιοσημείωτη βελτίωση:** Σημαντική μείωση των σφαλμάτων παρατηρείται κατά την αύξηση του  $k$  από 2 σε 3, αναδεικνύοντας τον αντίκτυπο του αριθμού των γειτόνων σε χαμηλότερες τιμές  $k$ .
- **Οπτικοποίηση της περιοχής απόφασης:** Τα οπτικά διαγράμματα καταδεικνύουν πώς εξελίσσονται τα όρια απόφασης με διαφορετικές τιμές  $k$ , βοηθώντας στην κατανόηση του διαχωρισμού κλάσεων του αλγορίθμου K-NN.
- **Βέλτιστη τιμή  $k$ :** Μια βέλτιστη τιμή  $k$  γύρω στο 9 ή 10 προσφέρει μια ισορροπία μεταξύ του ποσοστού σφάλματος και της πολυπλοκότητας του μοντέλου για αυτό το σύνολο δεδομένων.
- **Συνέπεια σφαλμάτων:** Συγκεκριμένες τιμές  $k$  (1, 2, 4, 6) παρουσιάζουν σταθερά ποσοστά σφάλματος, που ενδεχομένως αντικατοπτρίζουν τα χαρακτηριστικά του συνόλου δεδομένων.
- **Πρακτικές επιπτώσεις:** Η επιλογή ενός κατάλληλου  $k$  είναι ζωτικής σημασίας για την εξισορρόπηση της ακρίβειας και της υπολογιστικής απόδοσης στους ταξινομητές K-NN.

# Σύγκριση Bayesian Classifier και K-Nearest Neighbors (K-NN)

## Bayesian Classifier:

- **Θεωρητική βάση:** Bayes με κανονικές κατανομές.
- **Υποθέσεις:** Διαφορετική συνδιακύμανση επιτρέπει μοναδικές συνδιακυμάνσεις.
- **Όρια απόφασης:** Συνήθως ομαλά, γραμμικά ή τετραγωνικά.
- **Πολυπλοκότητα:**
  - Η εκπαίδευση περιλαμβάνει τον υπολογισμό των μέσων, των συνδιακυμάνσεων και των προτεραιοτήτων.
  - Γρηγορότερες προβλέψεις μετά την εκπαίδευση.
- **Απόδοση:**
- **Κοινή συνδιακύμανση:** Υψηλότερο σφάλμα (35,00%).
- **Διαφορετικές συνδιακυμάνσεις:** 86%, καλύτερη διάκριση κλάσεων.

## K-Nearest Neighbors (K-NN):

- Μη παραμετρική: Δεν υπάρχουν υποθέσεις σχετικά με την κατανομή των δεδομένων.
- Βασισμένη σε περιπτώσεις: Ταξινομεί με βάση τις πλησιέστερες επισημειωμένες περιπτώσεις.
- Όρια απόφασης: Πιο σύνθετα, όχι απαραίτητα ομαλά.
- Πολυπλοκότητα:
  - Ελάχιστη εκπαίδευση, καθώς αποθηκεύει δεδομένα.
  - Υψηλή πολυπλοκότητα πρόβλεψης λόγω υπολογισμών απόστασης.
- Απόδοση:
  - Ποικίλλει ανάλογα με το "k". Συγκεκριμένες τιμές 'k' επηρεάζουν σημαντικά το ποσοστό σφάλματος.
  - Η βέλτιστη επιλογή "k" εξισορροπεί την ευαισθησία στο θόρυβο και την υπερβολική εξιμάλυνση

# Συμπέρασμα σύγκρισης

- **Καταλληλότητα:** K-NN για σύνολα δεδομένων χωρίς τέτοιες υποθέσεις.
- **Απόδοση:** Ειδικά όταν το βέλτιστο "k" δεν είναι σαφές ή σε μεγάλα σύνολα δεδομένων.
- **Εφαρμογή:** Η επιλογή εξαρτάται από το συμβιβασμό μεταξύ της υπολογιστικής απόδοσης και της ανάγκης για μοντελοποίηση βάσει υποθέσεων.

## Μέρος Γ

Σκοπός του τρίτου μέρους του παραδοτέου είναι η χρήση ενός ταξινομητή SVM για την ταξινόμηση των δεδομένων από το αρχείο datasetCTest.csv.

- Ακολουθούμε την ίδια τακτική φόρτωσης δεδομένων με παραπάνω.
- Εισάγουμε την βιβλιοθήκη 'scikit-learn' που περιέχει τον γραμμικό ταξινομητή SVM προκειμένου να κάνουμε προβλέψεις στα δεδομένα και να προσαρμόσουμε τα δεδομένα στον ταξινομητή μας

# Αξιολόγηση της απόδοσης του ταξινομητή σε δεδομένα εκπαίδευσης και δοκιμής

## 1. Προβλέψεις και υπολογισμός σφαλμάτων:

- Χρησιμοποιούμε έναν γραμμικό ταξινομητή για να δημιουργήσετε προβλέψεις για δεδομένα εκπαίδευσης και δοκιμής.
- Training Data Prediction
- Testing Data Predictions

## 2. Σφάλμα ταξινόμησης για κάθε σημείο δεδομένων:

- Training Data
- Testing Data

## 3. Υπολογισμός μέσου σφάλματος:

Training Data:

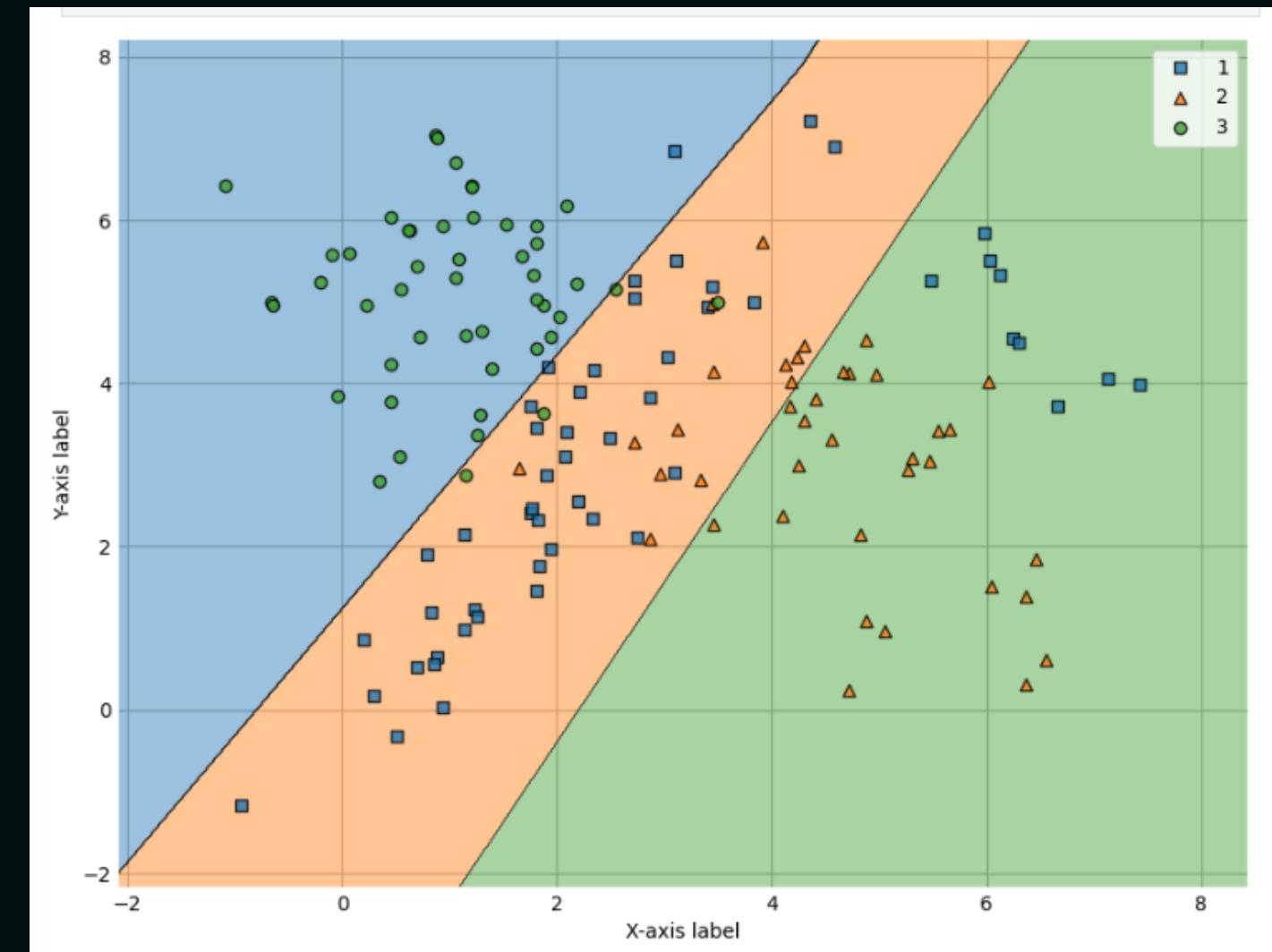
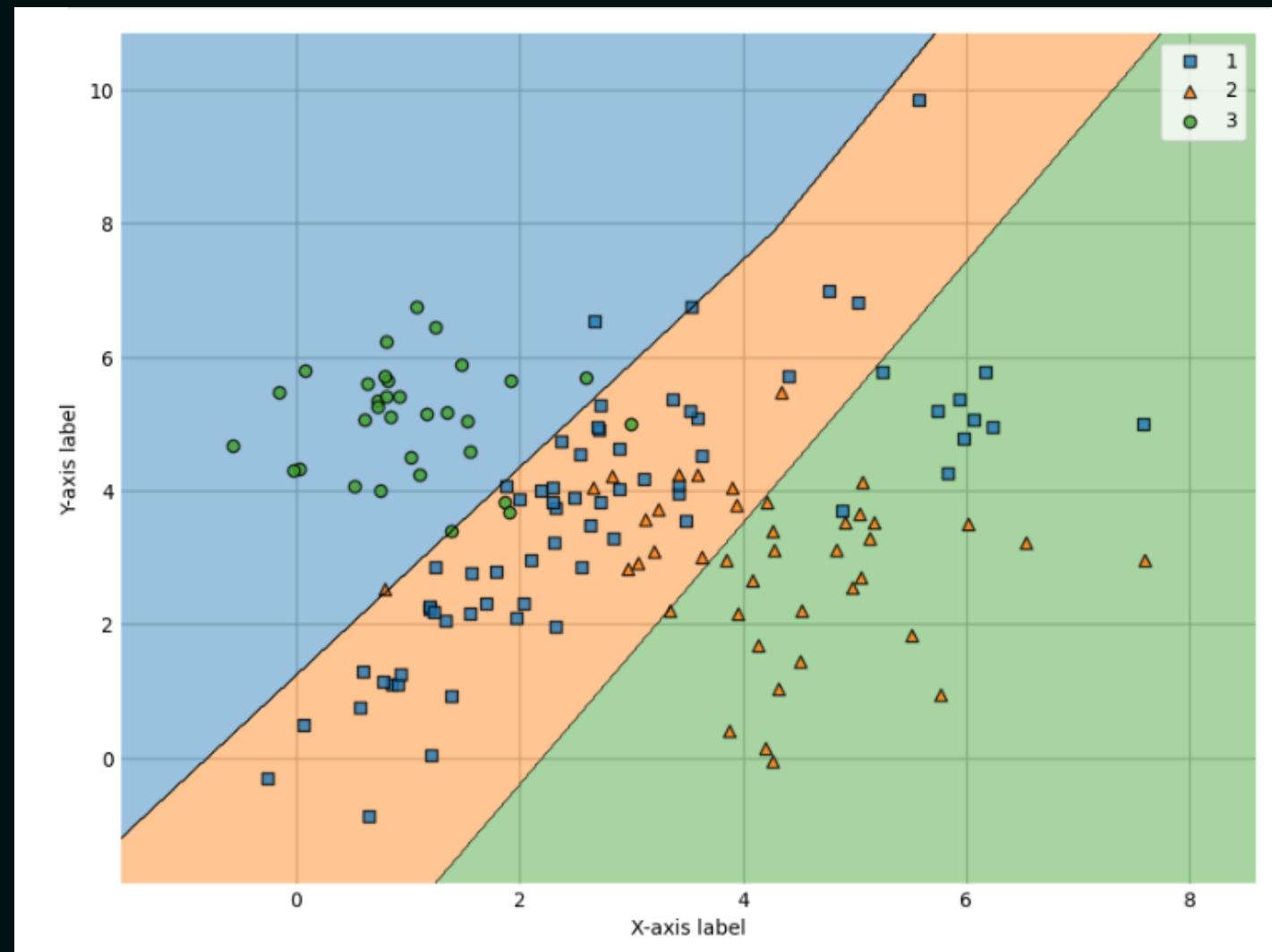
- $\text{average\_error} = 0.19285714285714287$

Testing Data:

- $\text{average\_error} = 0.21428571428571427$

# Οπτικοποίηση της απόδοσης του ταξινομητή με τα οριακά διαγράμματα αποφάσεων

- `plot_func`: Μια συνάρτηση για την οπτικοποίηση δεδομένων σε 2D με όρια απόφασης.
- Χρησιμοποιεί την `plot_decision_regions` για τη σχεδίαση.



# Χρήση:

## Σε δεδομένα εκπαίδευσης:

- `plot\_func(X\_train, y\_train, linear\_classifier)`.
- Παρατηρήσεις:
  - Επιτυχής ταξινόμηση των περισσότερων στοιχείων εκπαίδευσης.
  - Εσφαλμένη ταξινόμηση κοντά στα όρια απόφασης.
  - Αποτελεσματικός ορισμός των διανυσμάτων απόφασης από τον ταξινομητή.

## Στα δεδομένα δοκιμής:

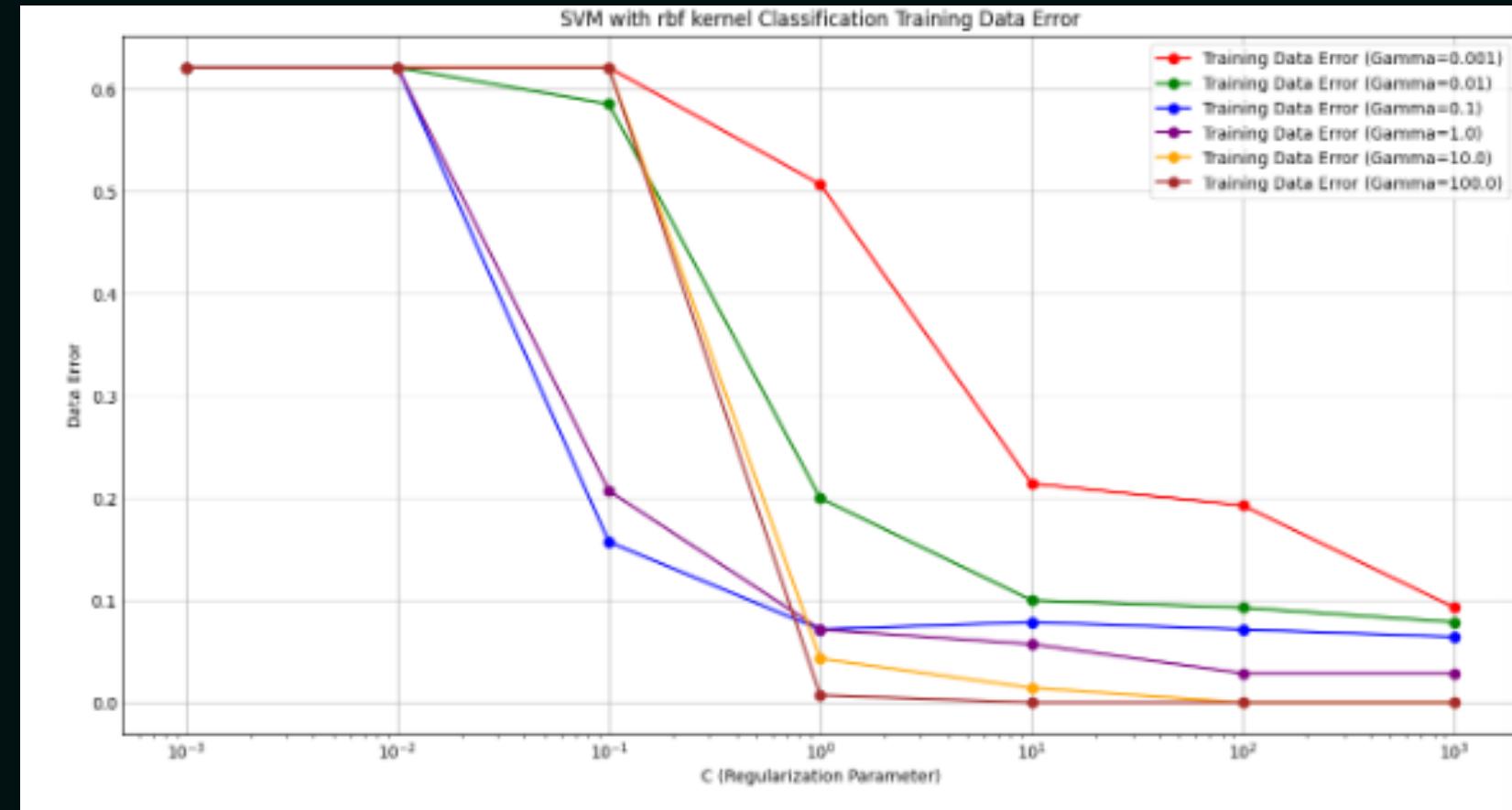
- `plot\_func(X\_test, y\_test, linear\_classifier)`.
- Παρατηρήσεις:
  - Απεικόνιση των επιδόσεων σε αφανή δεδομένα.
  - Τα όρια απόφασης δείχνουν το διαχωρισμό των κλάσεων.
  - Ακριβείς ταξινομήσεις σε σωστές περιοχές- λανθασμένες ταξινομήσεις σε λανθασμένες περιοχές.
  - Το ποσοστό σφάλματος ~21,4% αναδεικνύει την αποτελεσματικότητα του μοντέλου και τις περιοχές για βελτίωση.

# Αξιολόγηση ταξινομητή SVM με πυρήνα RBF

## Συνάρτηση `train_and_evaluate_classifier`:

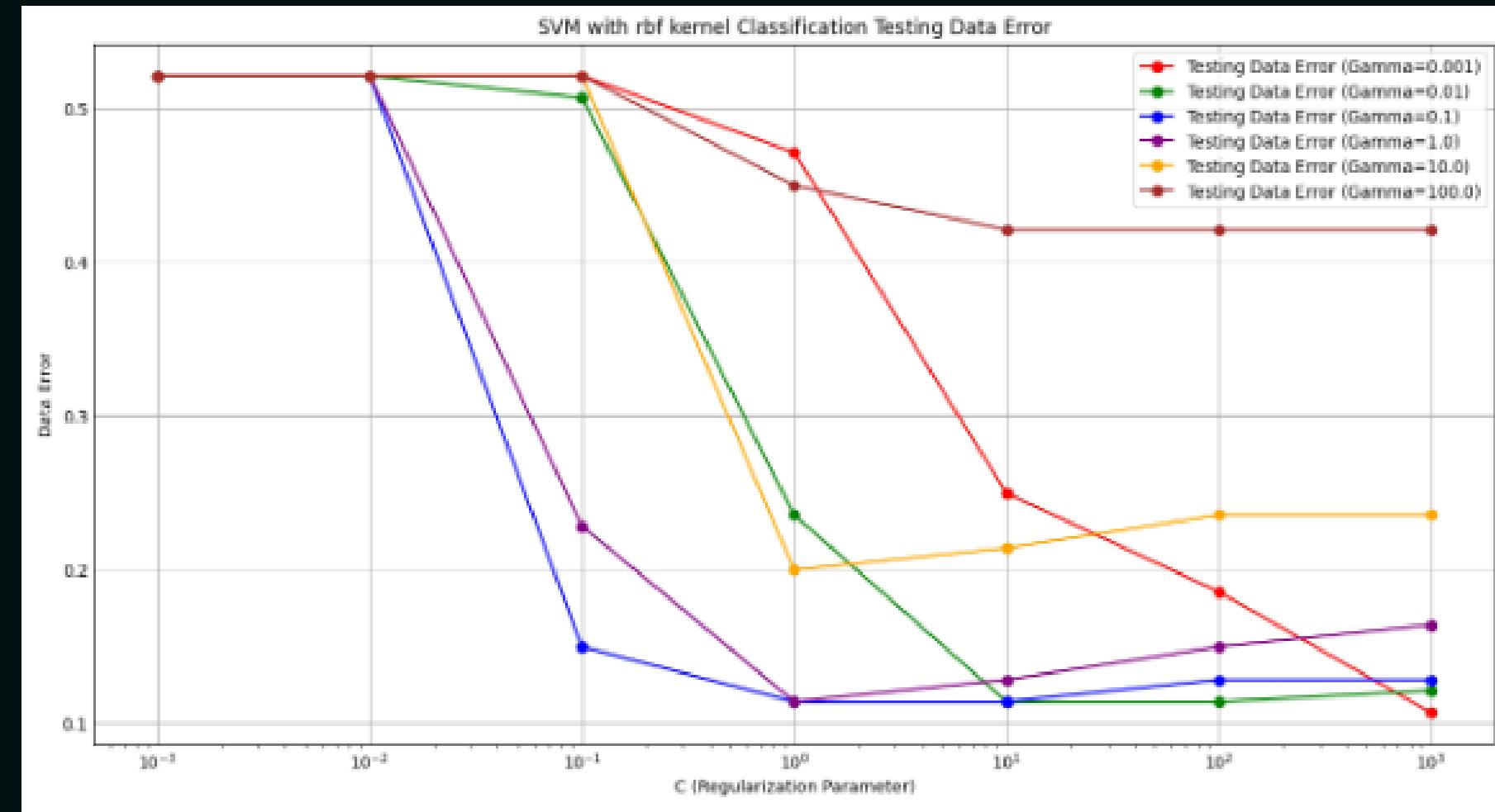
- Σκοπός Εκπαίδευση και αξιολόγηση του ταξινομητή SVM
- Εκπαίδευση:
  - Προσαρμόζει τον ταξινομητή με εικόνες και ετικέτες εκπαίδευσης
- Πρόβλεψη και υπολογισμός σφάλματος:
  - Training Data:
    - Προβλέπει τις ετικέτες για τις εικόνες εκπαίδευσης
    - Υπολογίζει τα σφάλματα ταξινόμησης
    - Υπολογίζει το μέσο σφάλμα
  - Testing Data:
    - Προβλέπει ετικέτες για εικόνες δοκιμής
    - Υπολογίζει τα σφάλματα ταξινόμησης
    - Υπολογίζει το μέσο σφάλμα

# Εικονικοποίηση των αποτελεσμάτων του ταξινομητή SVM με πυρήνα RBF



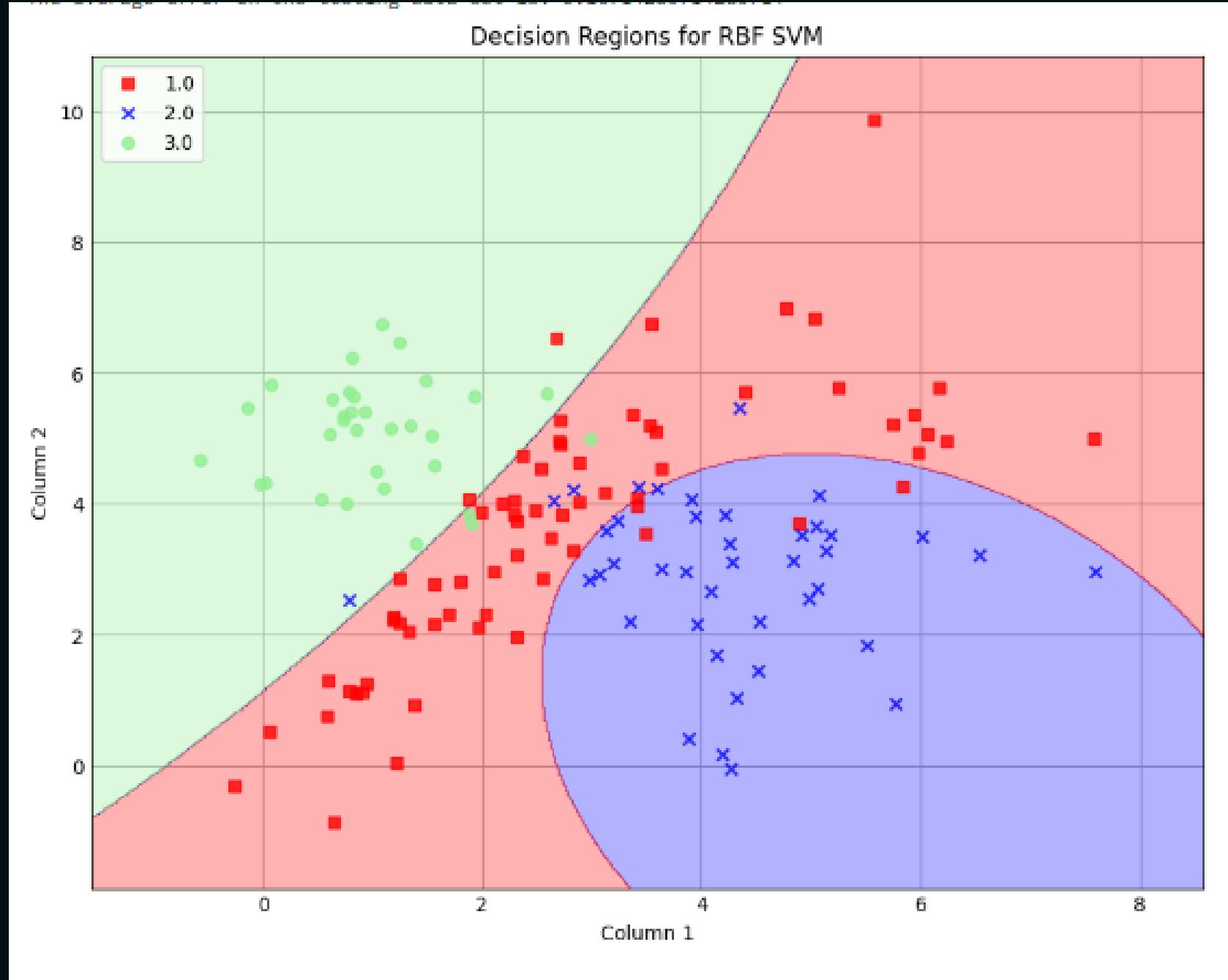
- **Ανασκόπηση γραφήματος:** SVM με πυρήνα RBF.
- **Διαφοροποίηση σφαλμάτων με  $C$ :** Αναλύει πώς μεταβάλλεται το μέσο σφάλμα δοκιμής σε διαφορετικές τιμές  $C$  σε κάθε ρύθμιση γ
- **Υψηλές τιμές γάμμα (100, 10, 1):**
  - Σταθερό ποσοστό σφάλματος σε διάφορες τιμές  $C$
  - Δείχνει αντίσταση στην υπερπροσαρμογή.
- **Χαμηλές τιμές γάμμα (<1):**
  - Μεγαλύτερη ευαισθησία στις αλλαγές  $C$  στην απόδοση του μοντέλου.
  - Σε χαμηλότερες τιμές  $C$ : Μικρότερο γ αποδίδει καλύτερη ακρίβεια.
  - Σε υψηλότερα  $C$  (>100): Μεγαλύτερο γ παρουσιάζει βελτιωμένη ακρίβεια.
- **Αλληλεπίδραση των  $C$  και Γamma:**
  - Αποδεικνύει την πολύπλοκη σχέση μεταξύ  $C$  και γ που επηρεάζει τη γενίκευση του μοντέλου.
  - Επισημαίνει τη σημασία της ρύθμισης και των δύο υπερπαραμέτρων για βέλτιστη απόδοση

# Εικονικοποίηση των αποτελεσμάτων του ταξινομητή SVM με πυρήνα RBF



- **Βελτίωση με την αύξηση του C:**
  - Σταθερή βελτίωση της απόδοσης καθώς αυξάνεται η τιμή C.
  - Ισχύει για όλες τις ρυθμίσεις γ.
- **Υψηλές τιμές γάμμα (100, 10, 1):**
  - Ταχεία επίτευξη μέγιστης ακρίβειας.
  - Διατήρηση της υψηλής ακρίβειας σε διάφορες τιμές C.
- **Χαμηλές τιμές γάμμα (<1):**
  - Υψηλή ακρίβεια σε χαμηλότερες τιμές C.
  - Αύξηση της ακρίβειας με μείωση της γ.
- **Σύγκλιση πέραν του C=100:**
  - Παρόμοια υψηλά επίπεδα ακρίβειας για όλες τις ρυθμίσεις γ.
  - Υποδηλώνει μειωμένη επίδραση της γ στην ακρίβεια της εκπαίδευσης πέραν της C = 100 .

# Διαγράμματα περιοχής απόφασης:



- Χρηση τη συνάρτηση `svm_plot_decision_regions`
- Οπτικοποιήση στον τρόπο με τον οποίο ο ταξινομητής RBF SVM διαχωρίζει τις διάφορες κλάσεις στο σύνολο δεδομένων δοκιμής.

# Συνοπτική ανάλυση του ταξινομητή SVM

- **Γραμμικός SVM:**
  - Μέσο σφάλμα εκπαίδευσης: 19.29%.
  - Μέσο σφάλμα δοκιμής: 21.43%.
  - Αποτελεσματικός σε βασικά σενάρια, αλλά περιορισμένος με πολύπλοκα δεδομένα.
- **RBF Kernel SVM:**
  - $C = 1000$ ,  $\gamma = 0,001$
  - Σφάλμα εκπαίδευσης: 9,29%: 10.71%.
  - Υπερέχει του γραμμικού SVM στο χειρισμό μη γραμμικών προτύπων.
- **Εμπειρίες οπτικοποίησης:**
  - Το RBF SVM εμφανίζει πιο προσαρμοστικά, σύνθετα όρια απόφασης σε σύγκριση με το γραμμικό SVM.
- **Επιπτώσεις υπερπαραμέτρων:**
  - Η ακρίβεια του RBF SVM είναι ιδιαίτερα ευαίσθητη στις τιμές  $C$  και  $\gamma$ , τονίζοντας την ανάγκη για ακριβή ρύθμιση των υπερπαραμέτρων.

# Σύγκριση SVM με Maximum Likelihood και K-NN

## Πολύπλοκοτητα μοντέλου:

- Bayesian ταξινομητής με ML είναι απλούστερος από το RBF SVM.
- Το RBF SVM απαιτεί ακριβή ρύθμιση των C και γ.

## Απόδοση:

- Η RBF SVM υπερτερεί έναντι του Bayesian ταξινομητή με κοινή συνδιακύμανση.
- Οι επιδόσεις πλησιάζουν περισσότερο όταν συγκρίνονται με τον Bayesian με διαφορετικές συνδιακυμάνσεις.
- Ο SVM προσφέρει μεγαλύτερη ευελιξία χωρίς ισχυρές υποθέσεις κατανομής.

## Σύγκριση με K-NN:

### Ερμηνευσιμότητα του μοντέλου:

- Το K-NN είναι διαισθητικό, ταξινομώντας με βάση την εγγύτητα των γειτόνων.
- Η λήψη αποφάσεων του RBF SVM είναι πιο αφηρημένη, περιλαμβάνοντας απεικονίσεις υψηλών διαστάσεων

## Απόδοση:

- Το RBF SVM υπερέχει έναντι του K-NN στο χειρισμό πολύπλοκων, μη γραμμικών δεδομένων.
- Ο K-NN είναι λιγότερο αποτελεσματικός με μη ομοιόμορφα δεδομένα και περίπλοκα όρια.

## Υπολογιστική αποδοτικότητα:

- Το SVM προβλέπει πιο αποτελεσματικά από το K-NN.
- Η φάση εκπαίδευσης του SVM, ιδίως με πυρήνα RBF, απαιτεί εκτεταμένο συντονισμό υπερπαραμέτρων.

## Μέρος Δ

### Ανάπτυξη Αλγορίθμου Ταξινόμησης

- Επιλογή Μεθόδου Ταξινόμησης για εκπαίδευση του **datasetC.csv** ως training set
- Εφαρμογή του Ταξινομητή σε test set **datasetCTest.csv** χωρίς Ετικέτες
- Παραγωγή Διανύσματος **labels28.npy** που περιέχει τα διάνυσμα των ετικετών του test set

Για την ανάπτυξη του ταξινομητή επιλέχθηκε η υλοποίηση ενός **Νευρωνικού Δικτύου**:

- Επιλέχθηκε λόγω της ικανότητάς του να μάθει και να εξάγει πολύπλοκες σχέσεις από τα δεδομένα.
- Η χρήση διαφόρων αρχιτεκτονικών, όπως πολυεπίπεδα νευρωνικά δίκτυα, είχε ως στόχο τον εντοπισμό της βέλτιστης απόδοσης

## Τεχνολογίες που χρησιμοποιήθηκαν:

- TensorFlow & Keras χρησιμοποιήθηκαν για την υλοποίηση του νευρωνικού δικτύου.
- Pandas & NumPy χρησιμοποιήθηκαν για τη φόρτωση, επεξεργασία και ανάλυση των δεδομένων.
- Neptune χρησιμοποιήθηκε για την αναπαράσταση και παρακολούθηση της εκπαίδευσης του μοντέλου.

## Γιατί Αυτές οι Τεχνολογίες;

- Ευελιξία & Αποτελεσματικότητα
  - Οι τεχνολογίες αυτές είναι κατάλληλες για εκπαίδευση μοντέλων μηχανικής μάθησης και ανάλυση δεδομένων.
- Κοινότητα & Υποστήριξη
  - Ευρεία κοινότητα χρηστών και συνεχής υποστήριξη για την επίλυση προβλημάτων.

## Πρώτο Πείραμα

Η διαδικασία που ακολουθήσαμε για το πρώτο πείραμα είναι η εξής:

- **Διαχωρισμός Δεδομένων:**
  - Διαχωρισμός του dataset σε σύνολα εκπαίδευσης και ελέγχου.
- **Κατασκευή Μοντέλου:**
- **Δομή του Μοντέλου:**
  - Χρήση δύο for loop για τη δοκιμή διαφορετικών μεγεθών hidden layer [10, 64, 128, 256]
  - Εξερεύνηση συνδυασμών μεγεθών για βέλτιστη απόδοση.
  - Χρήση Dense layers με συναρησεις ενεργοποίησης relu και softmax
- **Εκπαίδευση Μοντέλου:**
  - Εκπαίδευση με χρήση του Adam optimizer και απώλειας sparse categorical cross-entropy για 10 epochs.
- **Αξιολόγηση Μοντέλου:**
  - Αξιολόγηση απόδοσης στο test set.
  - Εκτύπωση των αποτελεσμάτων, συμπεριλαμβανομένης της απώλειας και της ακρίβειας.

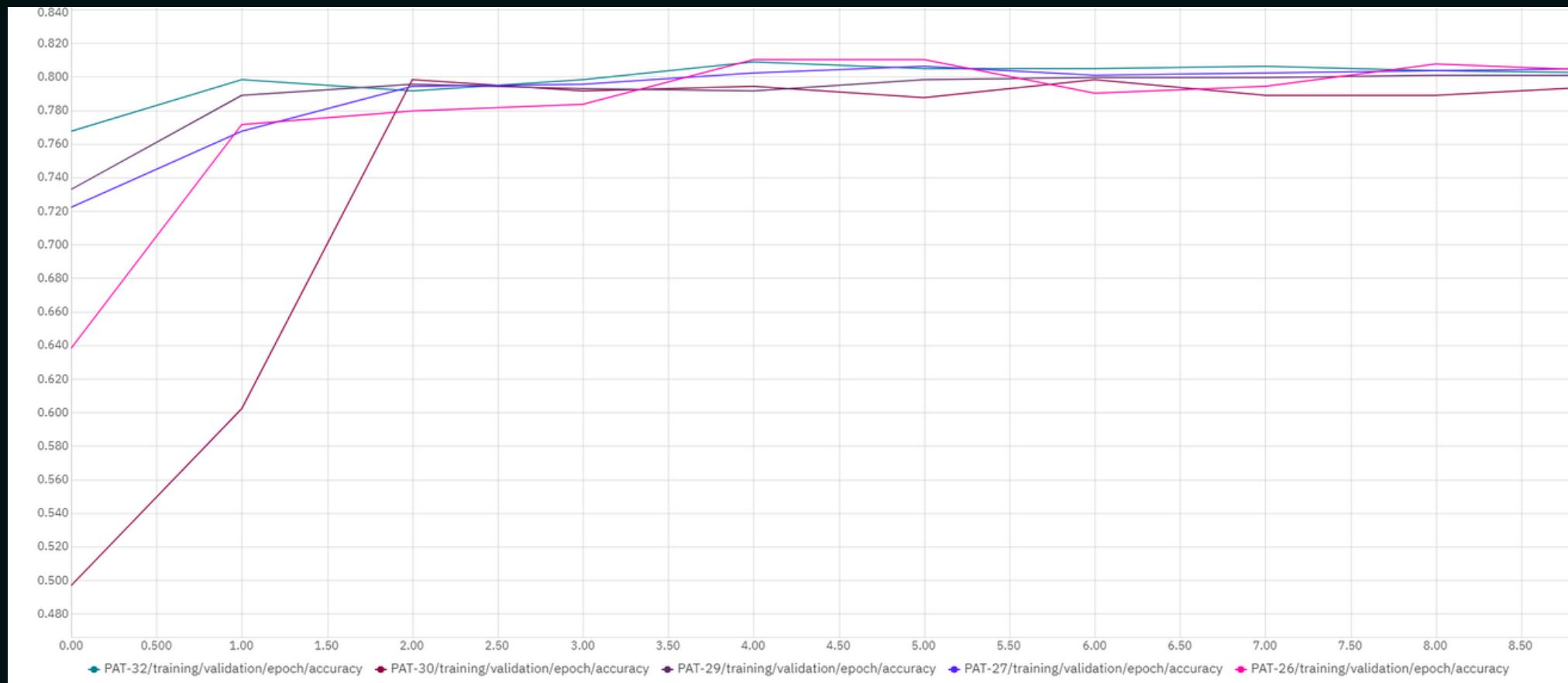
# Αποτελέσματα του πρώτου πειράματος

A ● ⚙ Id	# layer1_size	# layer2_size	Last training/validation/epoch/accuracy ↑↓	Last training/validation/epoch/loss	Last training/train/epoch/accuracy
● ⚡ PAT-27	128	64	0.805333	0.687598	1
● ⚡ PAT-26	128	10	0.804	0.685581	1
● ⚡ PAT-32	256	128	0.802667	0.755802	1
● ⚡ PAT-29	128	256	0.801333	0.809526	1
● ⚡ PAT-30	256	10	0.794667	0.732789	0.998857
● ⚡ PAT-33	256	256	0.789333	0.752884	1
● ⚡ PAT-24	64	128	0.785333	0.805287	1
● ⚡ PAT-31	256	64	0.784	0.746685	1
● ⚡ PAT-18	10	10	0.784	0.649608	0.899429
● ⚡ PAT-28	128	128	0.782667	0.816534	1
● ⚡ PAT-25	64	256	0.774667	0.896073	1

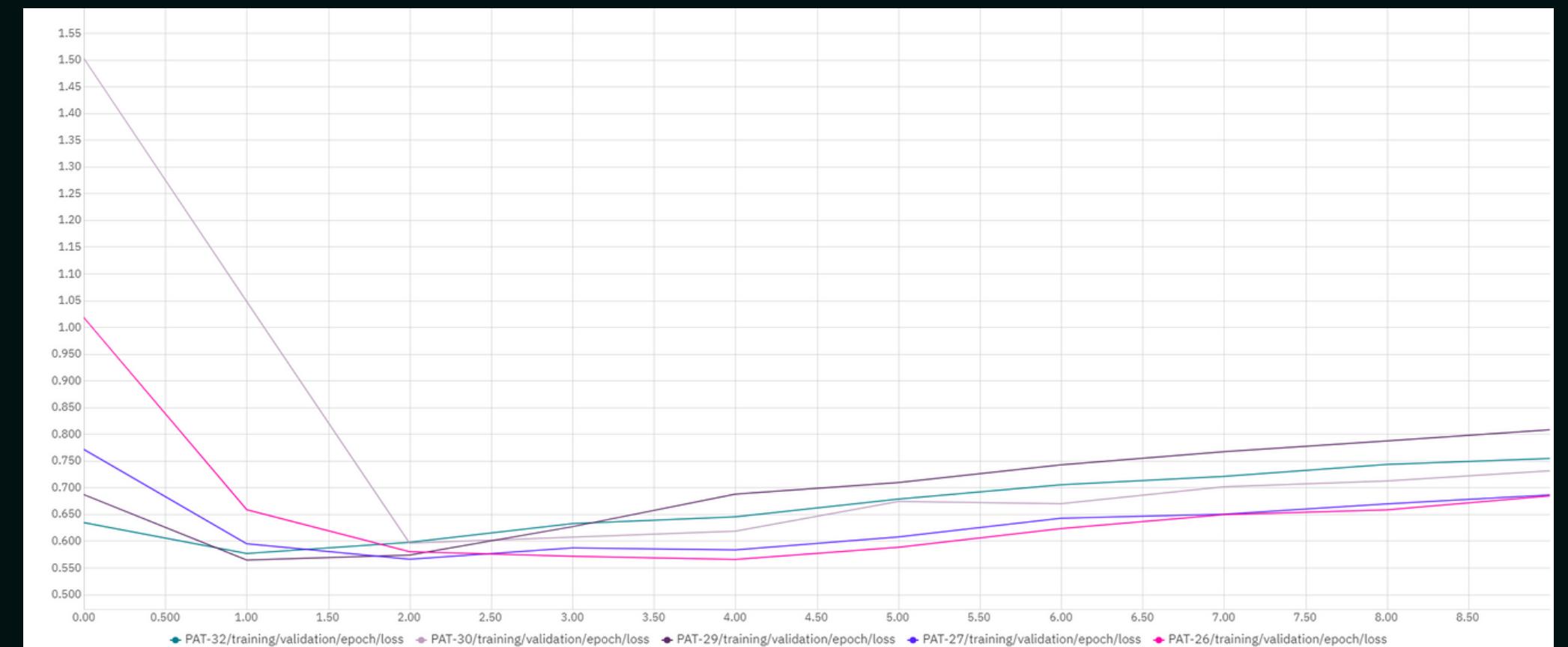
Παρατηρήσεις:

- Μικρές μεταβολές στις τιμές για διαφορετικές τιμές των hidden layers
- Υψηλότερη απόδοση όταν οι τιμές των hidden layer δεν έχουν μεγάλη διαφορά μεταξύ τους
- Δεν αναγνωρίζουμε κάποιο μοτίβο

## Epoch Accuracy Graph on the Validation Data



## Epoch Loss Graph on the Validation Data



Παρατηρούμε overfitting στα δεδομένα μας

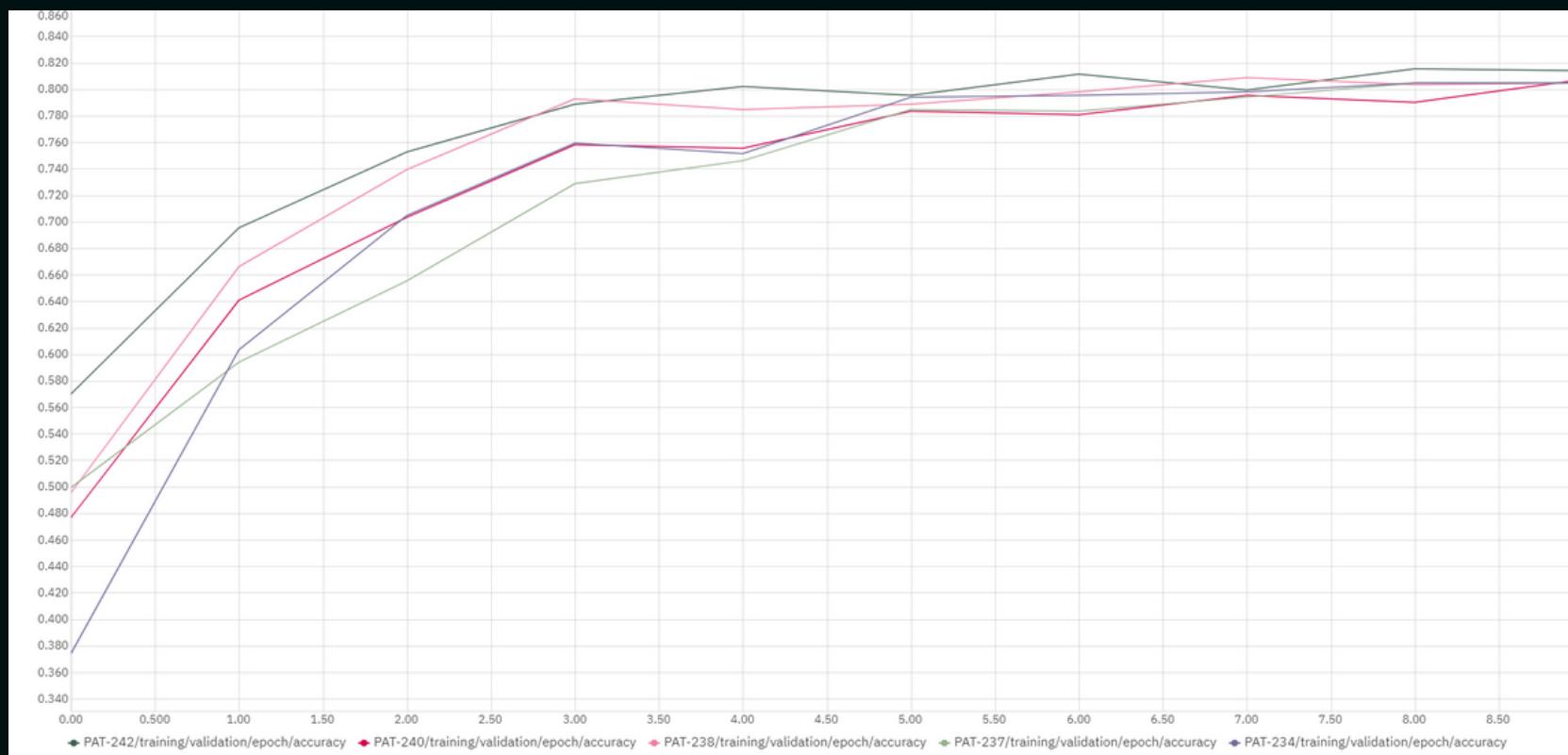
## **Δεύτερο Πείραμα: Εκπαίδευση ενός Νευρωνικού Δικτύου με τρία εσωτερικά επίπεδα**

- **Σκοπός του Δεύτερου Πειράματος:**
  - Ανάλυση της επίδρασης της αύξησης του αριθμού των επιπέδων στην ακρίβεια του μοντέλου.
  - Χρήση Νευρωνικού Δικτύου με τρία επίπεδα για εκτίμηση της επίδρασης αυτής.
- **Διαδικασία:**
  - Ίδια με το Πρώτο πείραμα
  - Προσθήκη Dropout layers για αποφυγή overfitting

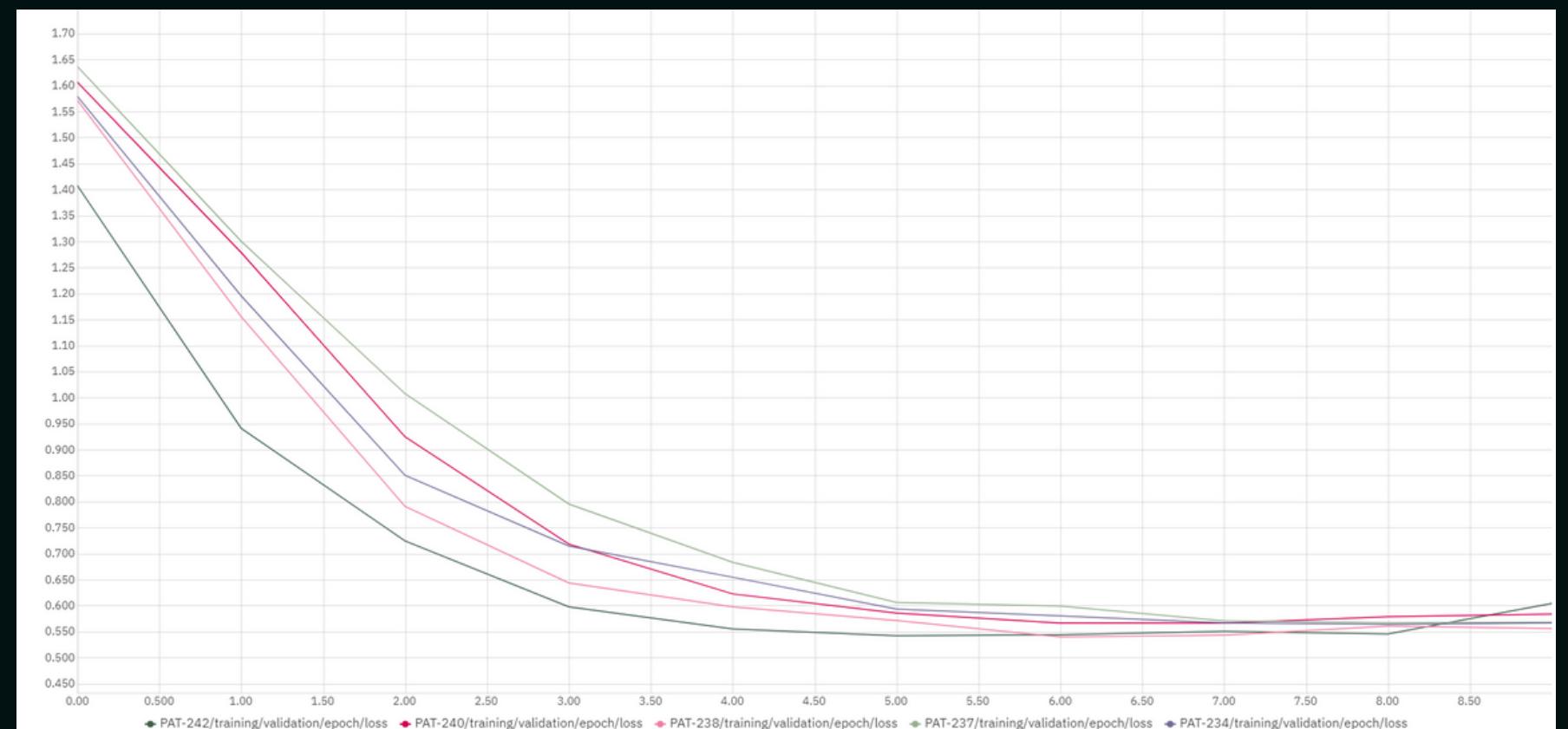
# Αποτελέσματα του δευτέρου πειράματος

A	Last ...ing/validation/epoch/accuracy	Last training/validation/epoch/loss	# layer1_size	# layer2_size	# layer3_size
Id					
PAT-242	0.814667	0.60578	256	256	256
PAT-240	0.808	0.585217	256	256	64
PAT-237	0.805333	0.569583	256	128	128
PAT-238	0.805333	0.557548	256	128	256
PAT-234	0.805333	0.568297	256	64	256
PAT-224	0.8	0.578574	128	256	64
PAT-229	0.798667	0.595598	256	10	128
PAT-233	0.797333	0.560306	256	64	128
PAT-218	0.794667	0.554412	128	64	256
PAT-225	0.794667	0.621037	128	256	128
PAT-241	0.792	0.626118	256	256	128
PAT-205	0.792	0.592814	64	128	128
PAT-206	0.790667	0.586314	64	128	256

## Epoch Accuracy Graph on the Validation Data



## Epoch Loss Graph on the Validation Data



## Παρατηρούμε:

- σταθεροποίηση του διαγράμματος
- βελτίωση του overfitting

## Τρίτο Πείραμα: Εκπαίδευση ενός CNN Νευρωνικού Δικτύου

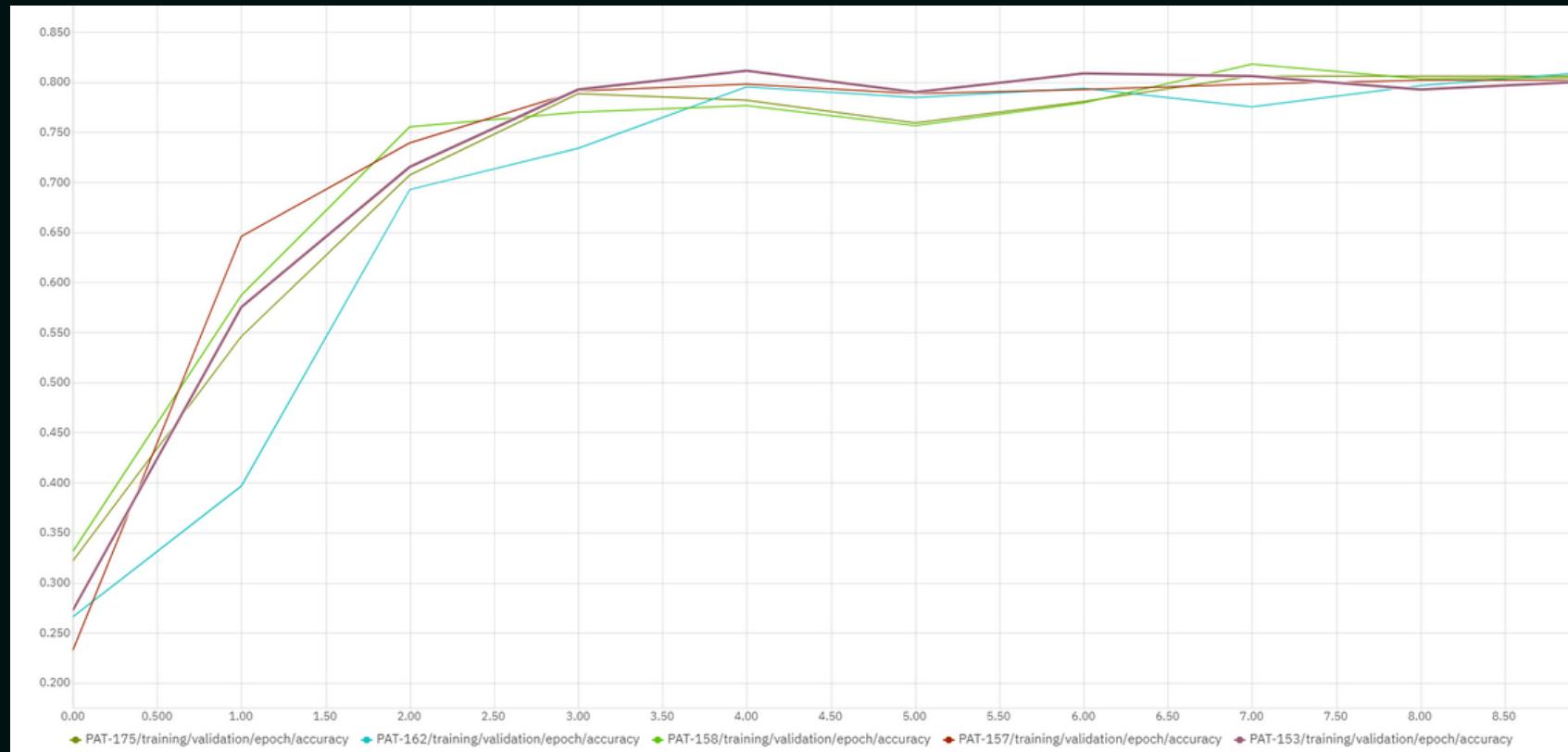
- **Σκοπός του Τρίτου Πειράματος:**
  - Διερεύνηση ενός CNN δικτύου προκειμένου να αναγνωρίζει μοτίβα στο σύνολο δεδομένων
- **Διαδικασία:**
  - Προσπάθεια αναγνώρισης βέλτιστης αρχιτεκτονικής
  - Αξιολόγηση απόδοσης και παρακολούθηση πειράματος μέσω του Neptune

# Αποτελέσματα του τρίτου πειράματος

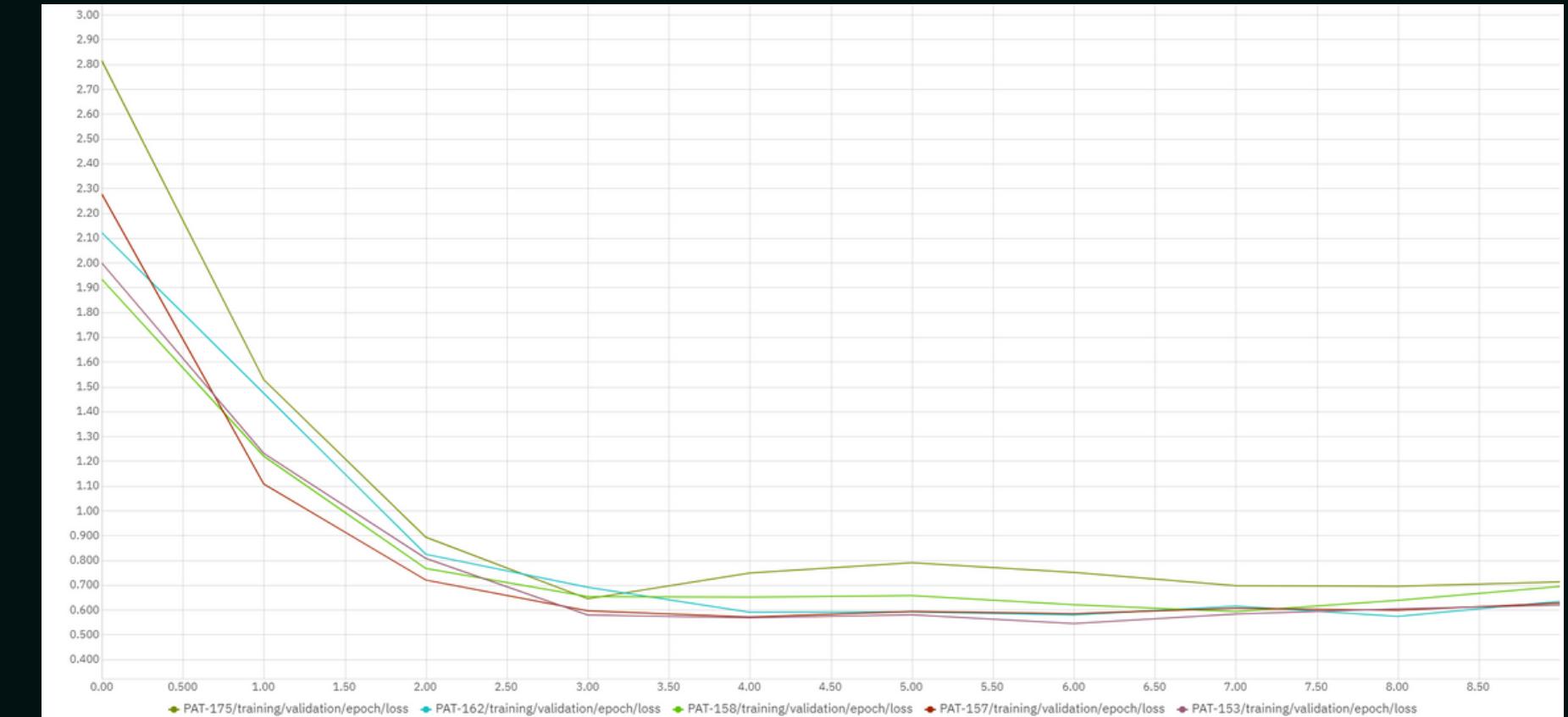
A	Last ...ing/validation/epoch/accuracy ↑	Last training/validation/epoch/loss	# layer1_size	# layer2_size	# layer3_size	
Id	①	②	③	④	⑤	
•	⑥ PAT-162	0.810667	0.637254	128	64	128
•	⑦ PAT-175	0.806667	0.715464	256	128	256
•	⑧ PAT-158	0.805333	0.697518	64	256	64
•	⑨ PAT-157	0.802667	0.630248	64	128	256
•	⑩ PAT-153	0.801333	0.622425	64	64	128
•	⑪ PAT-173	0.798667	0.678161	256	128	64
•	⑫ PAT-163	0.798667	0.660392	128	64	256
•	⑬ PAT-174	0.797333	0.685947	256	128	128
•	⑭ PAT-167	0.796	0.723071	128	256	64
•	⑮ PAT-156	0.796	0.620328	64	128	128
•	⑯ PAT-170	0.790667	0.624301	256	64	64
•	⑰ PAT-171	0.790667	0.670136	256	64	128

- Οι τιμές δεν έχουν μεγάλη διαφορά με προηγουμένως
- Δεν φαίνεται να υπάρχουν μοτίβα στα δεδομένα

## Epoch Accuracy Graph on the Validation Data



## Epoch Accuracy Graph on the Validation Data



- Γρηγορότερη σταθεροποίηση διαγράμματος
- Περισσότερες μεταβολές συγκριτικά με το δεύτερο πείραμα

## Βελτίωση μοντέλου

- Επιλογή μοντέλου με καλύτερη απόδοση
- Δοκιμή διαφόρων υπερπαραμέτρων με σκοπό την αύξηση της ακρίβειας

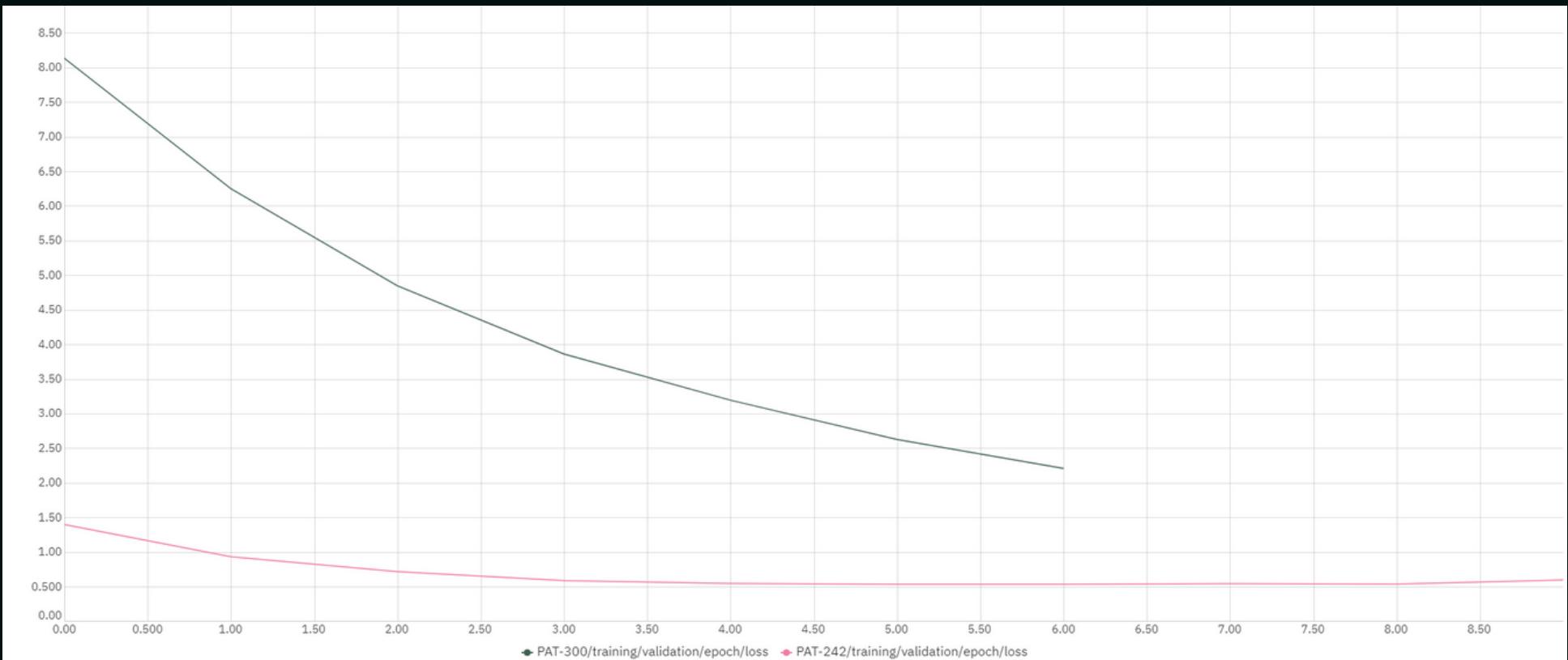
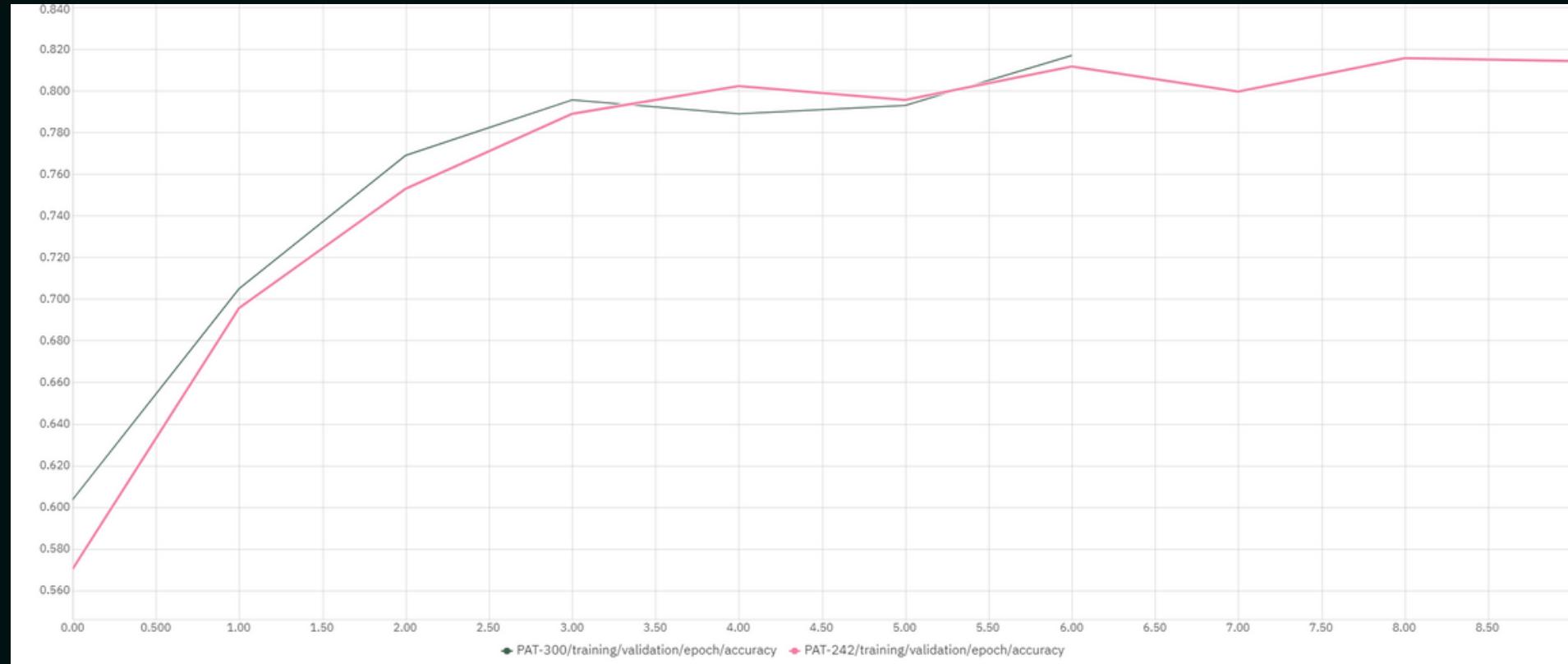
## Υπερπαραμέτροι που δοκιμάστηκαν

- Διαφορετικές τιμές για Dropout
- L1 και L2 kernel regularizers
- Διαφορετικές τιμές για epochs

Βελτιωμένο μοντέλο με την μεγαλύτερη απόδοση συγκριτικά με το μη βελτιωμένο μοντέλο

●	● PAT-300 ○	0.817333	2.217608	7
●	● PAT-242 ○	0.814667	0.60578	10

# Διαγράμματα σύγκρισης των 2 μοντέλων



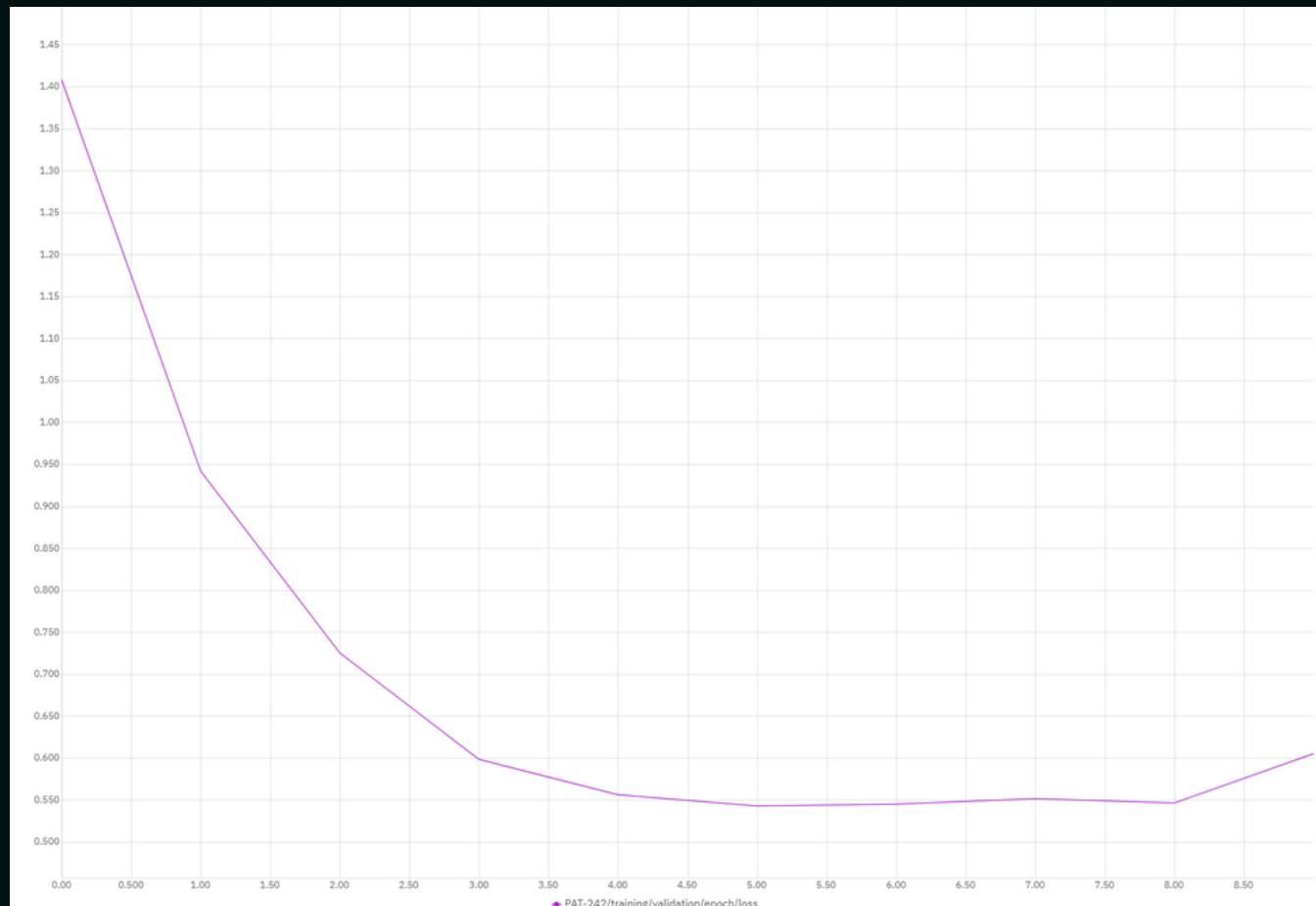
Παρατηρήσεις:

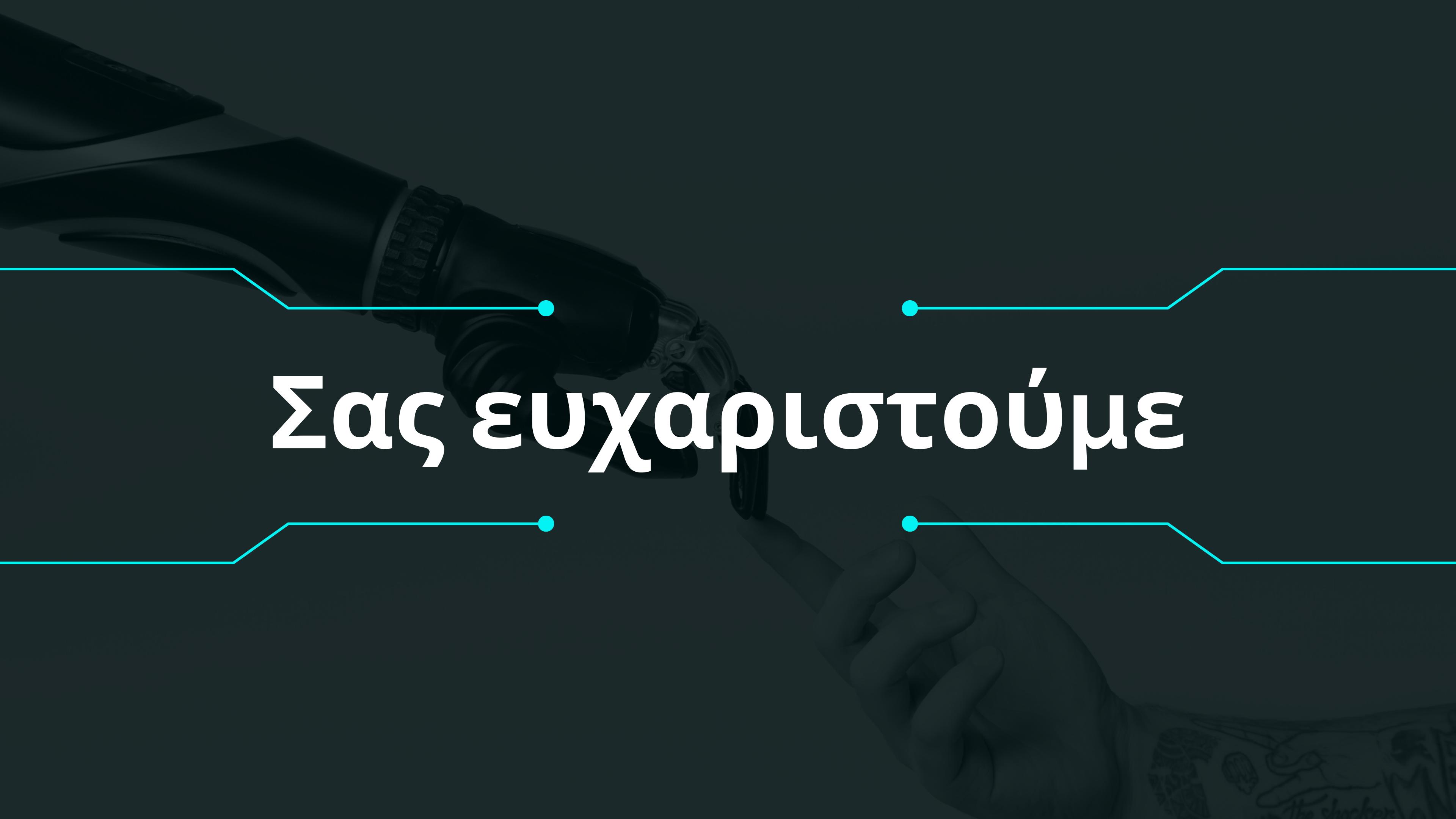
- Ελάχιστα αυξημένη ακρίβεια
- Μεγάλη αύξηση στις απώλειες

## Τελικό μοντέλο

- Νευρωνικό δίκτυο με 3 εσωτερικά επίπεδα
  - Εσωτερικό επίπεδο 1: 256 neurons, ReLU activation, και 50% dropout
  - Εσωτερικό επίπεδο 2: 256 neurons, ReLU activation, και 50% dropout
  - Εσωτερικό επίπεδο 3: 256 neurons, ReLU activation, και 50% dropout
- Εξωτερικό επίπεδο: Dense layer με 10 neurons και softmax activation
- Χωρίς χρήση kernel\_regularizer

neptune link :<https://app.neptune.ai/o/elemxm/org/Pattern-Recognition/runs/table?viewId=standard-view>





Σας ευχαριστούμε

