ΘΕΩΡΙΑ ΑΠΟΦΑΣΕΩΝ

ΕΡΓΑΣΙΑ 2
Παραχάραξη χαρτονομισμάτων

Ονοματεπώνυμο: Χρυσαυγή Πατέλη
Α.Μ.: 1084513
Εξάμηνο: 9ο
e-mail: up1084513@ac.upatras.gr

Διδάσκων : Δημήτριος Κοσμόπουλος

https://github.com/chryssa-pat/Decision-Theory

Τμήμα Μηχανικών Η/Υ και Πληροφορικής

Ακαδημαϊκό Έτος 2024-2025

ΠΕΡΙΕΧΟΜΕΝΑ

1. Βιβλιοθήκες

Οι βιβλιοθήκες που χρησιμοποιήθηκαν είναι:

- **pandas:** χρησιμοποιείται για την ανάγνωση του αρχείου csv και χρησιμοποιεί DataFrames για την ανάλυση και διαχείριση δεδομένων.
- **matplotlib:** χρησιμοποιείται για την δημιουργία γραφημάτων (συνάρτηση **pyplot**).
- **seaborn:** χρησιμοποιείται για την δημιουργία σύνθετων γραφημάτων (heatmaps).
- **sklearn:** από την συγκεκριμένη βιβλιοθήκη χρησιμοποιούνται οι υποβιβλιοθήκες:
    - **preprocessing:** προσφέρει εργαλεία για την κανονικοποίηση των δεδομένων χρησιμοποιείται η συνάρτηση **StandardScaler**, η οποία κανονικοποιεί τα δεδομένα ώστε να έχουν μέση τιμή 0 και τυπική απόκλιση 1.
    - **model_selection:** χρησιμοποιείται η συνάρτηση **train_test_split** για να χωρίσει τα δεδομένα σε train, test σύνολα. Επίσης, χρησιμοποιείται η συνάρτηση **GridSearchCV** ώστε να βρεθούν οι κατάλληλες υπερπαράμετροι μέσω cross-validation.
    - **decomposition:** χρησιμοποιείται η συνάρτηση **PCA** ώστε να μειωθούν οι διαστάσεις των δεδομένων.
    - **neighbors:** χρησιμοποιείται η συνάρτηση **KNeighborsClassifier**, για να υλοποιηθεί ο αλγόριθμος ταξινόμησης kNN.
    - **svm:** χρησιμοποιείται για την ταξινόμηση με SVM με γραμμικούς πυρήνες αλλά και μη γραμμικούς (RBF).
    - **metrics:** χρησιμοποιούνται οι συναρτήσεις **accuracy_score** για να υπολογιστούν οι ακρίβειες των μοντέλων**, classification_report** για την αναλυτική αναφορά με precision, recall, και F1-score, **confusion_matrix** και **roc_curve, auc** για την δημιουργία της καμπύλης ROC και τον υπολογισμό AUC.
    - **mixture:** χρησιμοποιείται η συνάρτηση **GaussianMixture**, για την μοντελοποίηση των θετικών δειγμάτων.
- **numpy:** χρησιμοποιείται για επιστημονικούς υπολογισμούς.

2. Προεπεξεργασία

Στόχος της συγκεκριμένης άσκησης είναι η ανάλυση και ταξινόμηση χαρτονομισμάτων, γνήσιων και πλαστών, χρησιμοποιώντας μεθόδους μηχανικής μάθησης. Ειδικότερα, η άσκηση στοχεύει στην εφαρμογή ταξινομητών όπως Parzen Window με PCA, k-NN και SVM. Τα δεδομένα περιέχουν τις στήλες variance, skewness, curtosis, entropy οι οποίες προέκυψαν μέσω της μεθόδου του μετασχηματισμού κυματιδίων (Wavelet Transform). Η τελευταία στήλη class υποδεικνύει αν τα χαρτονομίσματα είναι γνήσια (κλάση 0) ή πλαστά (κλάση 1).

Αρχικά, πραγματοποιείται διαχωρισμός των δεδομένων και οι ανεξάρτητες μεταβλητές (όλες οι στήλες εκτός από την τελευταία) αποθηκεύονται στην μεταβλητή x ενώ η εξαρτημένη μεταβλητή (τελευταία στήλη) αποθηκεύεται στην μεταβλητή y. Στην συνέχεια, το σύνολο δεδομένων διασπάται σε 60% και 40% χρησιμοποιώντας την συνάρτηση **train_test_split()**. Το 60% θα αποτελέσει το σύνολο εκπαίδευσης. Το 40% χωρίζεται περεταίρω σε 50% και 50% χρησιμοποιώντας την ίδια συνάρτηση και θα αποτελέσουν το σύνολο επικύρωσης και το σύνολο δοκιμής. Η παράμετρος **satisfy** στην συνάρτηση **train_test_split()** διασφαλίζει την αναλογία των κλάσεων ανάμεσα στα σύνολα, πιο συγκεκριμένα, τα σύνολα δεδομένων που προκύπτουν θα έχουν την ίδια κατανομή κλάσεων όπως στο αρχικό σύνολο.

Επίσης, γίνεται κανονικοποίηση των δεδομένων χρησιμοποιώντας τον **StandardScaler()**, ώστε κάθε χαρακτηριστικό να έχει μέση τιμή 0 και τυπική απόκλιση 1. Η συνάρτηση **fit_transform()** εφαρμόζεται στα δεδομένα εκπαίδευσης για να προσαρμόσει τον κανονικοποιητή και να μετασχηματίσει τα δεδομένα. Τέλος, η συνάρτηση **transform()** εφαρμόζεται στο σύνολο επικύρωσης και δοκιμής χρησιμοποιώντας τις ίδιες παραμέτρους κανονικοποίησης, ώστε να διατηρηθεί η συνέπεια.

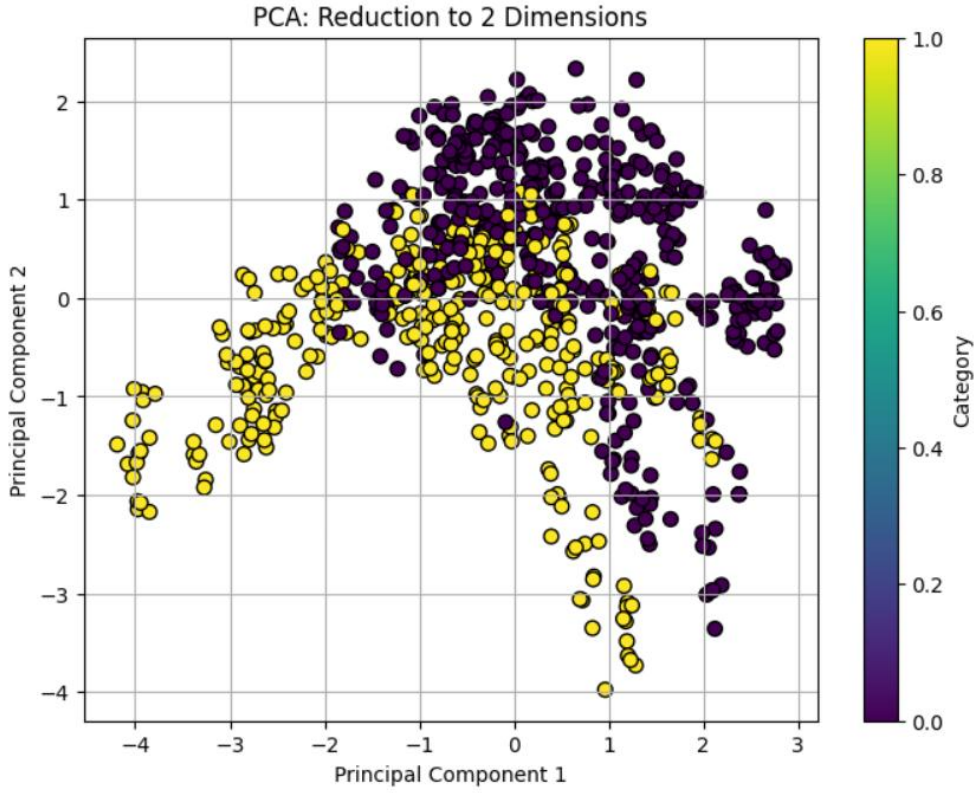
3. Parzen Window με PCA

3.1 PCA

Προκειμένου να πραγματοποιηθεί μείωση των διαστάσεων του dataset από 4 σε 2 εφαρμόζεται η τεχνική PCA καλώντας την συνάρτηση **PCA(n_components=2)** με όρισμα 2 που συμβολίζει την μείωση του dataset σε 2 κύριες συνιστώσες. Στην συνέχεια, η PCA εφαρμόζεται στο σύνολο εκπαίδευσης με την μέθοδο **fit_transform()** και μετασχηματίζει τα δεδομένα στην καινούρια διάσταση. Επιπλέον, μετασχηματίζονται και τα σύνολα επικύρωσης και δοκιμής χρησιμοποιώντας τον ίδιο μετασχηματισμό που εφαρμόστηκε και στο σύνολο εκπαίδευσης. Τέλος, εκτυπώνεται το ποσοστό διακύμανσης για κάθε συνιστώσα **pca.explained_variance_ratio_** και οπτικοποιείται το σύνολο εκπαίδευσης δημιουργώντας διάγραμμα διασποράς.

```
Explained Variance Ratio: [0.55211493 0.31525425]
```

Η μέθοδος PCA μετασχηματίζει τα αρχικά χαρακτηριστικά σε νέες συνιστώσες οι συνιστώσες που επιλέγονται αντιστοιχούν σε γραμμικούς συνδυασμούς των αρχικών χαρακτηριστικών που διατηρούν περισσότερη πληροφορία. Η 1η κύρια συνιστώσα εξηγεί το 55,21% της συνολικής διακύμανσης και η 2η κύρια συνιστώσα εξηγεί το 31,52% της συνολικής διακύμανσης. Συνολικά και οι δυο μαζί εξηγούν το 86,73%, αυτό το ποσοστό είναι αρκετά υψηλό γεγονός που συμβολίζει ότι το μεγαλύτερο ποσοστό πληροφορίας διατηρείται κατά την μείωση των διαστάσεων.



Στο συγκεκριμένο διάγραμμα διασποράς κάθε σημείο αντιστοιχεί σε ένα σύνολο δεδομένων. Τα σημεία χρωματίζονται ανάλογα με την κατηγορία στην οποία ανήκουν (0 ή 1) για να είναι ευδιάκριτες οι ομάδες που σχηματίζονται. Ο διαχωρισμός των ομάδων είναι γενικά ευδιάκριτος παρόλο που υπάρχει κάποιο ποσοστό επικάλυψης, αυτό σημαίνει ότι τα δεδομένα έχουν δομή που μπορεί να χρησιμοποιηθεί για ταξινόμηση.

3.2 Training-Validation

Αρχικά, δημιουργείται η συνάρτηση **gaussian_kernel(x, h)** για να υπολογίζει και να επιστρέφει μια πιθανότητα που βασίζεται στην απόσταση μεταξύ των query point και των δεδομένων. Το h που δίνεται και σαν όρισμα είναι το bandwidth που καθορίζει την επιρροή κάθε παρατήρησης. Ο τύπος που υπολογίζει τον gaussian kernel είναι: $K(x) = \frac{1}{2\sqrt{2\pi}h} e^{-\frac{1}{2}(\frac{x}{h})^2}$

Η δεύτερη συνάρτηση που δημιουργείται είναι η **parzen_window_classifier(data, labels, query_point, h)**, η οποία δημιουργεί τον ταξινομητή που βασίζεται σε Parzen παράθυρα. Δέχεται σαν όρισμα τις παραμέτρους:

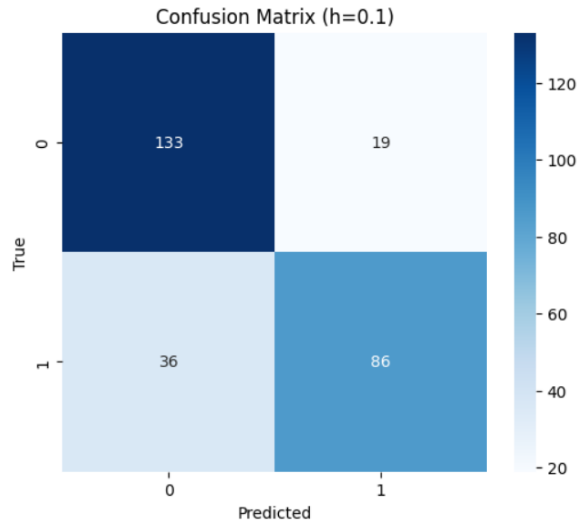
- data: σύνολο εκπαίδευσης
- labels: κατηγορίες των δεδομένων (0 και 1)
- query_point: σημείο προς ταξινόμηση
- h: bandwidth (παράθυρο)

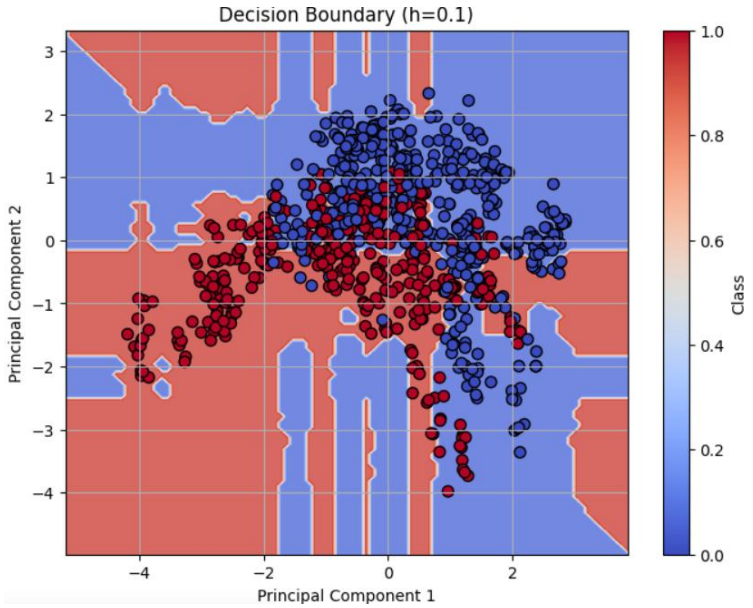
Για κάθε κατηγορία υπολογίζεται η πυκνότητα γύρω από το query point (φιλτράρονται τα δεδομένα που ανήκουν στην κατηγορία και υπολογίζεται το άθροισμα των gaussian kernel τιμών) και κανονικοποιούνται οι πυκνότητες. Η κατηγορία με την μεγαλύτερη πιθανότητα είναι αυτή που επιστρέφεται.

Εξετάζονται διαφορετικές τιμές για το bandwidth (μέγεθος παραθύρου) ώστε να βρεθεί ποια είναι η καταλληλότερη για το συγκεκριμένο σύνολο δεδομένων **h_values = [0.1, 0.5, 1.0, 2.0].** Για κάθε τιμή h εφαρμόζεται η συνάρτηση **parzen_window_classifier()** και υπολογίζεται η ακρίβεια **accuracy_score()** τόσο για το σύνολο εκπαίδευσης όσο και για το σύνολο επικύρωσης**,** οι τιμές ακρίβειας αποθηκεύονται στις λίστες **accuracy_train_values** και **accuracy_val_values** αντίστοιχα**.** Επιπλέον, για το σύνολο επικύρωσης για κάθε h δημιουργείται confusion matrix, ώστε να δείξει πόσο καλά ταξινομούνται τα δεδομένα και απεικονίζεται και το decision boundary για το training set, ώστε να φανούν τα όρια απόφασης του ταξινομητή. Τέλος, απεικονίζονται σε ένα γράφημα οι τιμές των accuracies για κάθε τιμή του h τόσο για το validation όσο και για το training set.

- **h = 0.1**:

```
Parzen Window Classification with h = 0.1
Training Accuracy: 0.8128797083839611
Validation Accuracy: 0.7992700729927007
```



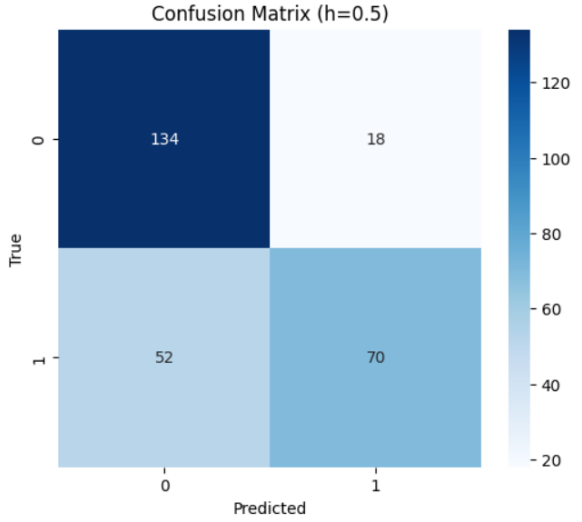


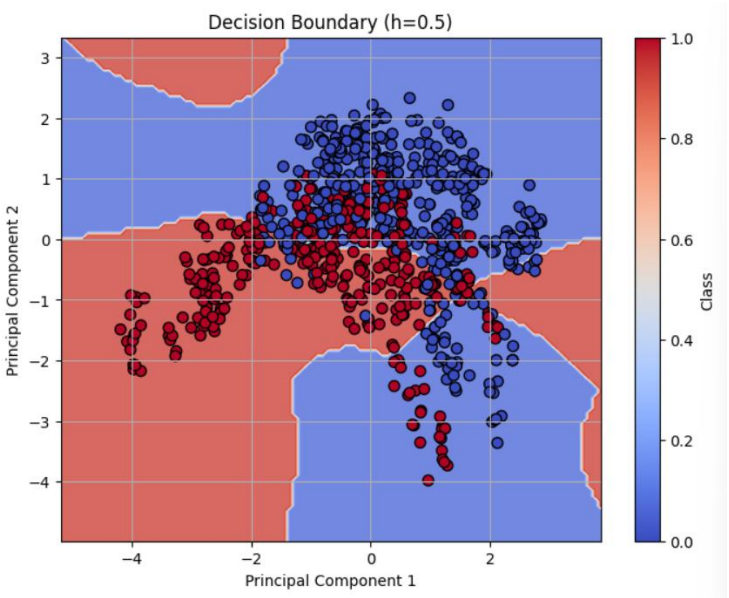
Από το decion boundary προκύπτει ότι τα σύνορα απόφασης είναι λεπτομερη και ακανόνιστα, γεγονός που υποδηλώνει ότι υπάρχει overfitting, καθώς το μοντέλο προσαρμόζεται υπερβολικά στις τοπικές διαφοροποιήσεις των δεδομένων εκπαίδευσης.

Από το confusion matrix προκύπτει ότι υπάρχουν 133 σωστές προβλέψεις για την κλάση 0 (true positives) και 86 σωστές προβλέψεις για την κλάση 1 (true negatives). Επίσης, υπάρχουν περισσότερα false negatives (36) από false positives (19) άρα η κλάση 1 ταξινομείται συχνά ως κλάση 0.

- **h = 0.5**:

```
Parzen Window Classification with h = 0.5
Training Accuracy: 0.764277035236938
Validation Accuracy: 0.7445255474452555
```



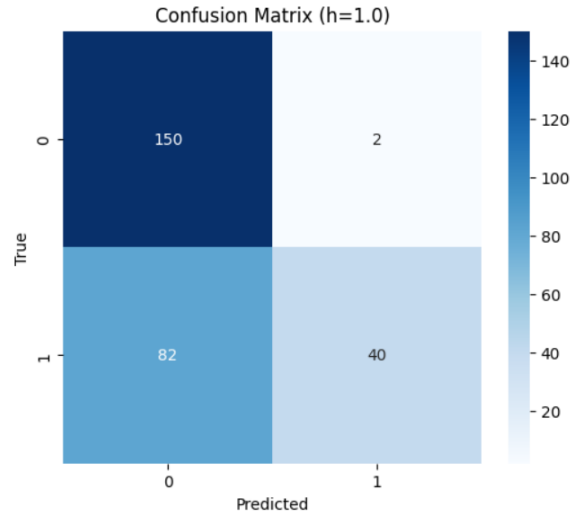


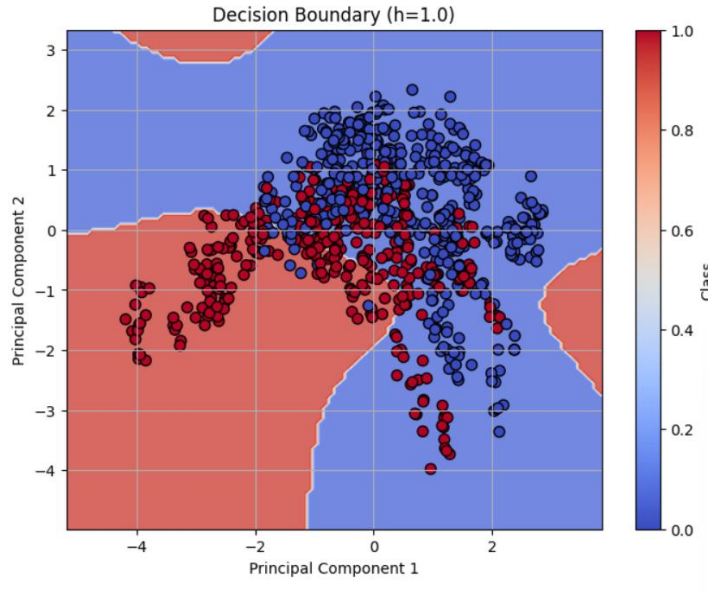
Από το decion boundary προκύπτει ότι τα σύνορα απόφασης είναι πιο ομαλά γεγονός που υποδηλώνει ότι υπάρχει καλή ισορροπία μεταξύ ευελιξίας και γενίκευσης, καθιστώντας το μοντέλο πιο ανθεκτικό στο σύνολο επικύρωσης.

Από το confusion matrix προκύπτει ότι υπάρχουν 134 σωστές προβλέψεις για την κλάση 0 και 70 σωστές προβλέψεις για την κλάση 1, με 52 false negatives και 18 false positives.

- **h = 1.0**:

```
Parzen Window Classification with h = 1.0
Training Accuracy: 0.7083839611178615
Validation Accuracy: 0.6934306569343066
```

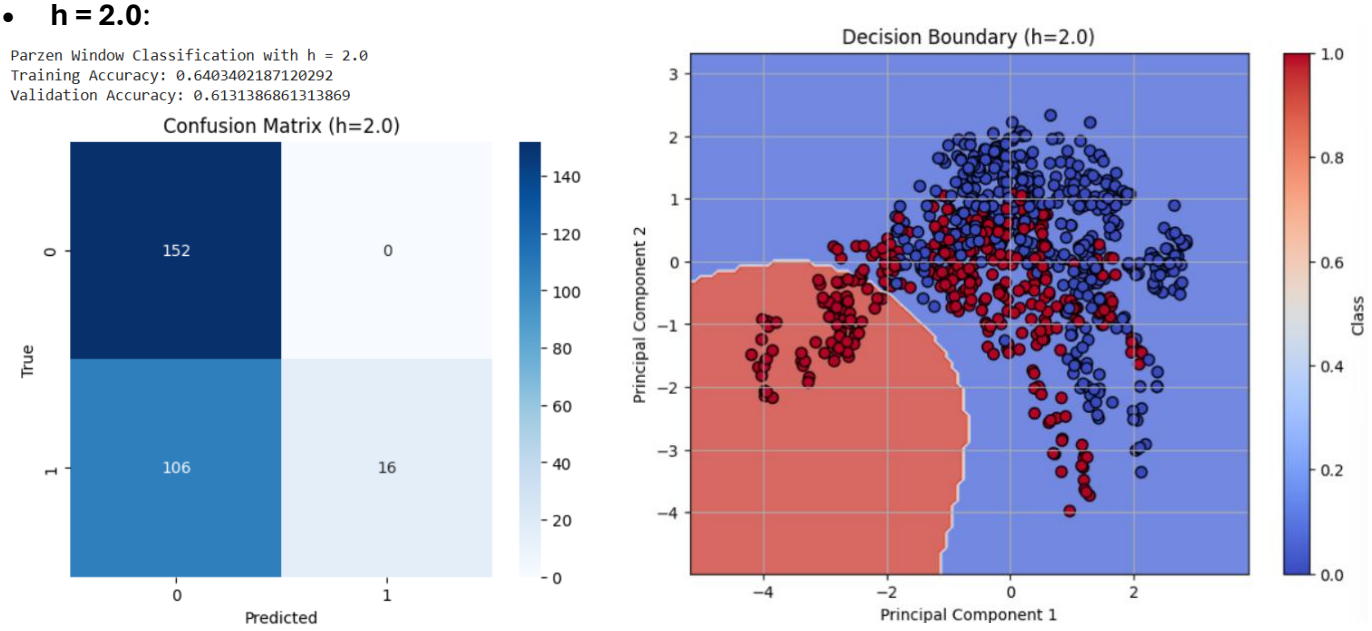




Από το decion boundary προκύπτει ότι τα σύνορα απόφασης είναι ευρύτερα και λιγότερα λεπτομερή, γεγονός που υποδηλώνει το μοντέλο είναι απλοποιημένο και θα οδηγήσει σε περισσότερες λανθασμένες ταξινομήσεις.

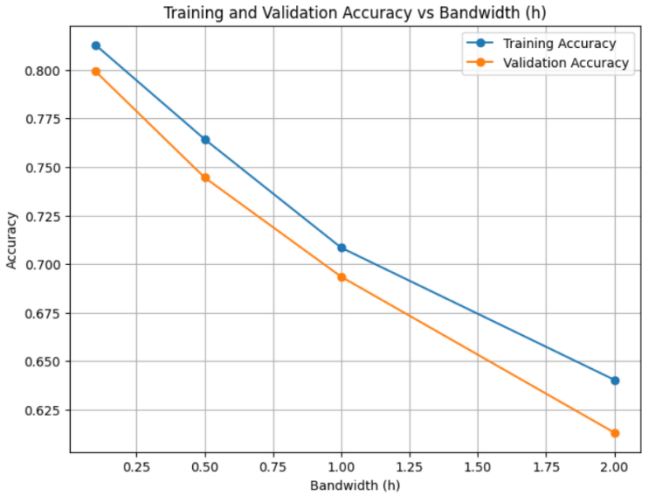
Από το confusion matrix προκύπτει ότι υπάρχουν, 150 σωστές προβλέψεις για την κλάση 0 και 40 σωστές προβλέψεις για την κλάση 1, με 82 false negatives και 2 false positives, άρα το μοντέλο γενικεύει λιγότερο καλά σε σχέση με τα μικρότερα h, με αύξηση των false negatives (82).

- **h = 2.0**:



Από το decion boundary προκύπτει ότι τα σύνορα απόφασης είναι υπερβολικά απλά και διαιρούν τον χώρο σε μεγάλες περιοχές. Αυτό έχει ως αποτέλεσμα την κακή απόδοση του μοντέλου, καθώς αποτυγχανει να αναγνωρίσει τις λεπτομέρειες των δεδομένων.

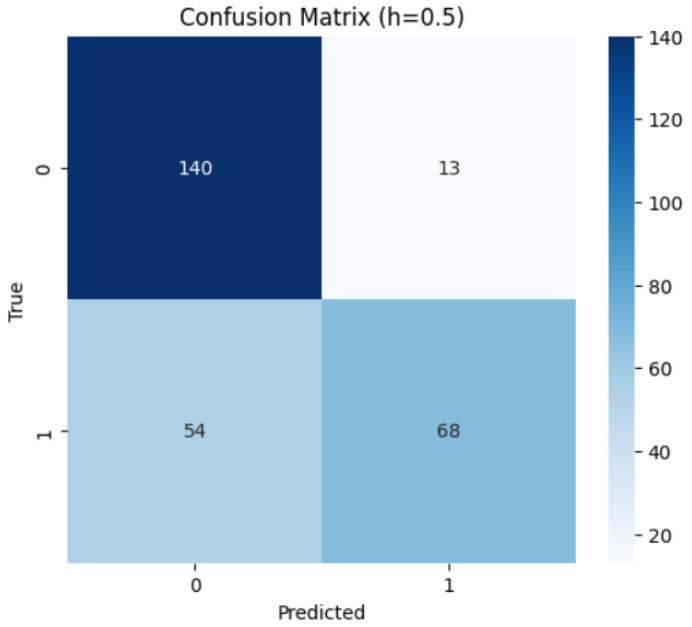
Από το confusion matrix προκύπτει ότι υπάρχουν, 152 σωστές προβλέψεις για την κλάση 0 και 176 σωστές προβλέψεις για την κλάση 1, με 106 false negatives και 0 false positives. Παρατηρείται σημαντική μείωση στις σωστές ταξινομήσεις για την κλάση 1 και υψηλά false negatives γεγονός που δηλώνει κακή αναγνώριση της κλάσης 1.



Από την γραφική παρατηρείται ότι όσο αυξάνεται το h η ακρίβεια μειώνεται, καθώς το μοντέλο γίνεται πιο γενικό. Αυτό προκύπτει και από τα decision boundaries καθώς υψηλότερες τιμές του h οδηγούν σε απλούστερα boundaries. Για πολύ μικρές τιμές του h μπορεί να υπάρξει overfitting, καθώς το μοντέλο θα προσαρμόζεται υπερβολικά στις τοπικές διαφοροποιήσεις των δεδομένων εκπαίδευσης. Ενώ για πολύ υψηλές τιμές του h μπορεί να υπάρξει underfitting, καθώς το μοντέλο θα χάνει σημαντικές λεπρομέρειες. Έτσι, θα επιλεχθεί το **h=0.5** ώστε να υπάρχει ισορροπία μεταξύ overfitting και underfitting και μεταξύ ακρίβειας και γενίκευσης.

3.3 Testing

Για κάθε σημείο του συνόλου δοκιμής υπολογίζεται η κατηγορία του χρησιμοποιώντας την συνάρτηση **parzen_window_classifier()** με ορίσματα τα δεδομένα εκπαίδευσης, το τρέχων σημείο και το επιλεγμένο bandwidth h=0.5. Στην συνέχεια, δημιουργείται το confusion matrix και υπολογίζεται το classification report. Επίσης, για κάθε σημείο του συνόλου δοκιμής υπολογίζεται η πιθανότητα κάθε κατηγορίας βασισμένη στις κανονικοποιημένες πυκνότητες. Τα αποτελέσματα με τις πιθανότητες θα χρησιμοποιηθούν για την δημιουργία της ROC curve. Χρησιμοποιώντας την πιθανότητα της κατηγορίας 1 υπολογίζονται οι δείκτες **fpr** (ποσοστό false positive), **tpr** (ποσοστό true positive). Τέλος, υπολογίζεται και η AUC (εμβαδό κάτω από την καμπύλη) που δείχνει πόσο καλά ο ταξινομητής διαχωρίζει τις κλάσεις και δημιουργείται η ROC curve.



Parzen Window Classification on Test Set with h = 0.5
Accuracy: 0.7563636363636363

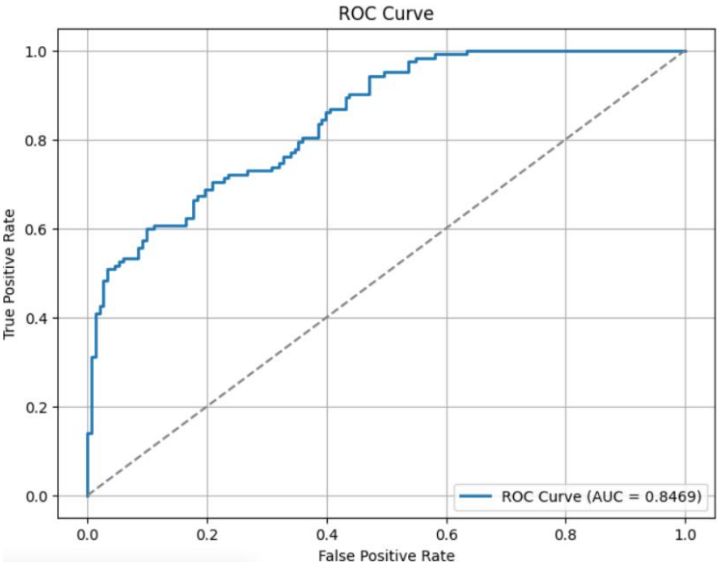
	precision	recall	f1-score	support
0	0.72	0.92	0.81	153
1	0.84	0.56	0.67	122
accuracy			0.76	275
macro avg	0.78	0.74	0.74	275
weighted avg	0.77	0.76	0.75	275

Από το confusion matrix προκύπτει ότι υπάρχουν 140 σωστές προβλέψεις για την κλάση 0 (true positives) και 68 σωστές προβλέψεις για την κλάση 1 (true negatives). Επίσης, υπάρχουν 54 false negatives και 13 false positives.

Σχολιασμός μετρικών λαμβάνοντας υπόψη το weighted average που λαμβάνει υπόψην τις αναλογίες των δειγμάτων κάθε κλάσης στο σύνολο:

- **Accuracy:** 0.756, Ο ταξινομητής αποδίδει καλά, ταξινομώντας σωστά περίπου το 75.6% των δειγμάτων.
- **Precision:** 0.77
- **Recall:** 0.76
- **F1-Score:** 0.75, δείχνει ότι ο ταξινομητής έχει μια γενική ισορροπία μεταξύ precision και recall, αλλά η επίδοση δεν είναι εξαιρετική.

Ο ταξινομητής αποδίδει καλύτερα στην ακρίβεια (precision) σε σχέση με την ανάκληση (recall). Αυτό σημαίνει ότι κάνει λιγότερα λάθη όταν προβλέπει τις κλάσεις, αλλά αποτυγχάνει να εντοπίσει αρκετά σωστά παραδείγματα από την κατηγορία 1.



Η AUC είναι 0.8469, επειδή είναι αρκετά μεγαλύτερη από τον τυχαίο ταξινομητή (0.5) και πλησιάζει το 1, ο ταξινομητής έχει καλή διαχωριστική ικανότητα. Ωστόσο, υπάρχει περιθώριο βελτίωσης για την κατηγορία 1.

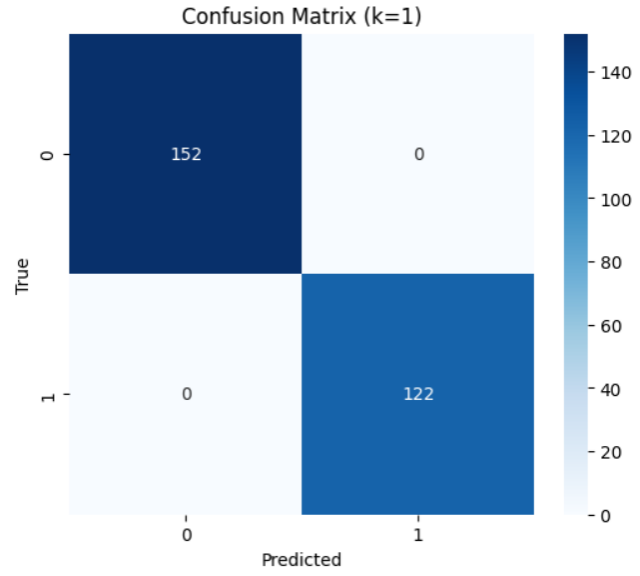
4. k-NN

4.1 Training- validation

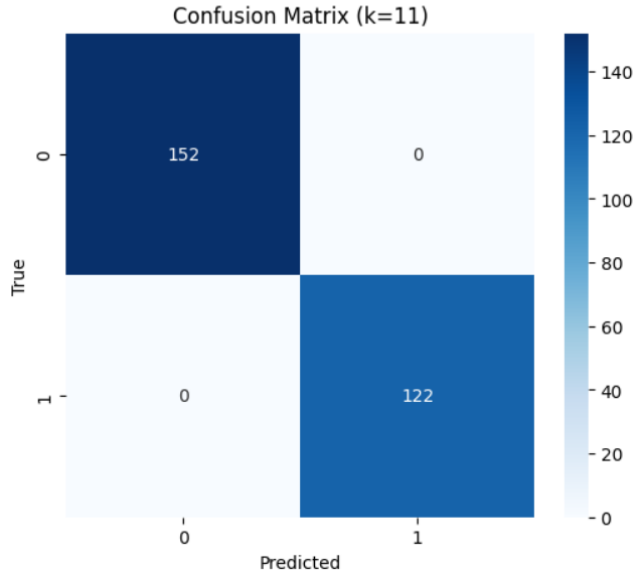
Για διάφορετικές τιμές του k **k_values = [1, 3, 5, 7, 9, 11, 13, 15]** εφαρμόζεται επανληπτικά η συνάρτηση **KNeighborsClassifier(n_neighbors=k),** η οποία δημιουργεί ένα kNN ταξινομητή. Στην συνέχεια, στην συνάρτηση **fi()** δίνονται ώς ορίσματα το σύνολο εκπαίδευσης **X_train_scaled,** και οι κατηγορίες **y_train** ώστε να εκαπιδευτεί ο ταξινομητής. Επιπλέον, με την συνάρτηση **predict()** προβλέπονται οι κατηγορίες για το σύνολο εκπαίδευσης και με την συνάρτηση **accurancy_score()**, υπολογίζεται η ακρίβεια του μοντέλου συγκρίνοντας τις πραγματικές κατηγορίες με τις προβλεπόμενες και αποθηκεύεται στη λίστα **accuracy_scores_train.** Με την συνάρτηση **predict()** προβλέπονται και οι κατηγορίες για το σύνολο επικύρωσης και όπώς και πριν με την συνάρτηση **accurancy_score()**, υπολογίζεται η ακρίβεια του μοντέλου συγκρίνοντας τις πραγματικές κατηγορίες με τις προβλεπόμενες και αποθηκεύεται στη λίστα **accuracy_scores_validation**. Για κάθε k, εκτυπώνεται η ακρίβεια του συνόλου επικύρωσης και του συνόλου εκπαίδευσης και δηιουργείται το confusion matrix για το σύνολο επικύρωσης. Τέλος, απεικονίζονται σε ένα γράφημα οι τιμές των accuracies για κάθε τιμή του k τόσο για το validation όσο και για το training set.

Ενδεικτικά παρατίθενται τυχαία 4 από τα 8 confusion matrix που δημιουργούνται:

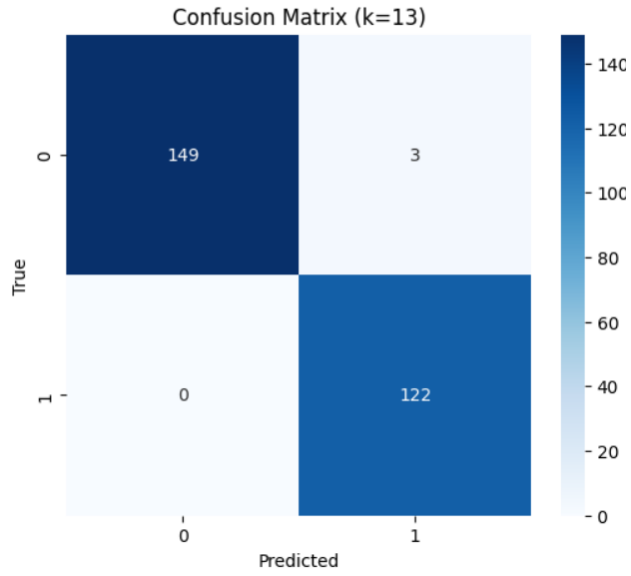
```
k-NN Classification with k = 1
Accuracy on training set: 1.0
Accuracy on validation set: 1.0
```



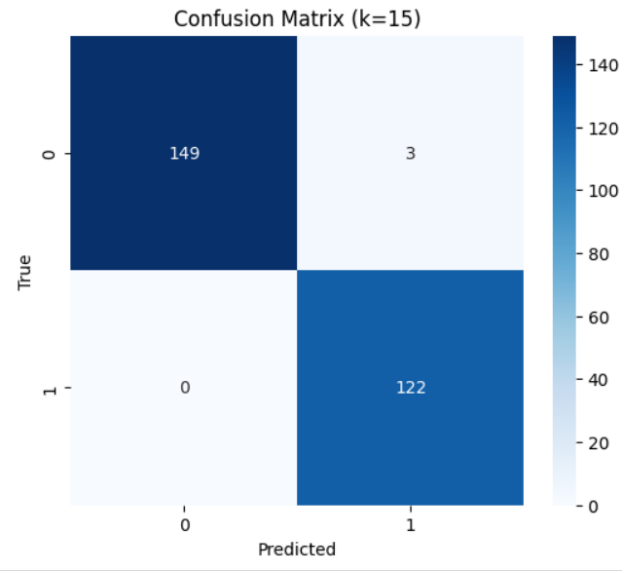

```
k-NN Classification with k = 11
Accuracy on training set: 0.9927095990279465
Accuracy on validation set: 1.0
```

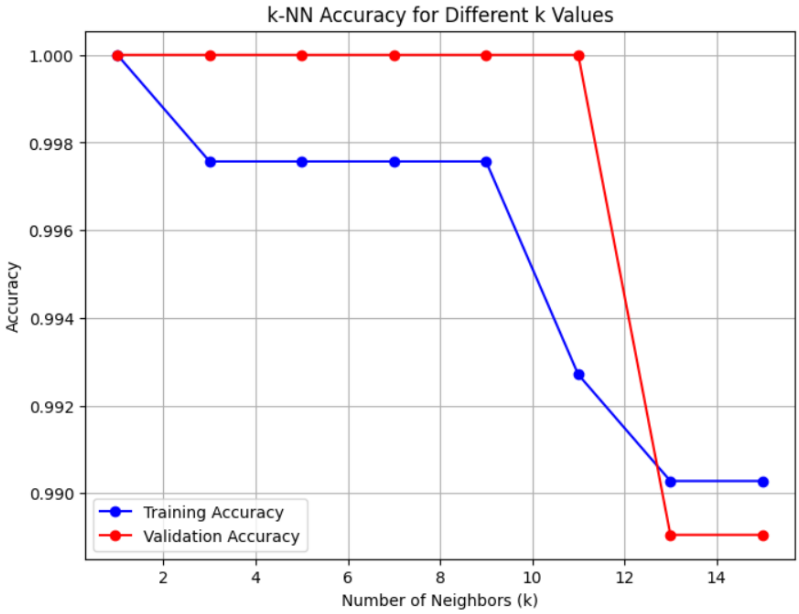


```
k-NN Classification with k = 13
Accuracy on training set: 0.9902794653705954
Accuracy on validation set: 0.9890510948905109
```




```
k-NN Classification with k = 15
Accuracy on training set: 0.9902794653705954
Accuracy on validation set: 0.9890510948905109
```

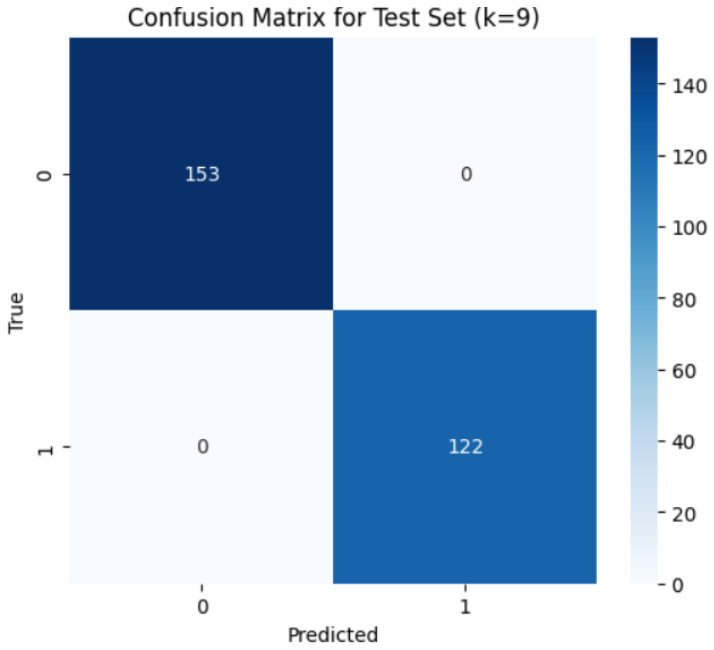




Παρατηρώντας την γραφική προκύπτει ότι η ακρίβεια για το σύνολο εκπαίδευσης μειώνεται όσο μεγαλώνει η τιμή του k. Μικρό k μπορεί να οδηγήσει ακόμα και σε overfitting, καθώς το μοντέλο είναι ευαίσθητο στις αλλαγές στα δεδομένα. Ενώ, μεγαλύτερο k εξασφαλίζει ότι το μοντέλο θα είναι περισσότερο ανθεκτικό αφού θα εξετάζονται περισσότεροι γείτονες πριν πραγματοποιήσει την πρόβλεψη. Ωστόσο, αν το k είναι πολύ μεγάλο, αγνοεί τις λεπτομέρειες και μπορεί να κάνει λάθος προβλέψεις, γιατί λαμβάνει υπόψη και δεδομένα που είναι πολύ μακριά από το σημείο που θέλει να προβλέψει. Όσον αφορά την ακρίβεια για το σύνολο επικύρωσης, παρατηρείται ότι μέχρι k=11 παραμένει μέγιστη και μετά μειώνεται απότομα. Επομένως, η τιμή του k που θα επιλεχθεί είναι η **k=9**, καθώς συνδυάζει υψηλή ακρίβεια στην επικύρωση και καλή γενίκευση.

4.2 Testing

Για k=9 δημιουργείται ο ταξινομητής και πραγματοποιούνται οι προβλέψεις στο test set χρησιμοποιώντας την μέθοδο **predict()**. Επιπλέον, δημιουργείται το confusion matrix και υπολογίζεται το classification report. Επίσης, με την συνάρτηση **predict_proba()** υπολογίζονται οι πιθανότητες για κάθε κατηγορία (0 και 1) και χρησιμοποιώντας την πιθανότητα της κατηγορίας 1 υπολογίζονται οι δείκτες **fpr** (ποσοστό false positive), **tpr** (ποσοστό true positive). Τέλος, υπολογίζεται και η AUC (εμβαδό κάτω από την καμπύλη) και δημιουργείται η ROC curve.



k-NN Classification on Test Set with k = 9
Accuracy: 1.0

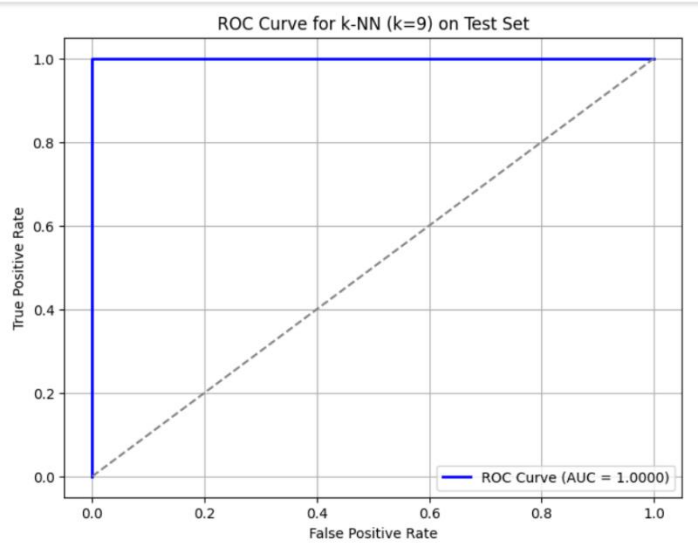
	precision	recall	f1-score	support
0	1.00	1.00	1.00	153
1	1.00	1.00	1.00	122
accuracy			1.00	275
macro avg	1.00	1.00	1.00	275
weighted avg	1.00	1.00	1.00	275

Από το confusion matrix προκύπτει ότι δεν υπάρχουν υπάρχουν λάθη στην ταξινόμηση, καθώς υπάρχουν 153 σωστές προβλέψεις για την κλάση 0 και 122 σωστές προβλέψεις για την κλάση 1, με 0 false negatives και 0 false positives.

Σχολιασμός μετρικών:

- **Accuracy:** 1.0, όλα τα δείγματα ταξινομήθηκαν σωστά.
- **Precision:** 1, δεν υπάρχουν ψευδώς θετικές προβλέψεις.
- **Recall:** 1, το μοντέλο πέτυχε όλες τις προβλέψεις.
- **F1-Score:** 1,δείχνει τέλεια ισορροπία μεταξύ precision και recall.

Συμπερασματικά,το μοντέλο αποδίδει τέλεια στο test set. Η άριστη απόδοση είναι συνεπής με την με την καλή απόδοση στο training και validation set, γεγονός που υποδηλώνει ότι το μοντέλο γενικεύει σωστά και δεν υπέστη overfitting.



Η AUC είναι **1**, που υποδεικνύει τέλεια απόδοση στη διάκριση μεταξύ των θετικών και αρνητικών κλάσεων στο σύνολο δοκιμής, χωρίς κανένα λάθος.

5. SVM

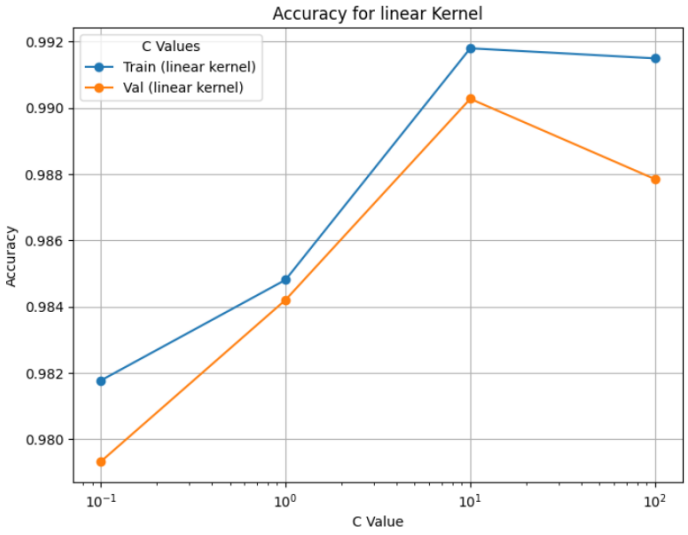
5.1 Training- validation

Αρχικά, δημιουργείται η συνάρτηση **svm_grid_search()** για να εφαρμόσει SVM με διαφορετικές υπερπαραμέτρους και να επιλέξει τις καλύτερες με βάση την ακρίβεια του μοντέλου. Δέχεται σαν ορίσματα τα δεδομένα εκπαίδευσης (**X_train**), τις κατηγορίες των δεδομένων εκπαίδευσης (**y_train**), τα δεδομένα επικύρωσης (**X_val**), τις κατηγορίες των δεδομένων επικύρωσης (**y_val**), τον τύπο του πυρήνα (**kernel**) που θα είναι γραμμικός ή μη γραμμικός (RBF) και τέλος το σύνολο υπερπαραμέτρων (**param_grid**). Μέσα στην συνάρτηση χρησιμοποιείται η **GridSearchCV()** προκειμένου μέσω cross-validation σε 5 υποσύνολα να υλοποιήσει διαφορετικούς συνδυασμούς υπερπαραμέτρων. Το μοντέλο εκπαιδεύεται μέσω της συνάρτησης **fit()** στους διάφορους συνδυασμούς παραμέτρων και οι παράμετροι που προσφέρουν την μεγαλύτερη ακρίβεια στο μοντέλο αποθηκεύονται. Επίσης, με την συνάρτηση **predict()** προβλέπονται οι κατηγορίες για το σύνολο εκαπίδευσης αλλά και για το σύνολο επικύρωσης και με την συνάρτηση **accurancy_score()**, υπολογίζεται η ακρίβεια του μοντέλου συγκρίνοντας τις πραγματικές κατηγορίες με τις προβλεπόμενες. Επιπλέον, εκτυπώνεται η ακρίβεια του συνόλου επικύρωσης και του συνόλου εκπαίδευσης και δηιουργείται το confusion matrix για το σύνολο επικύρωσης.

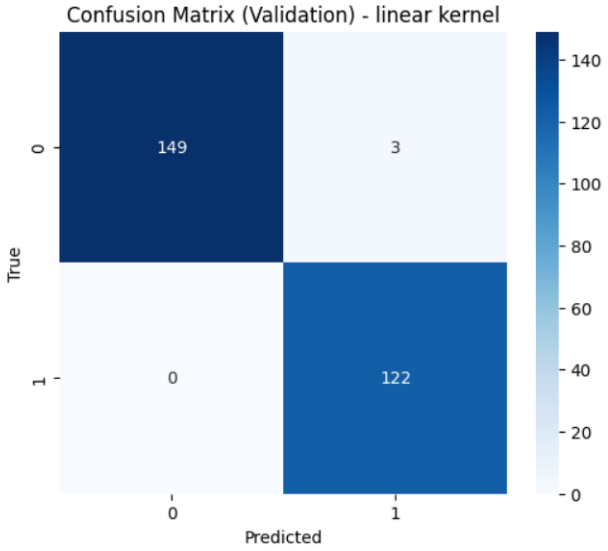
Στη συνέχεια, δημιουργείται το πλέγμα υπερπαραμέτρων **linear_param_grid = {'C': [0.1, 1, 10, 100]}** για το γραμμικό SVM και καλείται η συνάρτηση **svm_grid_search(),** για γραμμικό πυρήνα (linear kernel) με τις παραμέτρους του πλέγματος. Ακόμη, δημιουργείται το πλέγμα υπερπαραμέτρων **rbf_param_grid = {'C': [0.1, 1, 10, 100], 'gamma': [0.01, 0.1, 1, 10]}}** για το μη γραμμικό SVM και καλείται η συνάρτηση **svm_grid_search(),** για μη γραμμικό πυρήνα (RBF kernel) με τις παραμέτρους του πλέγματος.

Δημιουργείται η συνάρτηση **plot_accuracies()**, για να δημιουργήσει τα γραφήματα ακρίβειας για το σύνολο εκπαίδευσης και επικύρωσης, ανάλογα με τον πυρήνα που χρησιμοποιήθηκε. Δέχεται σαν όρισμα το αντικείμενο που δημιουργήθηκε από την συνάρτηση **svm_grid_search()** και το είδος του πυρήνα. Αν ο πυρήνας είναι RBF δημιουργεί γραφικές για κάθε τιμή της παραμέτρου gamma ξεχωριστά τόσο για το σύνολο εκπαίδευσης όσο και για το σύνολο επικύρωσης, ο άξονας x αντιπροσωπεύει τις διάφορες τιμές της παραμέτρου c και ο y τις ακρίβειες. Αν ο πυρήνας είναι γραμμικός δημιουργούνται 2 γραφικές μια για το σύνολο εκπαίδευση και μια για το σύνολο επικύρωσης για τις διάφορες τιμές του c. Τέλος, καλείται η συνάρτηση για να δημιουργηθούν τα γραφήματα ακρίβειας για το γραμμικό SVM και το μη γραμμικό SVM.

Αποτελέσματα Linear SVM :



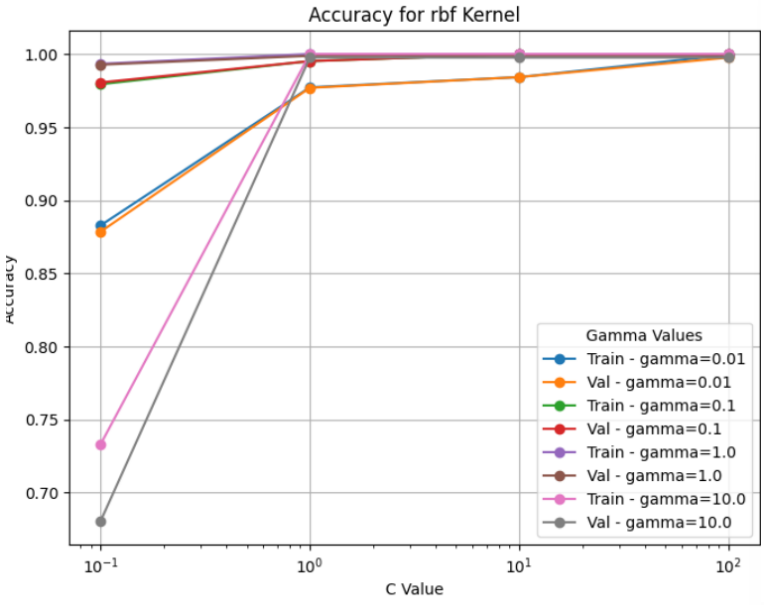

```
Best Parameters for linear kernel: {'C': 10}

Train Accuracy (linear kernel): 0.9927095990279465
Validation Accuracy (linear kernel): 0.9890510948905109
```



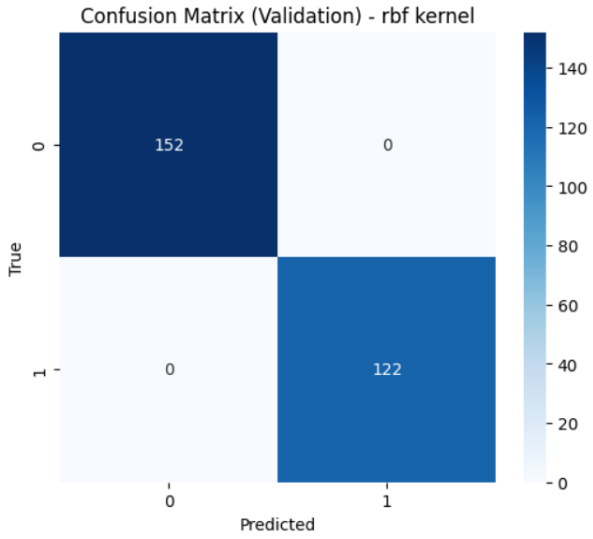
Από την γραφική παρατηρείται ότι όσο η τιμή της παραμέτρου c αυξάνεται τόσο αυξάνεται και η ακρίβεια τόσο στο σύνολο επικύρωσης όσο και στο σύνολο εκπαίδευσης και φτάνει την βέλτιστη τιμή για c=10. Για μεγαλύτερες τιμές της c η ακρίβεια του συνόλου επικύρωση μειώνεται ενώ στο σύνολο εκπαίδευσης παραμένει υψηλή, γεγονός που υποδεικνύει ότι υπάρχει overfitting δηλαδή μοντέλο προσαρμόζεται υπερβολικά στα δεδομένα εκπαίδευσης, χάνοντας τη γενίκευση στα δεδομένα επικύρωσης. Άρα μεγαλύτερη ακρίβεια πετυχαίνει για c=10 και αυτή η τιμή επιλέγεται και από το cross-validation.

Από το confusion matrix προκύπτει ότι υπάρχουν, 149 σωστές προβλέψεις για την κλάση 0 και 122 σωστές προβλέψεις για την κλάση 1, με 0 false negatives και 3 false positives. Άρα το μοντέλο γενικεύει αρκετά σωστά.

Αποτελέσματα Non-Linear SVM (RBF):




```
Best Parameters for rbf kernel: {'C': 10, 'gamma': 0.1}

Train Accuracy (rbf kernel): 1.0
Validation Accuracy (rbf kernel): 1.0
```



Από την γραφική παρατηρείται ότι πετυχαίνει ακρίβεια 1 και στα δύο σύνολα (εκπαίδευσης, επικύρωσης) όταν gamma=0.1, C=10 και gamma=0.1, C=100, εφόσον για C=10 το μοντέλο πετυχαίνει

τέλεια ακρίβεια δεν υπάρχει λόγος να αυξηθεί η τιμή του C και να εισαχθεί περιττή πολυπλοκότητα στο μοντέλο. Για μικρότερες τιμές του c και όλες τις τιμές του gamma υπάρχει underfitting, καθώς έχει χαμηλή απόδοση τόσο στα δεδομένα εκπαίδευσης όσο και στα δεδομένα επικύρωσης. Ενώ για μεγαλύτερες τιμές του c και όλες τις υπόλοιπες του gamma υπάρχει overfitting, καθώς η ακρίβεια του συνόλου επικύρωσης είναι χαμηλότερη από την ακρίβεια του συνόλου εκπαίδευσης.

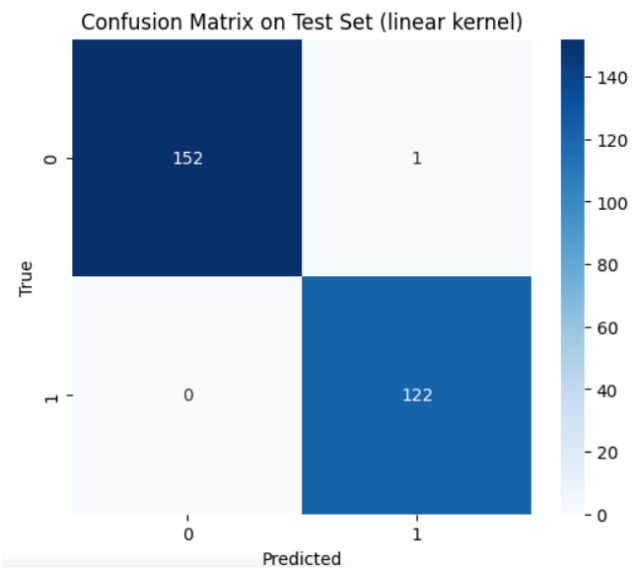
Από το confusion matrix προκύπτει ότι υπάρχουν, 152 σωστές προβλέψεις για την κλάση 0 και 122 σωστές προβλέψεις για την κλάση 1, με 0 false negatives και 0 false positives. Δεν υπάρχει κανένα λάθος άρα το μοντέλο γενικεύει σωστά.

5.2 Testing

Δημιουργείται η συνάρτηση **test_set_predictions(),** η οποία δέχεται σαν είσοδο ένα εκπαιδευμένο μοντέλο SVM (**model**), το test set (**X_test**), τις κατηγορίες του το test set (**y_test**) και τον πυρήνα του μοντέλου SVM (**kernel**). Με την συνάρτηση **predict()** προβλέπονται οι κατηγορίες για το σύνολο δοκιμής και με την συνάρτηση **accurancy_score()**, υπολογίζεται η ακρίβεια του μοντέλου συγκρίνοντας τις πραγματικές κατηγορίες με τις προβλεπόμενες. Επίσης, δημιουργείται το confusion matrix και υπολογίζεται το classification report. Με την συνάρτηση **predict_proba()** υπολογίζονται οι πιθανότητες για κάθε κατηγορία (0 και 1) και χρησιμοποιώντας την πιθανότητα της κατηγορίας 1 υπολογίζονται οι δείκτες **fpr** (ποσοστό false positive), **tpr** (ποσοστό true positive). Τέλος, υπολογίζεται και η AUC (εμβαδόν κάτω από την καμπύλη) και δημιουργείται η ROC curve.

Η συνάρτηση καλείται δύο φορές μια χρησιμοποιώντας ως όρισμα το γραμμικό μοντέλο (**linear_svm_model**) που έχει δημιουργηθεί και γραμμικό πυρήνα (**linear**) και μια χρησιμοποιώντας το μη γραμμικό μοντέλο (**rbf_svm_model**) που έχει δημιουργηθεί και μη γραμμικό (**rbf**) πυρήνα.

Αποτελέσματα Linear SVM:



Test Accuracy (linear kernel): 0.9963636363636363

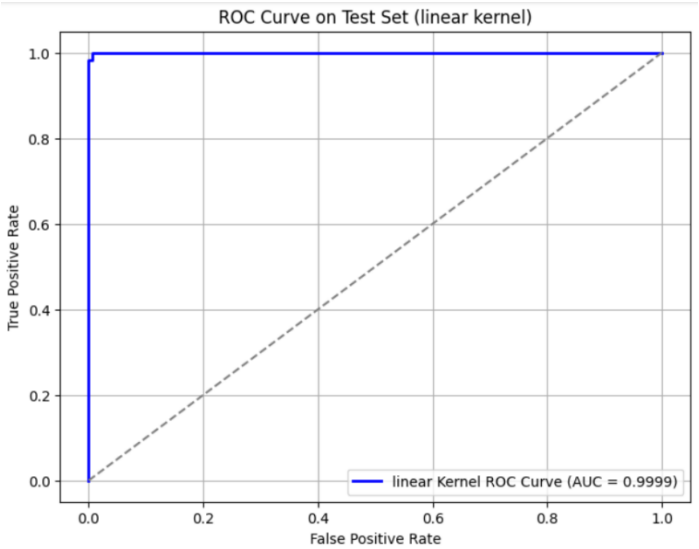
	precision	recall	f1-score	support
0	1.00	0.99	1.00	153
1	0.99	1.00	1.00	122
accuracy			1.00	275
macro avg	1.00	1.00	1.00	275
weighted avg	1.00	1.00	1.00	275

Από το confusion matrix προκύπτει ότι υπάρχουν, 152 σωστές προβλέψεις για την κλάση 0 και 122 σωστές προβλέψεις για την κλάση 1, με 0 false negatives και 1 false positives.

Σχολιασμός μετρικών:

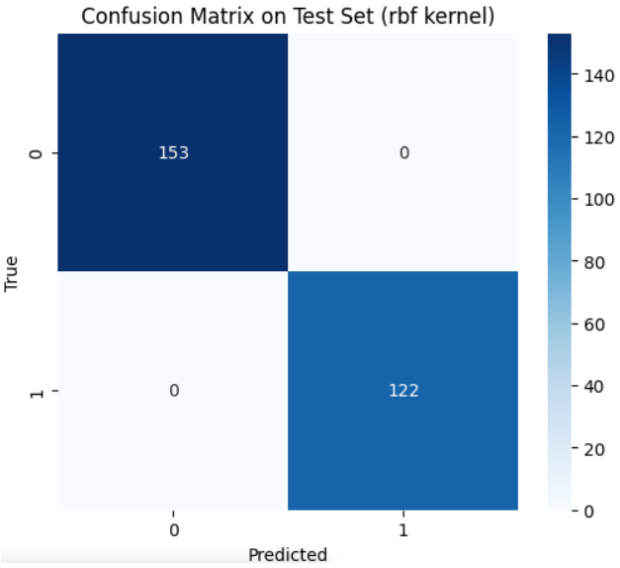
- **Accuracy:** 0.9963, υποδεικνύει εξαιρετική απόδοση του μοντέλου στη διάκριση των κατηγοριών στο test set.
- **Precision:** 1, αλλά υπάρχει 1 ψευδώς θετική πρόβλεψη.
- **Recall:** 1, το μοντέλο πέτυχε σχεδόν όλες τις προβλέψεις.
- **F1-Score:** 1,δείχνει τέλεια ισορροπία μεταξύ precision και recall.

Συμπερασματικά,το μοντέλο αποδίδει πολύ καλά στο test set. Η υψηλή απόδοση είναι συνεπής με την με την καλή απόδοση στο training και validation set, γεγονός που υποδηλώνει ότι το μοντέλο γενικεύει σωστά και δεν υπέστη overfitting.



Η AUC είναι **0.9999**, εξαιρετικά υψηλή τιμή, που υποδεικνύει σχεδόν τέλεια απόδοση στη διάκριση μεταξύ των θετικών και αρνητικών κλάσεων.

Αποτελέσματα Non-Linear SVM (RBF):



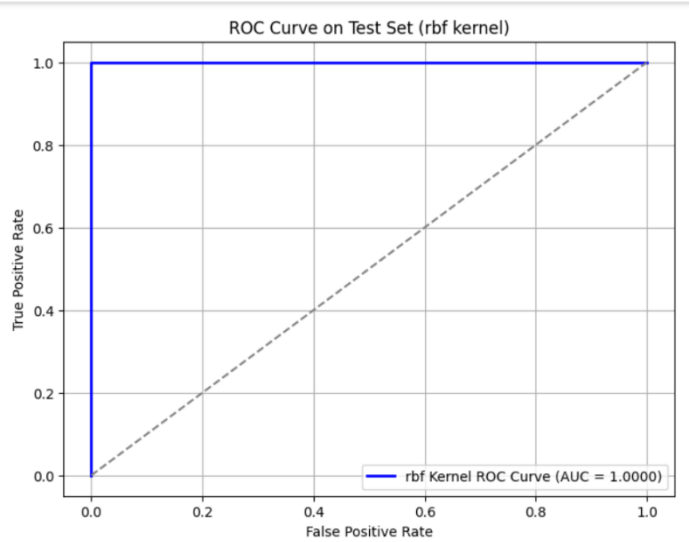

```
Test Accuracy (rbf kernel): 1.0
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       153
           1       1.00      1.00      1.00       122

    accuracy                           1.00       275
   macro avg       1.00      1.00      1.00       275
weighted avg       1.00      1.00      1.00       275
```

Από το confusion matrix προκύπτει ότι δεν υπάρχουν υπάρχουν λάθη στην ταξινόμηση, καθώς υπάρχουν 153 σωστές προβλέψεις για την κλάση 0 και 122 σωστές προβλέψεις για την κλάση 1, με 0 false negatives και 0 false positives.

Σχολιασμός μετρικών:

- **Accuracy:** 1.0, υποδεικνύει τέλεια απόδοση του μοντέλου στο test set.
- **Precision:** 1, δεν υπάρχουν ψευδώς θετικές προβλέψεις.
- **Recall:** 1, το μοντέλο πέτυχε όλες τις προβλέψεις.
- **F1-Score:** 1,δείχνει τέλεια ισορροπία μεταξύ precision και recall.

Συμπερασματικά,το μοντέλο αποδίδει τέλεια στο test set. Η άριστη απόδοση είναι συνεπής με την με την καλή απόδοση στο training και validation set, γεγονός που υποδηλώνει ότι το μοντέλο γενικεύει σωστά και δεν υπέστη overfitting.



Η AUC είναι **1**, που υποδεικνύει τέλεια απόδοση στη διάκριση μεταξύ των θετικών και αρνητικών κλάσεων στο σύνολο δοκιμής, χωρίς κανένα λάθος.

6. Συμπεράσματα

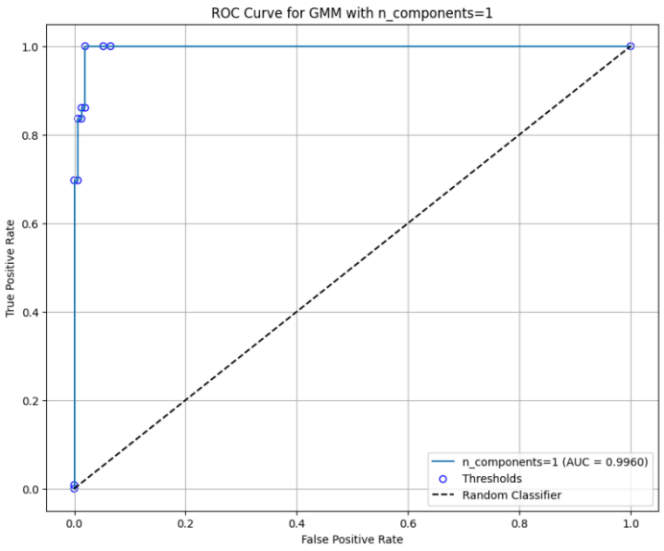
Συγκρίνοντας τόσο το **macro average F1-score** (υπολογίζει το F1-score για κάθε κλάση ξεχωριστά και παίρνει τον μέσο όρο τους), όσο και το **weighted average F1-score** (υπολογίζει τον μέσο όρο των F1-scores, σταθμισμένο με το πλήθος κάθε κλάσης) που προκύπτουν από το classification report για κάθε διαφορετική υλοποίηση (parzen windows με pca, k-NN, linear SVM, RBF SVM) παρατηρείται ότι το k-NN και τα SVM (linear και RBF kernel) επιτυγχάνουν άριστα αποτελέσματα με F1-score ίσα με 1.0.

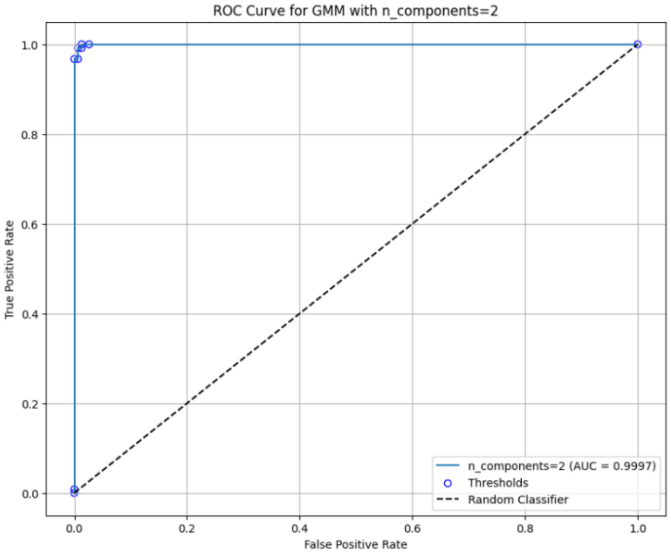
Αντίθετα, η μέθοδος Parzen Window με PCA, εμφανίζει σημαντικά χαμηλότερες επιδόσεις, F1-score 0.74 (macro avg), 0.75 (weighted avg) υποδηλώνοντας περιορισμούς στη διαχωριστική ικανότητα της μεθόδου. Το PCA πιθανώς να επηρέασε αρνητικά την ακρίβεια, καθώς κατά την εφαρμογή του μπορεί να αφαιρέθηκαν πληροφορίες που είναι σημαντικές για τον διαχωρισμό των κλάσεων.

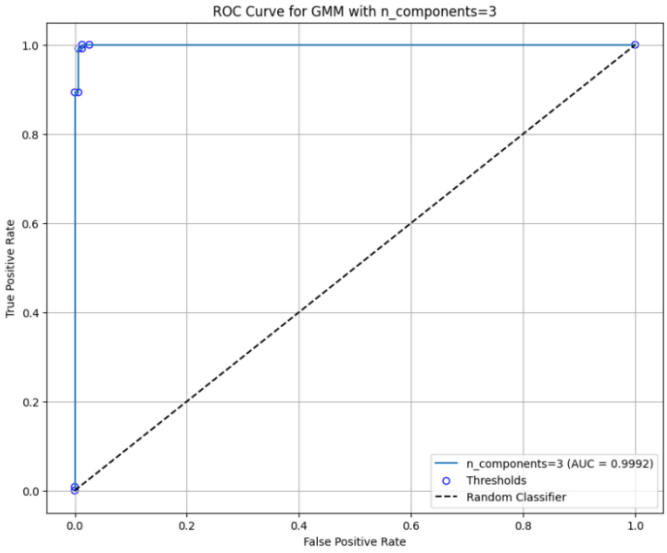
Τα αποτελέσματα των άλλων μεθόδων (k-NN και SVM) δείχνουν ότι τα δεδομένα χωρίς PCA επιτρέπουν την επίτευξη άριστης ακρίβειας και F1-score. Αυτό υποδηλώνει ότι οι κλάσεις ήταν σαφώς διαχωρίσιμες και τα χαρακτηριστικά ήταν εξαρχής κατάλληλα για την εκπαίδευση των μοντέλων.

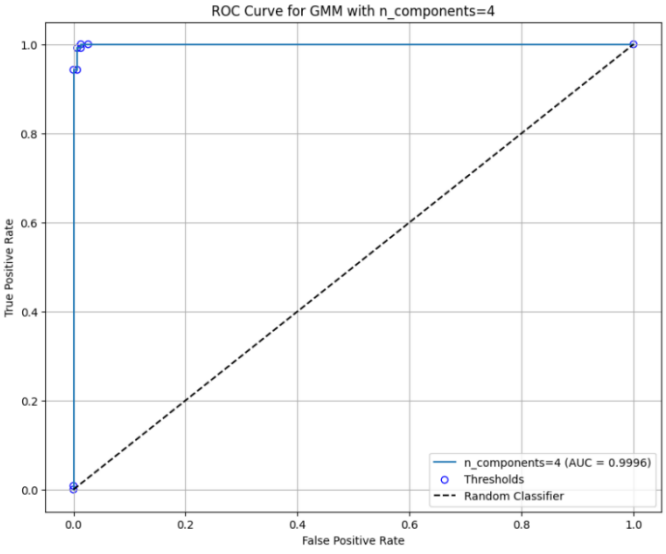
7. Gaussian Mixture

Επιλέγονται τα θετικά δείγματα (κλάση 1) για να εφαρμοστεί σε αυτά το gaussian mixture model. Επίσης η λίστα **n_components_list[]** περιέχει το διαφορετικό πλήθος gaussian componets που θα χρησιμοποιηθούν για αξιολόγηση. Στην συνέχεια, για κάθε αριθμό που περιέχεται στην λίστα δημιουργείται ένα μοντέλο gmm χρησιμοποιώντας την συνάρτηση **GaussianMixture()** με όρισμα τον αντίστοιχο αριθμό και με **random_state=42**. Το μοντέλο εκπαιδεύεται χρησιμοποιώντας την συνάρτηση **fit()** στα θετικά δείγματα. Επιπλέον, υπολογίζονται οι λογαριθμικές πιθανότητες για κάθε δείγμα του συνόλου δοκιμλης χρησιμοποιώντας την συνάρτηση **score_sample()** και εφαρμόζεται η εκθετική συνάρτηση για την μετατροπή των λογαρίθμων σε πιθανότητες. Ακόμη, υπολογίζονται οι τιμές **fpr**, **tpr, thesholds** (αναπαριστά τις τιμές του κατωφλίου που χρησιμοποιούνται για την ταξινόμηση στην θετική ή αρνητική κλάση) και το εμβαδόν κάτω από την καμπύλη AUC. Τέλος, δημιουργείται η ROC καμπύλη και εκυπώνεται ο αριθμός components που μεγιστοποιεί την τιμή AUC.










```
Best n_components: 2 with AUC = 0.9997
```

Από τις γραφικές παρατηρείται ότι το μοντέλο με components=2 παρουσιάζει το υψηλότερο AUC και η καμπύλη ROC βρίσκεται πολύ κοντά στο 1, γεγονός που υποδηλώνει τέλεια διάκριση μεταξύ των κατηγοριών. Οι διαφορές για μεγαλύτερο πλήθος components είναι μικρές, αλλά δεν βελτιώνουν την απόδοση και μπορεί να οδηγήσουν σε περιττή πολυπλοκότητα.