# 2η Εργασία
# Αναγνώριση Προτύπων
# Μπούρχα Ιωάννα 58019

Επιβλέπων καθηγητής: Ηλίας Θεοδωρακόπουλος

Ακαδημαϊκό έτος: 2023 - 2024

# ΠΕΡΙΕΧΟΜΕΝΑ

## ΑΣΚΗΣΗ 1

***Κατεβάστε από εδώ το αρχείο δεδομένων Data_ex1.txt που περιέχει δεδομένα δύο χαρακτηριστικών x = [$x_1$, $x_2$] από 3 κλάσεις ($\omega_1$, $\omega_2$, $\omega_3$). Κάθε γραμμή του αρχείου περιέχει δεδομένα στη μορφή: $x_1$, $x_2$, class_label .***

Προκειμένου να μην χρειάζεται κάθε φορά να φορτώνω χειροκίνητα το αρχείο που περιέχει τα δεδομένα, συνδέω το google colab με τον λογαριασμό μου στο google drive μέσω της βιβλιοθήκης `google.colab`. Με την εντολή `mount` φορτώνονται τα περιεχόμενα του Drive μου. Έπειτα, προσδιορίζω την διεύθυνση του αρχείου Data_ex1.txt, το οποίο και διαβάζω με την βιβλιοθήκη `numpy`, ώστε να το αντιμετωπίζω σαν πίνακα, ο οποίος περιέχει δεδομένα τύπου float. Σε διαφορετική περίπτωση, όπως αναφέρεται και στον κώδικα, αφαιρώ από τα σχόλια την τελευταία εντολή του πρώτου κελιού.

Αρχικά, διαχωρίζω τα δεδομένα σε ξεχωριστούς πίνακες ανάλογα με την κλάση στην οποία ανήκουν, δηλαδή ανάλογα με την τιμή της τρίτης στήλης του πίνακα. Οι ξεχωριστοί πίνακες έχουν δύο στήλες, καθώς έχουμε δύο χαρακτηριστικά x1 και x2, και τόσες γραμμές όσο το πλήθος των στοιχείων που ανήκουν στην αντίστοιχη κλάση. Για την συγκεκριμένη άσκηση, όλες οι κλάσεις αποτελούνται από 100 στοιχεία.

***Ερώτημα Α:***

***Υπολογίστε κατάλληλο κώδικα ώστε να εκτιμήσετε τις πυκνότητες πιθανότητας $p(x|\omega_1)$ , $p(x|\omega_2)$ και $p(x|\omega_3)$ με τη μέθοδο παραθύρων Parzen, για συνάρτηση παραθύρου:***

$$\varphi(x - x_i) = \frac{1}{h\sqrt{2\pi}} e^{-\frac{\|x-x_i\|_2^2}{2h^2}}$$

***και για h = 0.3 να απεικονίσετε κατάλληλα τις κατανομές σε κοινό γράφημα.***

Στόχος του ερωτήματος είναι να προσδιορίσω την κατανομή των στοιχείων κάθε κλάσης με μη παραμετρικές τεχνικές. Η προσέγγιση μιας άγνωστης συνάρτησης πυκνότητας πιθανότητας (σ.π.π.) γίνεται αξιοποιώντας το ιστόγραμμα. Για την μονοδιάσταση περίπτωση, ο άξονας των στοιχείων (x) διαιρείται σε διαδοχικές περιοχές μήκους h. Για κάθε περιοχή εκτιμάται η πιθανότητα ένα στοιχείο x να βρίσκεται σε αυτήν.

Έστω N το συνολικό πλήθος των στοιχείων και k το πλήθος των στοιχείων που βρίσκονται εντός μίας περιοχής. Η σ.π.π. θεωρείται σταθερή σε όλο το εύρος της περιοχής και ισούται με:

$$p(x) = \frac{k}{hN}$$

Στην πολυδιάσταση περίπτωση ο χώρος d διαιρείται σε υπεκύβους μήκους ακμής h και όγκου $h^d$. Το πλήθος των σημείων k που βρίσκονται εντός του υπερκύβου καθορίζεται από την συνάρτηση φ(). Η σ.π.π. τότε ισούται με:

$$p(x) = \frac{1}{h^d N} \sum_{i=1}^{N} \varphi(x_i - x) \text{, όπου } \varphi(\cdot) = \begin{cases} 1, & \text{εάν βρίσκεται εντός του υπερκύβου} \\ 0, & \text{εάν βρίσκεται εκτός του υπερκύβου} \end{cases}$$

Θεωρούμε ότι ο υπερκύβος έχει κέντρο το x.

Επομένως, η εκτίμηση της συνεχούς συνάρτησης σ.π.π. γίνεται μέσω ενός αναπτύγματος όρων μη συνεχών βηματικών συναρτήσεων φ(). Στην προσπάθειά του να γενικεύσει τον παραπάνω τύπο ο Parzen χρησιμοποίησε ομαλές συναρτήσεις στην θέση της φ.

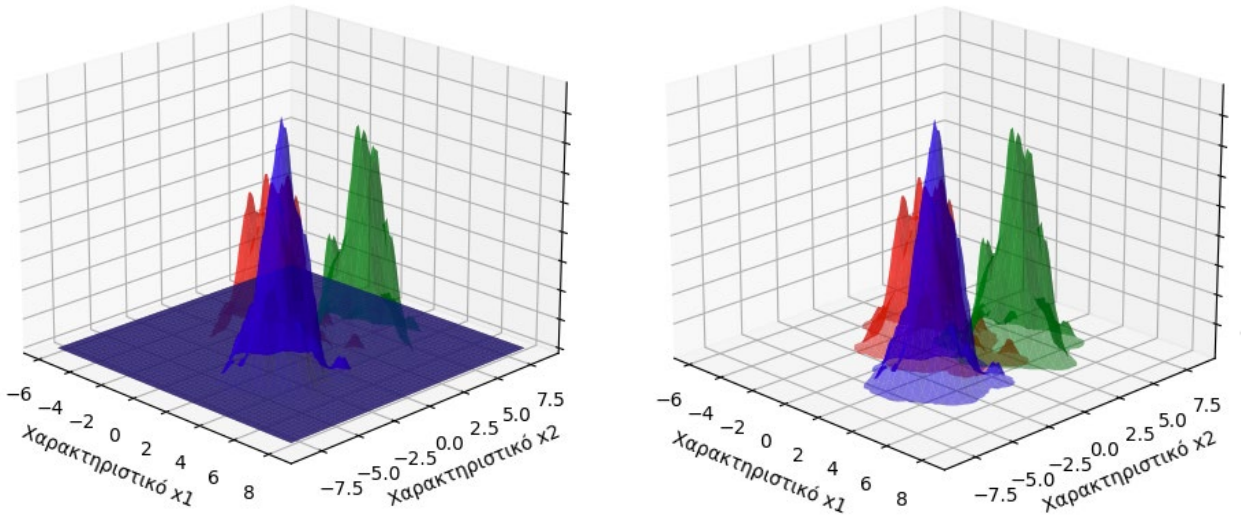
$$\int_x \varphi(x)\, dx = 1 \quad \text{και} \quad \varphi(x) \geq 0$$

Οι ομαλές αυτές συναρτήσεις είναι γνωστές ως παράθυρα Parzen. Ένα χαρακτηριστικό παράδειγμα είναι το παράθυρο Gauss. Η άγνωστη σ.π.π. προσεγγίζεται ως μέσος όρος N Gaussian συναρτήσεων που κάθε μία από αυτές έχει ως κέντρο ένα διαφορετικό σημείο.

Συνοπτικά, για κάθε σημείο του test set και με κέντρο αυτό τοποθετείται ένα παράθυρο καθορισμένου πλάτους h. Υπολογίζεται το άθροισμα των δεδομένων εκπαίδευσης που βρίσκονται εντός του παραθύρου και το σημείο του test set ταξινομείται στην κλάση με το μεγαλύτερο άθροισμα. Ο όγκος παραμένει σταθερός ενώ το πλήθος των σημείων k ποικίλει. Οι παράμετροι που καθορίζονται από τον χρήστη είναι το μήκος του παραθύρου h και φυσικά το σύνολο των στοιχείων εκπαίδευσης.

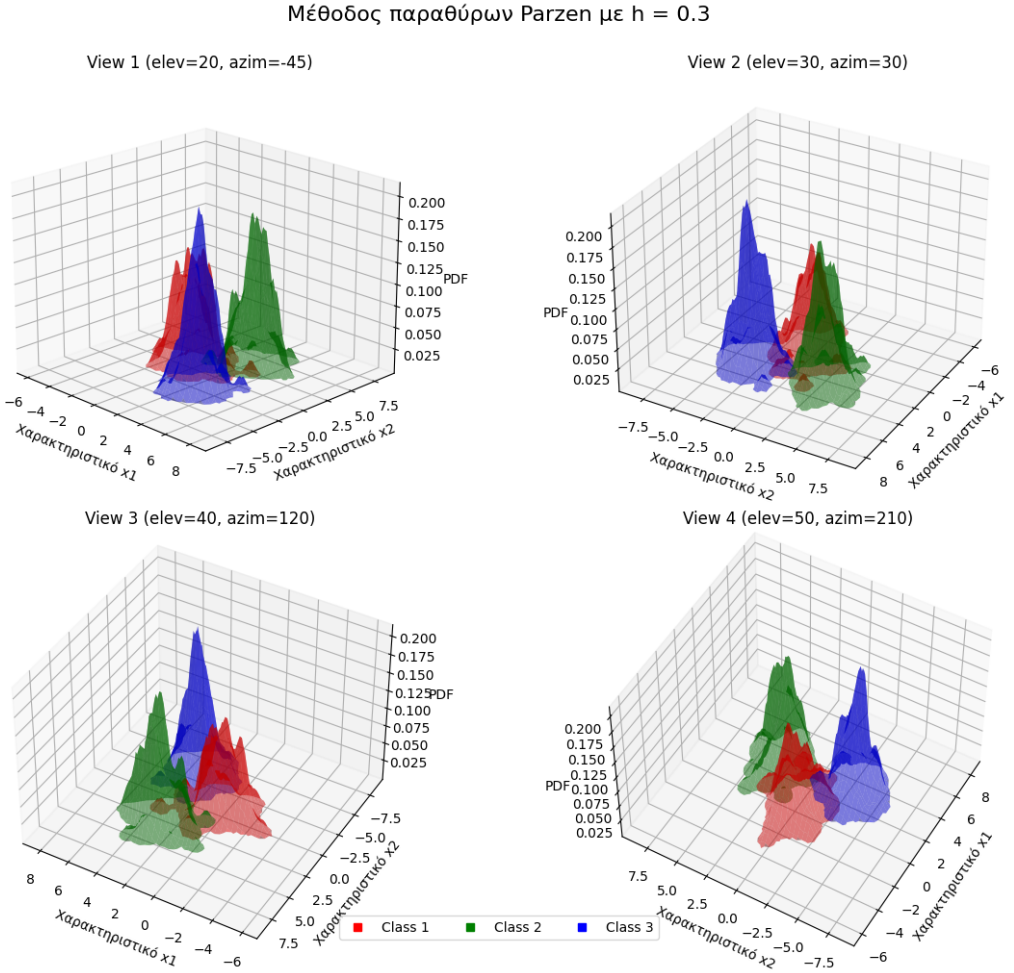
*Μη διαδραστικό διάγραμμα:*

Κάθε κατανομή απεικονίζεται ως μία επιφάνεια. Οι επιφάνειες τοποθετούνται διαδοχικά η μία επάνω στην άλλη με την σειρά που δηλώθηκαν. Απόρροια αυτού είναι να γίνονται αλληλοεπικαλύψεις για τις χαμηλές τιμές της τρέχουσας επιφάνειας με τις μεγαλύτερες τιμές των προηγούμενων. Για να αντιμετωπίσω το πρόβλημα αυτό όρισα ένα threshold για τις τιμές οι οποίες πρόκειται να απεικονιστούν. Η τιμή του threshold είναι ξεχωριστή για κάθε κατανομή και καθορίζεται από την μικρότερη τιμή των δεδομένων της προσαυξημένη με μία μικρή τιμή ανοχής, η τιμή της οποίας καθορίζεται αυθαίρετα.



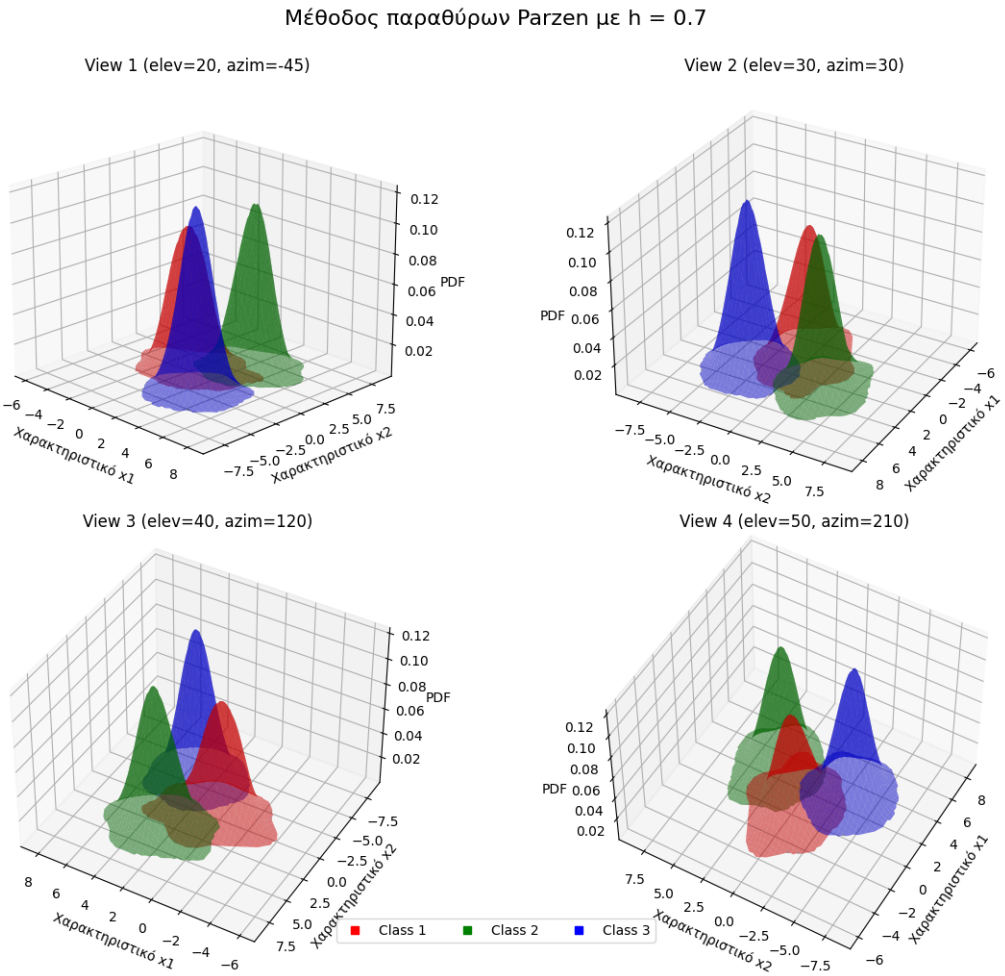

*Εικόνα 1: (Αριστερά) Απεικόνιση χωρίς threshold, (Δεξιά) Απεικόνιση με threshold.*

Αποφάσισα να απεικονίσω το διάγραμμα τέσσερις φορές με διαφορετικές γωνίες θέασης προκειμένου να έχω μία πιο αντιπροσωπευτική αντίληψη των κατανομών. Οι τιμές των γωνιών θέασης, επίσης, είναι αυθαίρετες.




*Εικόνα 2: Τριδιάστατο μη διαδραστικό διάγραμμα με threshold σε διαφορετικές γωνίες θέασης για h=0.3*

***Δοκιμάστε να κάνετε εκτίμηση για h=0.7 και h=0.1. Τι παρατηρείτε?***




*Εικόνα 3: Τριδιάστατο μη διαδραστικό διάγραμμα με threshold σε διαφορετικές γωνίες θέασης για h=0.7*

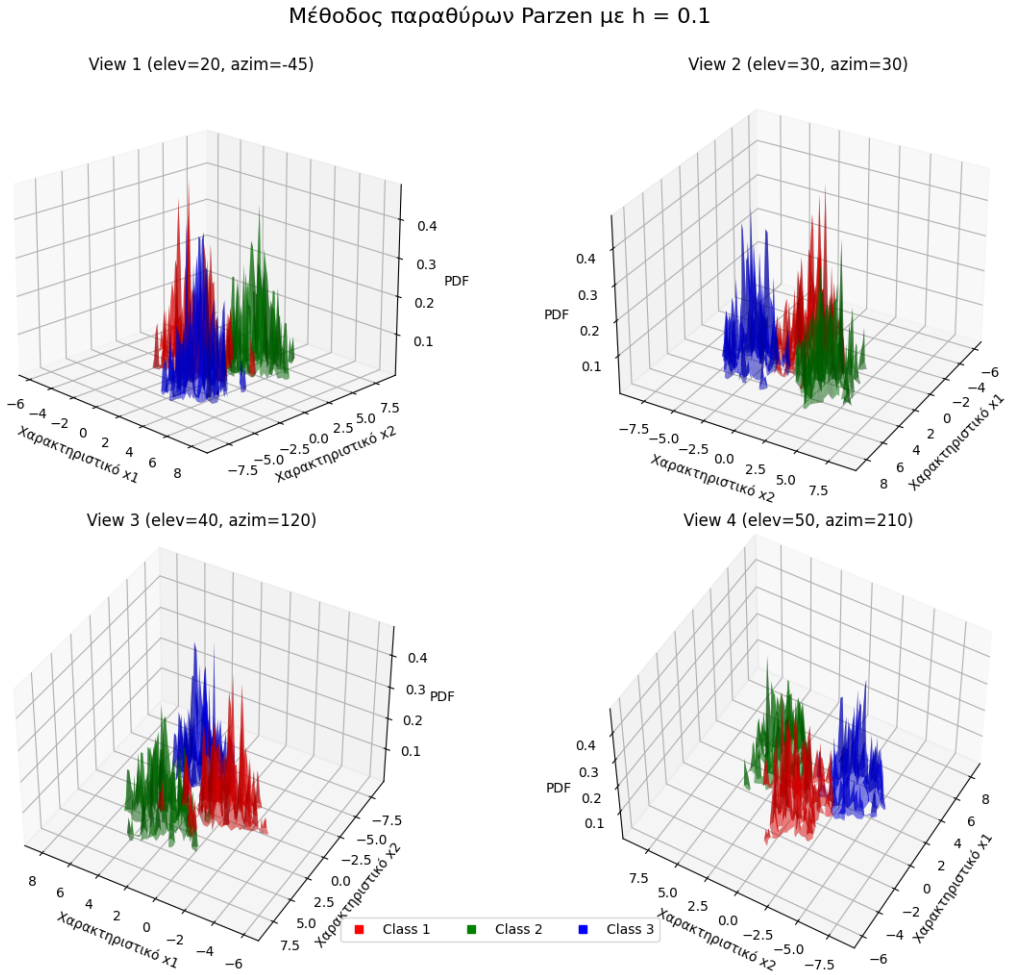
Εφόσον αυξάνεται η τιμή του h, το παράθυρο γίνεται πιο φαρδύ, δηλαδή περιλαμβάνονται περισσότερα στοιχεία στον υπολογισμό της σ.π.π. για κάθε σημείο x. Η σ.π.π. προσεγγίζεται από ένα πεπερασμένο πλήθος συναρτήσεων μεγάλης διασποράς με κέντρα τα σημεία του συνόλου εκπαίδευσης. Απόρροια αυτού είναι να έχω πιο ομαλές εκτιμήσεις των πυκνοτήτων χωρίς θόρυβο, όπως φαίνεται και στα παραπάνω διαγράμματα. Ωστόσο, χάνεται η τοπική δομή των δεδομένων. Τα θεωρητικά αυτά συμπεράσματα επιβεβαιώνονται από την εικόνα 3.

Στο σημείο αυτό τονίζεται ότι η αύξηση του h μπορεί να οδηγήσει σε υπερβολικά λείες εκτιμήσεις αγνοώντας την δομή των δεδομένων με συνέπεια την υποεκτίμηση των κορυφών των κατανομών (underfitting).

Εφόσον μειώνεται η τιμή του h, το παράθυρο γίνεται πιο στενό, δηλαδή περιλαμβάνονται λιγότερα στοιχεία στον υπολογισμό της σ.π.π. για κάθε x. Για πολύ μικρές τιμές μάλιστα, η σ.π.π. προσεγγίζεται από ένα πεπερασμένο πλήθος αιχμηρών συναρτήσεων τύπου δέλτα με κέντρα τα σημεία του συνόλου εκπαίδευσης. Έτσι, καθώς κινούμαστε μέσα στον χώρο, η απόκριση της πιθανότητας είναι πολύ υψηλή κοντά στα στοιχεία του συνόλου εκπαίδευσης. Απόρροια αυτού είναι να έχω πιο ακριβές σχήμα των κατανομών με ανίχνευση λεπτομερειών στα δεδομένα. Τα θεωρητικά αυτά συμπεράσματα επιβεβαιώνονται από την εικόνα 4.

Στο σημείο αυτό τονίζεται ότι η μείωση του h μπορεί να οδηγήσει σε υπερβολικά ανώμαλες εκτιμήσεις που προκύπτουν από τον θόρυβο στα δεδομένα (overfitting).

Συνεπώς, η επιλογή κατάλληλης τιμής για το h είναι ζωτικής σημασίας. Παρόλο που έχουν προταθεί αρκετές προσεγγίσεις στη βιβλιογραφία, ένας απλός τρόπος είναι να ξεκινήσουμε με μια αρχική τιμή την οποία θα τροποποιούμε επαναληπτικά προκειμένου να ελαχιστοποιήσουμε το προκύπτον σφάλμα ταξινόμησης. Το σφάλμα αυτό εκτιμάται μέσω κατάλληλου χειρισμού του συνόλου εκπαίδευσης.




*Εικόνα 4: Τριδιάστατο μη διαδραστικό διάγραμμα με threshold σε διαφορετικές γωνίες θέασης για h=0.1*

***Εάν είχατε στη διάθεσή σας μόνο το 25% των δεδομένων, τι τιμή θα έπρεπε να έχει το h για να κάνετε εκτίμηση με παρόμοια λεπτομέρεια με την αρχική? Επιβεβαιώστε την απάντησή σας πειραματικά.***

Εφόσον έχω στην διάθεσή μου μονάχα το 25%, θα πρέπει να αφαιρέσω το υπόλοιπο 75%. Προκειμένου να διατηρήσω την ίδια αναλογία των δεδομένων κάθε κλάσης, δηλαδή κάθε κλάση να συνεχίσει να αντιπροσωπεύει το ίδιο κομμάτι του συνόλου δεδομένων. Με αυτόν τον τρόπο, αποφεύγεται επιπλέον και η υπερπροσαρμογή (overfitting) σε μία συγκεκριμένη κλάση η οποία θα περιέχει περισσότερα στοιχεία έναντι των υπολοίπων κλάσεων. Στον κώδικα, τα δεδομένα που τελικά παραμένουν προσδιορίζονται από την γραμμή στην οποία βρίσκονται. Η τιμή της γραμμής παράγεται με τυχαίο τρόπο.

Εφόσον μειώνεται το πλήθος των δειγμάτων (N), η επιρροή κάθε δεδομένου στην εκτιμώμενη κατανομή αυξάνεται, επειδή υπάρχουν λιγότερα σημεία δεδομένων για να "γεμίσουν" τον ίδιο χώρο. Αυτό σημαίνει ότι το h θα πρέπει να αυξηθεί για να διατηρηθεί η ίδια επίδραση στην εκτιμώμενη κατανομή. Για μία πρώτη εκτίμηση της νέας τιμής του h επιλέγω να χρησιμοποιήσω τον γενικευμένο κανόνα του Silverman (rule of thumb):

$$\mathrm{h} = 1.06 * \sigma * N^{\frac{-1}{5}}$$

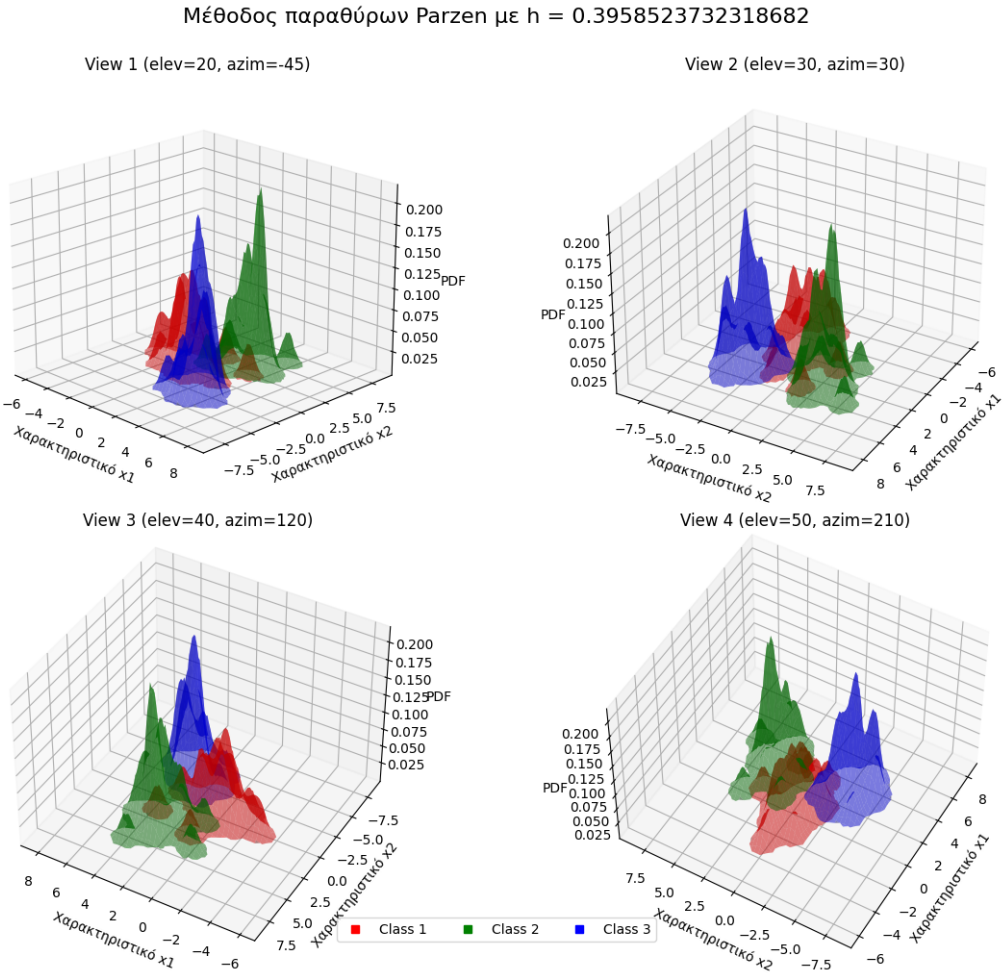
όπου: $\sigma$ Διακύμανση
$N$ Πλήθος δειγμάτωνh

Για το συγκεκριμένο πρόβλημα έχουμε:

$$\frac{h_{old}}{h_{new}} = \frac{1.06 * \sigma * 1^{\frac{-1}{5}}}{1.06 * \sigma * 0.25^{\frac{-1}{5}}} \Leftrightarrow h_{new} = 0.3 * 0.25^{-0.2} = 0.3959$$

Το αποτέλεσμα φαίνεται στην εικόνα 5. Εφόσον η τιμή αυτή είναι ενδεικτική δοκίμασα και με παραπλήσιες τιμές. Το αποτέλεσμα ωστόσο του κανόνα Silverman θεώρησα πως ήταν το καλύτερο. Όπως ήταν αναμενόμενο οι κατανομές μοιάζουν με τις αρχικές, αλλά δεν είναι ίδιες, καθώς κάποια από τα δεδομένα που

προσδιορίζουν την τοπική δομή έχουν αφαιρεθεί. Σημειώνω ότι τα αποτελέσματα εξαρτώνται από τα δεδομένα που τελικώς παραμένουν στην κλάση και προκύπτουν με τυχαίου τρόπο, διαφορετικό για κάθε εκτέλεση του προγράμματος.




*Εικόνα 5: Τριδιάστατο μη διαδραστικό διάγραμμα με threshold σε διαφορετικές γωνίες θέασης για h=0.3958*

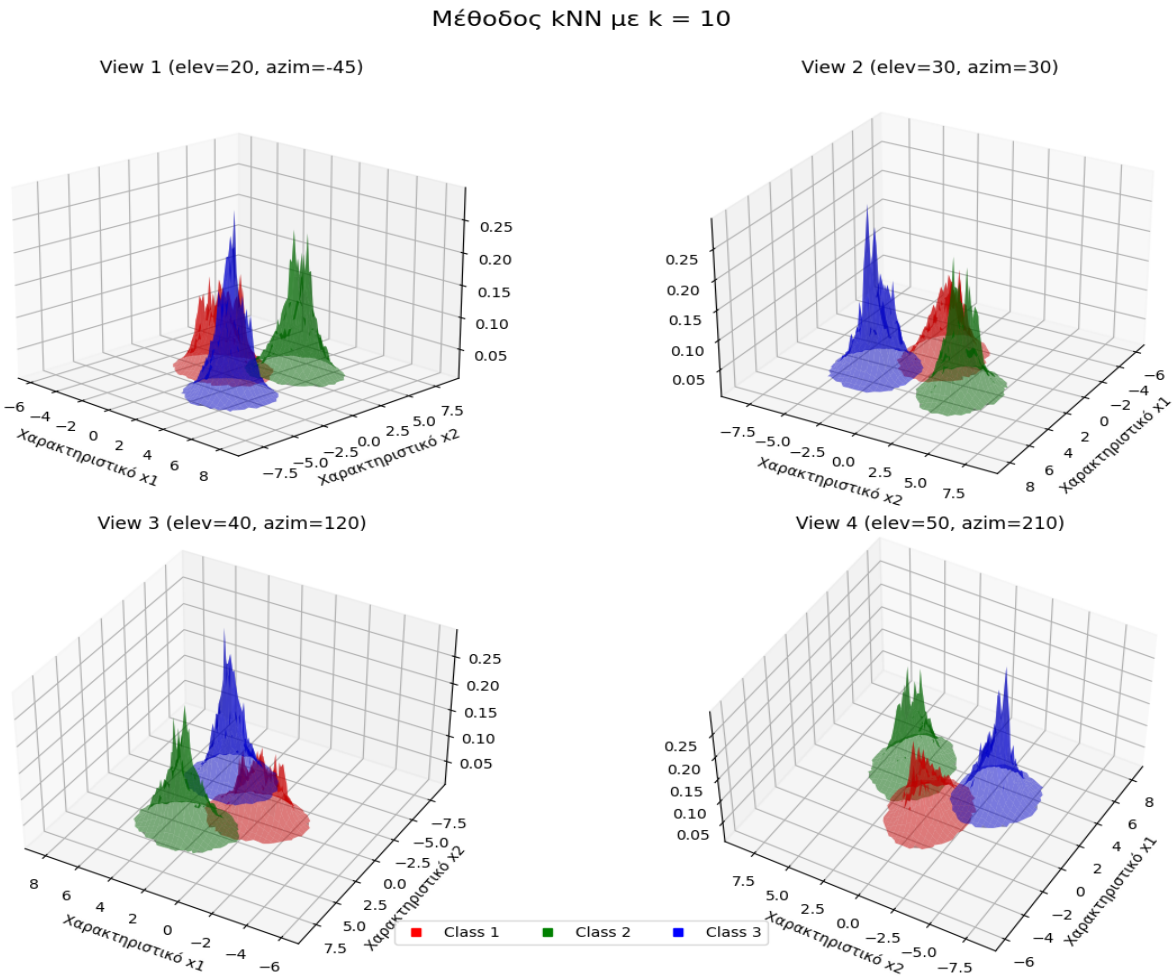
***Ερώτημα Β***

***Υλοποιείστε κατάλληλο κώδικα ώστε να εκτιμήσετε τις πυκνότητες πιθανότητας $p(x|\omega_1)$ , $p(x|\omega_2)$ και $p(x|\omega_3)$ με τη μέθοδο k-NN, και για k=10 να απεικονίσετε κατάλληλα τις κατανομές σε κοινό γράφημα.***

Όπως αναφέρθηκε, στην μέθοδο παραθύρων Parzen, διατηρείται σταθερός ο όγκος του κύβου και μεταβάλλεται το πλήθος των στοιχείων k του συνόλου εκπαίδευσης που βρίσκονται εντός αυτού με βάση τα οποία καθορίζεται η ταξινόμηση. Είναι φανερό λοιπόν, πώς μία άλλη εκδοχή του προσδιορισμού των κατανομών με μη παραμετρικές τεχνικές είναι να κάνω το αντίστροφο, δηλαδή να διατηρήσω σταθερό το πλήθος k των στοιχείων εκπαίδευσης που καθορίζουν την ταξινόμηση και να μεταβάλλω τον όγκο. Έτσι, σε περιοχές μικρής πυκνότητας ο όγκος θα είναι μεγάλος ενώ σε περιοχές υψηλής πυκνότητας ο όγκος θα είναι μικρός. Η μέθοδος αυτή είναι γνωστή ως kNN (k Nearest Neighbors).

Για κάθε σημείο του test set υπολογίζεται η απόσταση του από όλα τα σημεία στοιχεία του συνόλου εκπαίδευσης. Επομένως, για μεγάλα datasets ενδέχεται να απαιτείται μεγάλη υπολογιστική ισχύς. Η απόσταση αυτή συνήθως είναι η Ευκλείδεια, οπότε πλέον δεν έχω υπερκύβο αλλά υπερσφαίρα. Από αυτά, κρατάω τα k στοιχεία με την μικρότερη απόσταση και ταξινομώ το στοιχείο του test set στην κλάση της οποίας τα στοιχεία είναι περισσότερο εντός του k. Οι παράμετροι που καθορίζονται από τον χρήστη είναι το πλήθος τον κοντινότερων γειτόνων k που καθορίζουν την ταξινόμηση.

Αναφορικά με την υλοποίηση του αλγορίθμου αυτού σε κώδικα μία προσέγγιση είναι να αξιοποιήσω την βιβλιοθήκη `sklearn.neighbors` προκειμένου να προσδιορίσω τα k στοιχεία που βρίσκονται πιο κοντά. Κατά την εκτέλεση του προγράμματος παρατήρησα ότι απαιτείται αρκετός χρόνος λόγω της επαναληπτικής διαδικασίας για την εκτίμηση των κοντινότερων γειτόνων, οπότε επιχείρησα να βρω πιο αποδοτική μέθοδο. Συγκεκριμένα, χρησιμοποίησα την βιβλιοθήκη `scipy.spatial`.

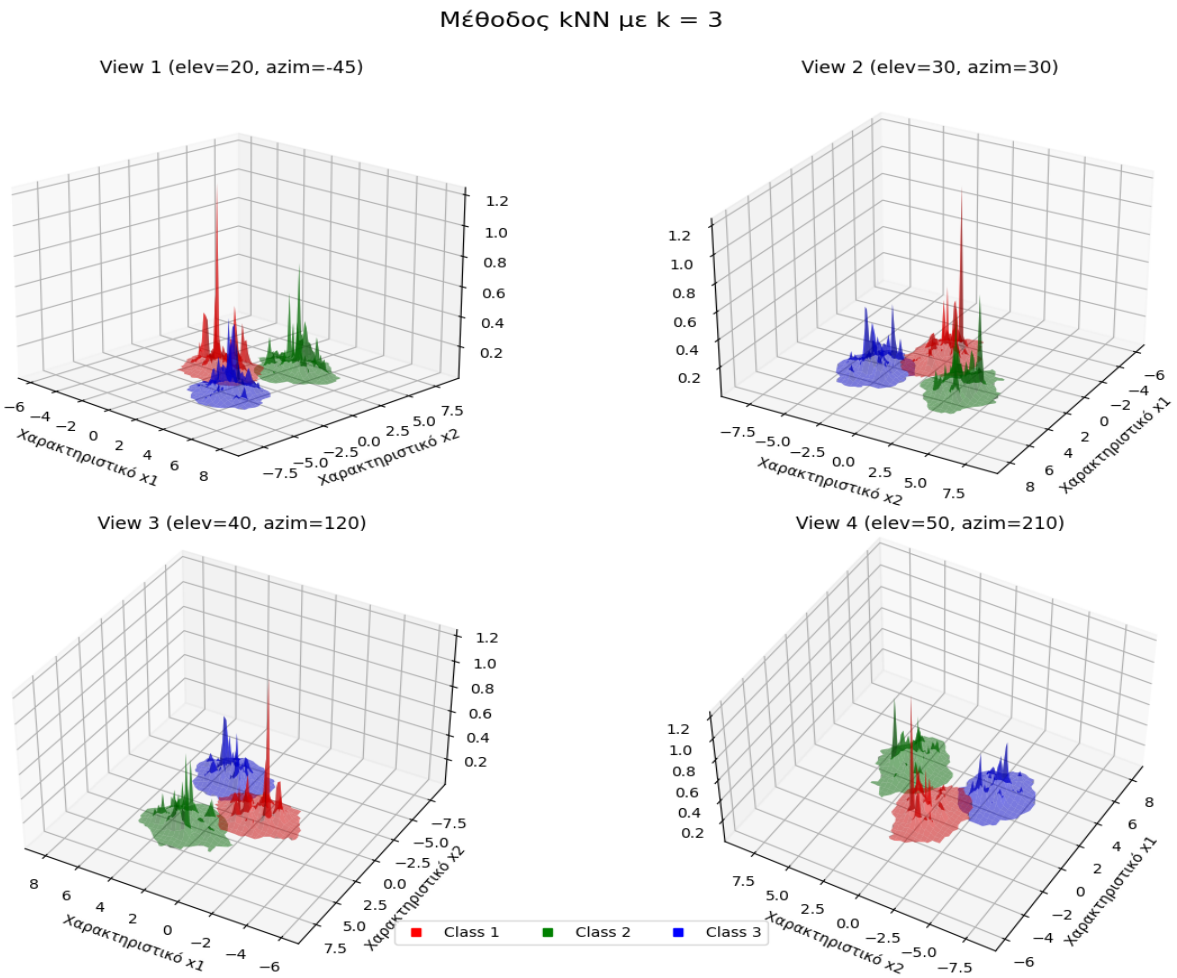



*Εικόνα 6: Τριδιάστατο μη διαδραστικό διάγραμμα με threshold σε διαφορετικές γωνίες θέασης για k = 10*

***Επαναλάβετε το προηγούμενο βήμα για k = 3 και για k = 30. Τί παρατηρείτε;***

Η μείωση της τιμής του k συνεπάγεται τον περιορισμό των κοντινότερων γειτόνων, με αποτέλεσμα να περιλαμβάνονται λιγότερα δεδομένα στον υπολογισμό της πυκνότητας πιθανότητας. Απόρροια αυτού είναι η αύξηση της ευαισθησίας του αλγορίθμου ως προς την τοπική δομή των δεδομένων (μείωση του bias), καθώς αυξάνεται η διακύμανση. Με άλλα λόγια η πρόβλεψη προσαρμόζεται καλύτερα στις τοπικές περιοχές.

Ωστόσο, η πολύ μικρή τιμή του k, υπερευαισθητοποιεί τον αλγόριθμο στα δεδομένα (τα οποία ενδέχεται να περιέχουν και θόρυβο), με αποτέλεσμα να οδηγούμαστε σε υπερβολική προσαρμογή (overfitting, δεν μπορεί να γενικεύσει). Επιπλέον, για την περίπτωση που τα δεδομένα μας εμπεριέχουν θόρυβο, η ταξινόμηση ενός σημείου ενδέχεται να καθοριστεί από ένα δείγμα outlier.

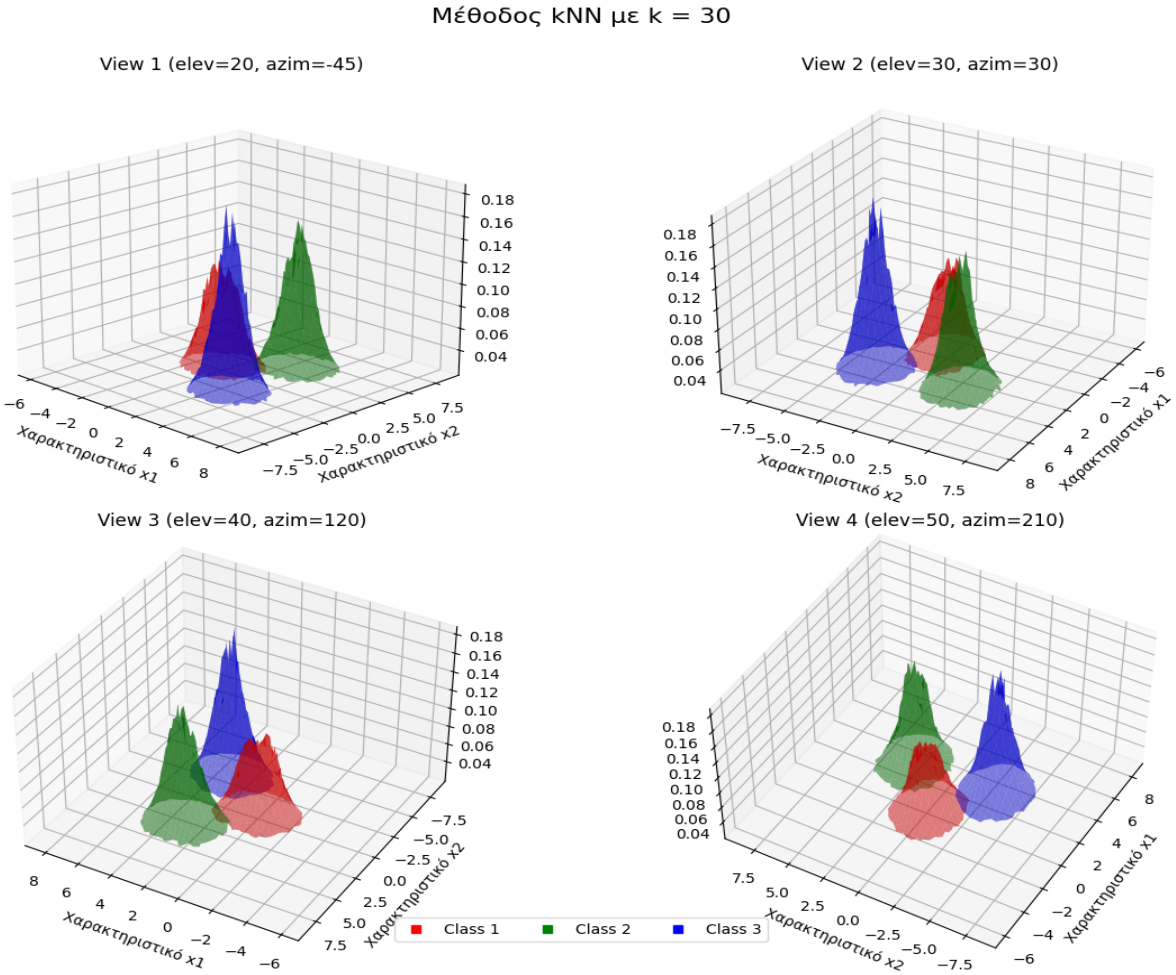



*Εικόνα 7: Τριδιάστατο μη διαδραστικό διάγραμμα με threshold σε διαφορετικές γωνίες θέασης για k = 3*

Αντίθετα, η αύξηση της τιμής του k συνεπάγεται την διεύρυνση των κοντινότερων γειτόνων, με αποτέλεσμα να περιλαμβάνονται περισσότερα δεδομένα στον υπολογισμό της πυκνότητας πιθανότητας. Απόρροια αυτού είναι η μείωση της ευαισθησίας του αλγορίθμου ως προς την τοπική δομή των δεδομένων (αύξηση του bias), καθώς μειώνεται η διακύμανση. Σε περίπτωση που τα δεδομένα έχουν θόρυβο, αυτός αγνοείται καθώς η απόφαση βασίζεται σε περισσότερα δείγματα. Με άλλα λόγια η πρόβλεψη προσεγγίζει μία πιο απλή δομή.

Ωστόσο, η πολύ μεγάλη τιμή του k, υπεργενικεύει τον αλγόριθμο, αφού χάνεται η ιδιότητα αναγνώρισης της πολυπλοκότητας των δεδομένων (αγνοείται η τοπική δομή). Έτσι, οδηγούμαστε σε underfitting (δεν μπορεί να εξειδικεύσει).

Κατά συνέπεια, η επιλογή κατάλληλης τιμή του k εξαρτάται από τη φύση των δεδομένων και καθορίζεται ύστερα από αρκετές δοκιμές. Στόχος είναι να βρεθεί μία τιμή k η οποία θα ισορροπεί μεταξύ underfitting και overfitting.

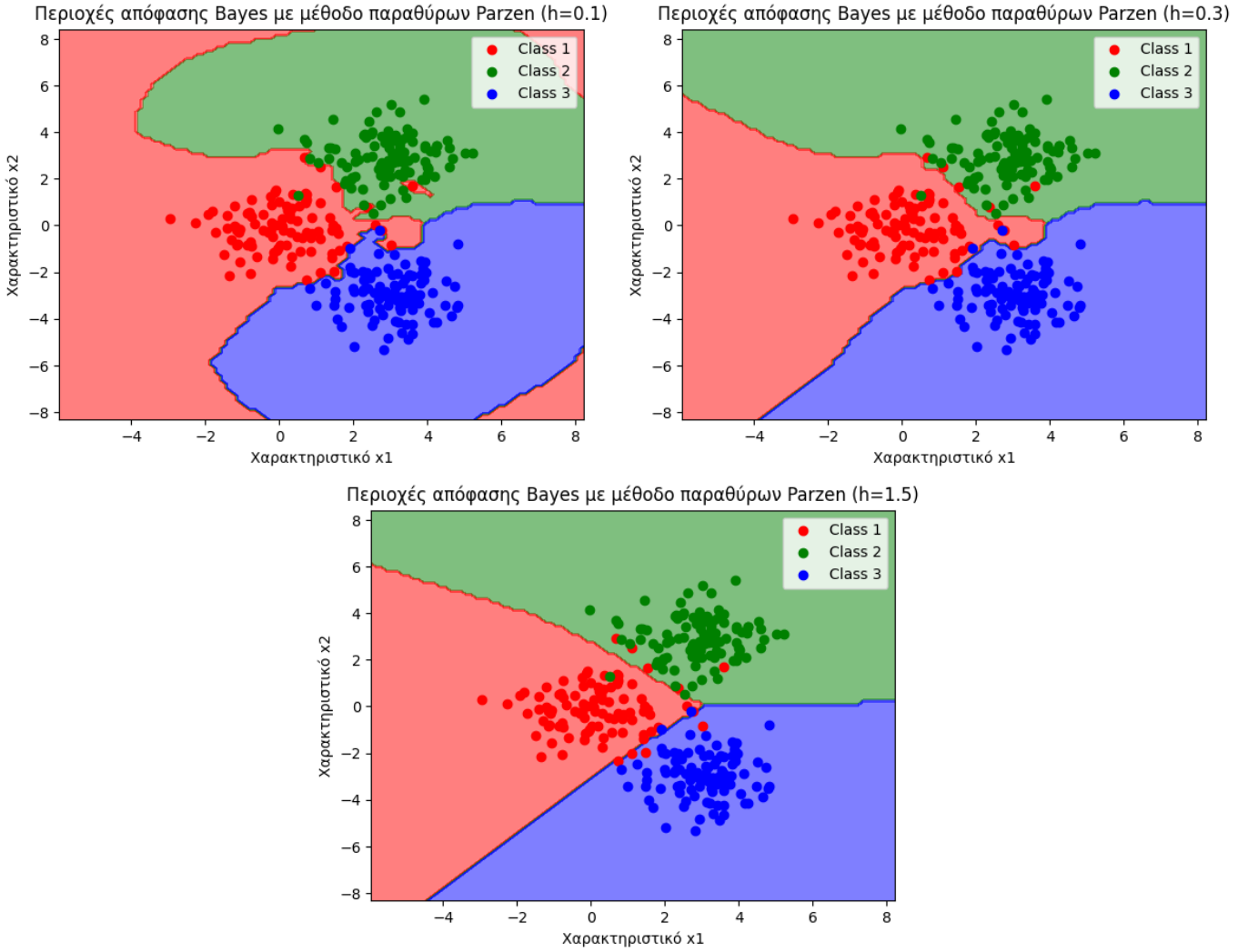



*Εικόνα 8: Τριδιάστατο μη διαδραστικό διάγραμμα με threshold σε διαφορετικές γωνίες θέασης για k = 30*

***Ερώτημα Γ***

***Θεωρώντας ότι όλες οι κλάσεις έχουν ίδιες apriori πιθανότητες, να απεικονίσετε τα δεδομένα και τις περιοχές απόφασης για ταξινόμηση σύμφωνα με τον κανόνα του Bayes και κατανομές που εκτιμώνται με τη μέθοδο των παραθύρων parzen για h =0.1, h = 0.3, h = 1.5. Πώς επηρεάζονται οι περιοχές απόφασης από την παράμετρο παραθύρου;***

Αρχικά, θα πρέπει να υπολογίσω την πυκνότητα πιθανότητας με την μέθοδο παραθύρων Parzen για κάθε τιμή της παραμέτρου h. Σύμφωνα με τον κανόνα του Bayes, όπως είδαμε και στην προηγούμενη εργασία, το στοιχείο ταξινομείται στην κλάση με την υψηλότερη εκ των υστέρων πιθανότητα (posterior). Εφόσον όλες οι κλάσεις έχουν την ίδια εκ των προτέρων πιθανότητα (a priori), η posterior καθορίζεται από την πιθανοφάνεια (likelihood), δηλαδή την κατανομή $p(x|\omega_i)$. Άρα, η κλάση στην οποία ανήκει τελικά το στοιχείο είναι εκείνη που παρουσιάζει την μεγαλύτερη τιμή πιθανοφάνειας. Τα αποτελέσματα φαίνονται στην εικόνα 9.



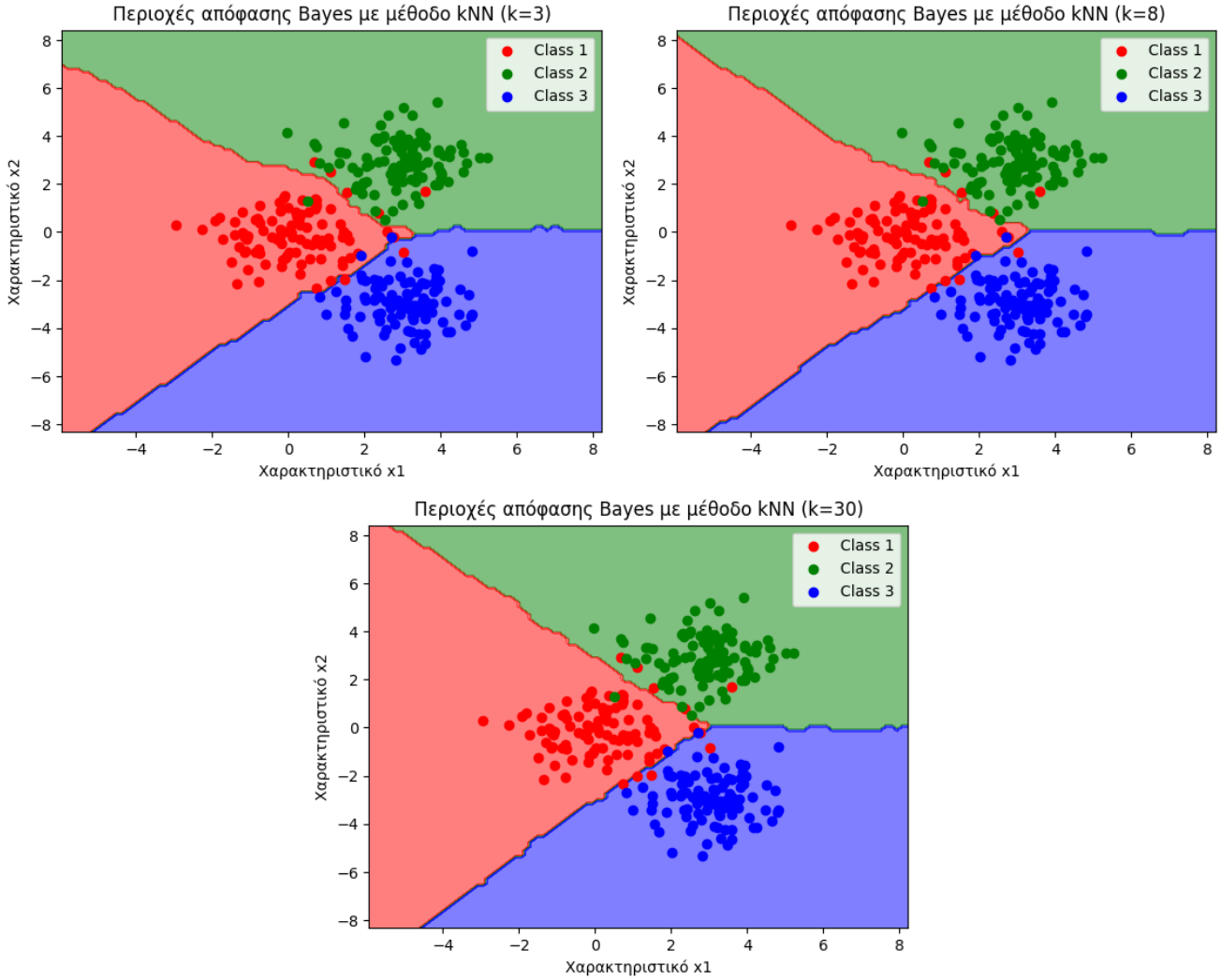

*Εικόνα 9: Συγκεντρωμένα αποτελέσματα για μέθοδο Parzen με διαφορετικά h*

Όπως φαίνεται και από την εικόνα 9 ο παράγοντας h παίζει κρίσιμο ρόλο στην εκτίμηση της σ.π.π. για τα δεδομένα. Επομένως, επηρεάζει άμεσα τις περιοχές απόφασης οι οποίες με την σειρά τους καθορίζουν το πώς ταξινομούνται τα στοιχεία του test set. Όπως προαναφέρθηκε, για μικρό h η εκτίμηση της σ.π.π. γίνεται πιο ευαίσθητη στα τοπικά χαρακτηριστικά, με αποτέλεσμα οι περιοχές απόφασης να γίνονται πολύ ακανόνιστες περικλείοντας μεμονωμένα στοιχεία ή μικρές ομάδες στοιχείων, αντικατοπτρίζοντας την υψηλή διακύμανση στην εκτίμηση της σ.π.π. Αντίθετα, για μεγάλες τιμές του h η εκτίμηση γίνεται ανθεκτική στις τοπικές διακυμάνσεις ομαλοποιώντας την εκτίμηση της σ.π.π. Οι περιοχές απόφασης γίνονται πιο ομαλές αφού παρουσιάζεται μικρότερη ευαισθησία στον θόρυβο ή σε outliers. Τονίζεται πως για ακραίες τιμές του h εγκυμονεί ο κίνδυνος για over- ή underfitting αντίστοιχα. Συμπερασματικά, η επιλογή της κατάλληλης τιμής του h αποτελεί ένα trade off μεταξύ της ακρίβειας και της γενίκευσης.

***Ερώτημα Δ***

***Να απεικονίστε τα δεδομένα και τις περιοχές απόφασης για ταξινόμηση σύμφωνα με τον κανόνα k-NN για k=3, k=8 και k=30. Πως επηρεάζονται οι περιοχές απόφασης από την παράμετρο k; Σχολιάστε.***

Ακολουθώ την ίδια μεθοδολογία με το προηγούμενο ερώτημα, μόνο που αυτήν την φορά καλώ την συνάρτηση `knn_classifier` έναντι της `parzen_classifier`. Τα αποτελέσματα φαίνονται στην εικόνα 10.




*Εικόνα 10: Συγκεντρωτικά αποτελέσματα για μέθοδο kNN με διαφορετικά k*

Όπως φαίνεται και από την εικόνα 10 ο παράγοντας k παίζει κρίσιμο ρόλο στην εκτίμηση της σ.π.π. για τα δεδομένα. Επομένως, επηρεάζει άμεσα τις περιοχές απόφασης οι οποίες με την σειρά τους καθορίζουν το πώς ταξινομούνται τα στοιχεία του test set. Όπως προαναφέρθηκε, για μικρό k η εκτίμηση της σ.π.π. γίνεται πιο ευαίσθητη στα τοπικά χαρακτηριστικά, με αποτέλεσμα οι περιοχές απόφασης να γίνονται ακανόνιστες, αντικατοπτρίζοντας την υψηλή διακύμανση στην εκτίμηση της σ.π.π. Αντίθετα, για μεγάλες τιμές του k η εκτίμηση γίνεται ανθεκτική στις τοπικές διακυμάνσεις ομαλοποιώντας την εκτίμηση της σ.π.π. Οι περιοχές απόφασης γίνονται πιο ομαλές αφού παρουσιάζεται μικρότερη ευαισθησία στον θόρυβο ή σε outliers. Τονίζεται πως για ακραίες τιμές του k εγκυμονεί ο κίνδυνος για over- ή underfitting αντίστοιχα. Συμπερασματικά, η επιλογή της κατάλληλης τιμής του k αποτελεί ένα trade off μεταξύ της ακρίβειας και της γενίκευσης.

***Ερώτημα Ε***

***Ποια τα κυριότερα πλεονεκτήματα και μειονεκτήματα των δύο τεχνικών τόσο στη μεταξύ τους σύγκριση, όσο και σε σύγκριση με γεωμετρικές τεχνικές ταξινόμησης (π.χ. linear discriminants, SVMs κλπ.);***

Οι μέθοδοι παραθύρων Parzen και kNN είναι δύο δημοφιλείς μη παραμετρικές τεχνικές ταξινόμησης, με αποτέλεσμα να αναπαριστούν πολύπλοκες κατανομές. Μη υποθέτοντας κάποια συγκεκριμένη μορφή κατανομής, καθίστανται κατάλληλες για προβλήματα όπου η κατανομή των δεδομένων είναι άγνωστη ή πολύπλοκη. Ωστόσο, για ακριβείς εκτιμήσεις χρειάζονται μεγάλο όγκο δεδομένων, ειδικά σε μεγάλες διαστάσεις (curse of dimensionality). Επίσης, η τιμή των παραμέτρων h και k αντίστοιχα είναι κρίσιμη, όπως φάνηκε από τα προηγούμενα ερωτήματα, και μπορεί να αποβεί ιδιαίτερα δύσκολη. Η μέθοδος παραθύρων Parzen εκτιμά την σ.π.π. ανάλογα με το πόσα στοιχεία βρίσκονται εντός ενός προκαθορισμένου παραθύρου γύρω από κάθε σημείο του χώρου (σταθερός όγκος). Η ταξινόμηση με την μέθοδος kNN βασίζεται στην πλειοψηφία των κλάσεων των k πλησιέστερων γειτόνων (σταθερό k). Η επιλογή μεταξύ των δύο μεθόδων εξαρτάται από την πρακτικότητα και την ευκολία εφαρμογής στο συγκεκριμένο πρόβλημα και το είδος των δεδομένων. Για περιπτώσεις όπου η πυκνότητα των δεδομένων είναι σημαντική, η μέθοδος Parzen μπορεί να παρέχει πιο ακριβείς εκτιμήσεις. Ως προς την ταχύτητα, η μέθοδος kNN είναι αργή για μεγάλα datasets, αλλά και η μέθοδος παραθύρων Parzen χρειάζεται αρκετούς υπολογισμούς, ιδίως για μικρά h. Η μέθοδος kNN μπορεί να αντιμετωπίζει καλύτερα τα outliers και τον θόρυβο, ενώ η μέθοδος Parzen είναι πιο ευαίσθητη σε αυτά λόγω της επίδρασης όλων των σημείων στην εκτίμηση της σ.π.π.

Στον αντίποδα, οι γεωμετρικές τεχνικές (Linear Discriminants, SVM) έχουν ισχυρή θεωρητική βάση και παρέχουν σαφείς γεωμετρικές ερμηνείες για τα σύνορα απόφασης. Αυτές οι μέθοδοι είναι κατάλληλες για προβλήματα με μεγάλο όγκο δεδομένων και υψηλές διαστάσεις. Επίσης, παρουσιάζουν ιδιαίτερα καλή εκτίμηση σε νέα δεδομένα για καλή παραμετροποίηση, η οποία όμως μπορεί να αποβεί δύσκολη. Στην μέθοδο Linear Discriminants η υπόθεση για γραμμικό διαχωρισμό μπορεί να μην είναι πάντα σωστή, ενώ η SVM μπορεί να είναι ευαίσθητη σε νέα δεδομένα και απαιτείται προσεκτική επιλογή της παραμέτρου C.

## ΑΣΚΗΣΗ 2

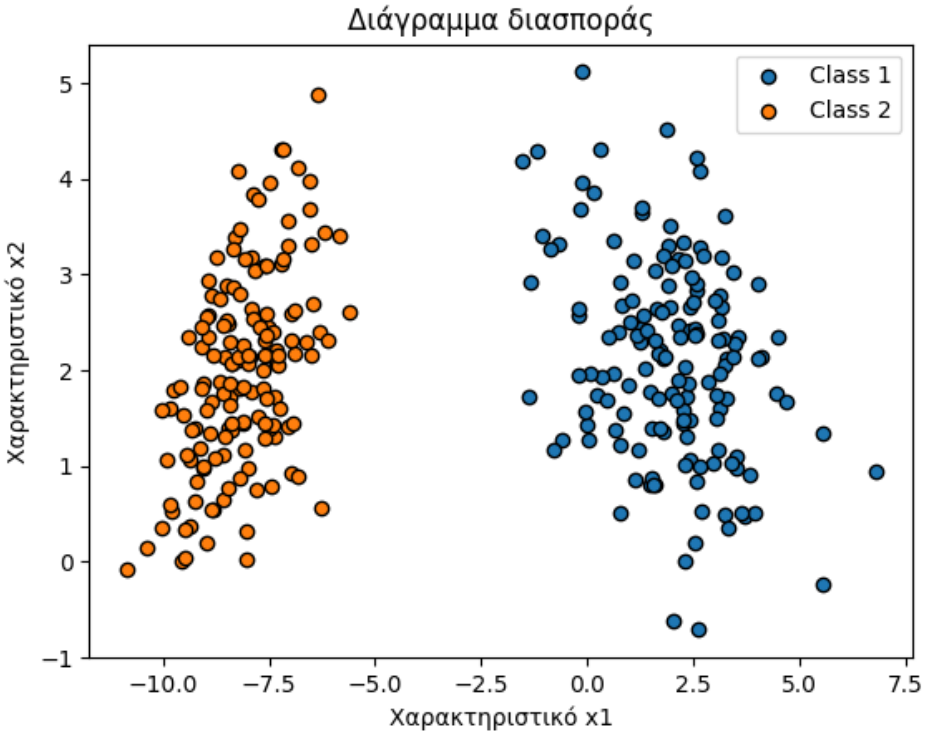
***Έστω δυο κατηγορίες ω1 και ω2 , των οποίων τα δείγματα σε 2-διαστάσεις ακολουθούν κατανομές που περιγράφονται από τις ακόλουθες πυκνότητες πιθανότητας*** $p(x|\omega_1) = N(\mu_1, \Sigma_1)$ ***&*** $p(x|\omega_2) = N(\mu_2, \Sigma_2)$ ***όπου:***

$$\mu_1 = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \qquad \Sigma_1 = \begin{pmatrix} 2 & -0.5 \\ -0.5 & 1 \end{pmatrix}$$

$$\mu_2 = \begin{pmatrix} -8 \\ 2 \end{pmatrix} \qquad \Sigma_2 = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

***Γράψτε κώδικα ώστε να παράξετε 150 τυχαία δείγματα από καθεμία από αυτές τις κατηγορίες και απεικονίστε τα δείγματα κατάλληλα ώστε να διακρίνονται οι κλάσεις.***

Αρχικά, εισάγω τα δεδομένα μου και δημιουργώ 150 τυχαία δείγματα που ακολουθούν τις δοθείσες κατανομές. Η απεικόνιση αυτών φαίνεται στην εικόνα 11.




*Εικόνα 11: Απεικόνιση δειγμάτων δεύτερης άσκησης.*

***Ερώτημα Α***

***Υλοποιείστε τον αλγόριθμό του Batch Perceptron και με αυτόν υπολογίστε έναν γραμμικό ταξινομητή για τις κλάσεις αυτές. Απεικονίστε την επιφάνεια απόφασης που προέκυψε επάνω στο προηγούμενο γράφημα.***

Ο αλγόριθμος Batch Perceptron είναι μια παραλλαγή του αλγορίθμου του Perceptron. Ο Perceptron είναι ένας από τους πρώτους αλγορίθμους μηχανικής μάθησης και χρησιμοποιείται για επιβλεπόμενη ταξινόμηση (supervised classification), δηλαδή για να αποφασίσει αν ένα νέο δείγμα ανήκει σε μια κλάση ή σε μια άλλη στην απλή περίπτωση των δύο κλάσεων. Ο αλγόριθμος αυτός είναι δυαδικός και γραμικός, δηλαδή διαχωρίζει τα δεδομένα σε δύο κλάσεις με την χρήση μιας γραμμικής εξίσωσης. Σε περίπτωση περισσότερων κλάσεων υπάρχουν διάφορες προσεγγίσεις όπως η one-vs-one ή one-vs-all. Για τον διαχωρισμό ο αλγόριθμος βρίσκει ένα γραμμικό επίπεδο το οποίο διαχωρίζει τα δείγματα σε δύο κλάσεις.

Ο όρος "Batch" αναφέρεται στη μέθοδο ενημέρωσης των βαρών κατά την εκπαίδευση του αλγορίθμου. Αν ένα δείγμα ταξινομηθεί σωστά, τότε τα βάρη δεν αλλάζουν. Αν ταξινομηθεί λανθασμένα, τα βάρη προσαρμόζονται ως εξής:

$$w_{new} = w_{old} + \eta(y - \hat{y})x$$
$$b_{new} = b_{old} + \eta(y - \hat{y})$$

| | | |
|---|---|---|
| όπου | $\eta$ | Ρυθμός μάθησης |
| | $y$ | Πραγματική ετικέτα |
| | $\hat{y}$ | Προβλεπόμενη ετικέτα |

| | |
|---|---|
| $w_{new}$ | Νέα τιμή βαρών |
| $w_{old}$ | Παλιά τιμή βαρών |
| $b_{old}$ | Παλιά τιμή κατωφλίου |
| $b_{new}$ | Νέα τιμή |

Η ενημέρωση των βαρών γίνεται μετά από την εξέταση όλων των δειγμάτων στο σύνολο δεδομένων, αντί για κάθε επιμέρους δείγμα. Η κάθε επανάληψη ονομάζεται εποχή. Ο αλγόριθμος Batch Perceptron θα συγκλίνει σε έναν βέλτιστο διαχωρισμό αν τα δεδομένα είναι γραμμικά διαχωρίσιμα. Διαφορετικά, ο αλγόριθμος μπορεί να μην συγκλίνει, και θα χρειαστούν πρόσθετες τεχνικές για να χειριστεί τις μη γραμμικές αυτές περιπτώσεις.

Όπως φαίνεται από τις παραπάνω σχέσης οι αρχικές τιμές των βαρών παίζουν σημαντικό ρόλο στην τελική απόδοση, καθώς καθορίζουν το σημείο εκκίνησης της εκμάθησης. Ανάλογα με αυτές ο αλγόριθμος προσαρμόζει τα βάρη από διαφορετική «θέση» στον χώρο των λύσεων. Καλές αρχικές τιμές μπορεί να επιτύχουν την σύγκλιση του αλγορίθμου στην βέλτιστη λύση, ενώ κακές αρχικές τιμές πρόκειται να καθυστερήσουν ή ακόμα και να εμποδίσουν την σύγκλιση με απόρροια την κακή απόδοση. Επίσης, σε περίπλοκα προβλήματα ενδέχεται ο αλγόριθμος να εγκλωβιστεί σε τοπικά ελάχιστα της συνάρτησης σφάλματος. Συνήθως οι αρχικές τιμές είναι κοντά στο μηδέν ώστε η εκμάθηση να ξεκινήσει χωρίς ισχυρές προκαταλήψεις και διαφορετικές μεταξύ τους, καθώς αυτό μπορεί να περιορίσει την αποτελεσματικότητα. Στην υλοποίησή μου επέλεξα τυχαίες τιμές από το -0,5 έως το 0,5.

Το κατώφλι βοηθάει σημαντικά στην λήψη της απόφασης καθώς χρησιμεύει ως μία τιμή αναφοράς για την ταξινόμηση των δειγμάτων στις δύο κλάσεις. Εάν η συνολική ενεργοποίηση (προϊόν βαρών και εισόδων) υπερβαίνει το κατώφλι, το δείγμα ταξινομείται στην μία κλάση, ενώ αν είναι κάτω από αυτό, ταξινομείται στην άλλη. Επομένως, με το κατώφλι ρυθμίζεται η ευαισθησία του μοντέλου: υψηλή τιμή καθιστά το μοντέλο πιο συντηρητικό στην ταξινόμηση των δειγμάτων σε μία συγκεκριμένη κλάση. Μία ουδέτερη τιμή που δεν μεροληπτεί υπέρ της μίας ή της άλλης κλάσης είναι το 0 και για αυτό υιοθετήθηκε στην υλοποίησή μου.

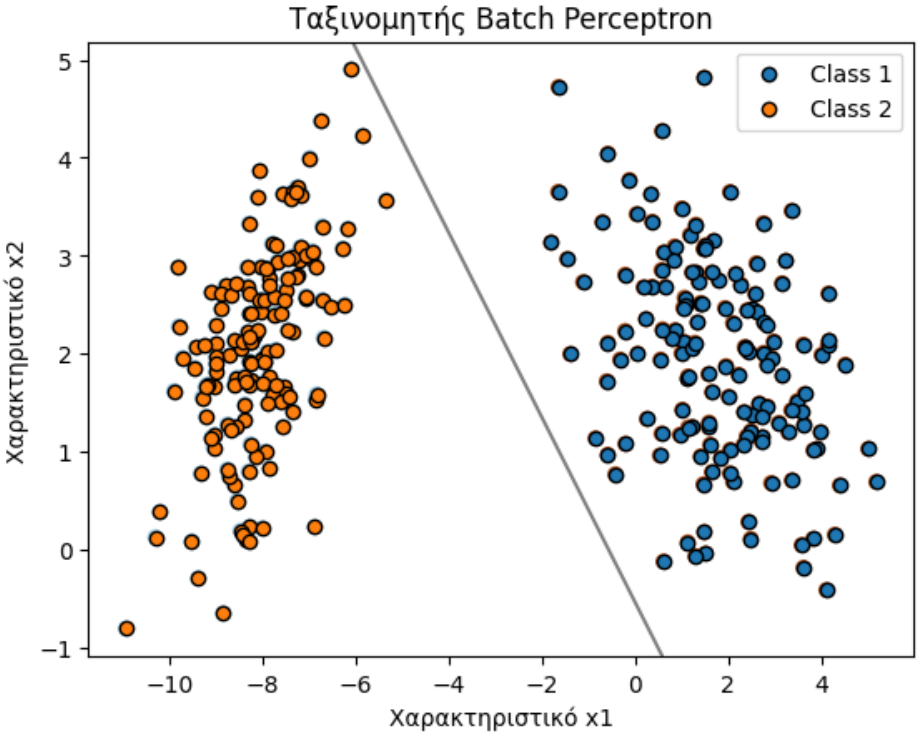
Ο ρυθμός εκμάθησης, όπως σε όλους τους αλγορίθμους της μηχανικής μάθησης, καθορίζει τόσο την ταχύτητα όσο και τον τρόπο που μαθαίνει το μοντέλο. Με άλλα λόγια προσδιορίζει το μέγεθος των βημάτων κατά την ενημέρωση των βαρών. Ένας υψηλός συντελεστής μπορεί να οδηγήσει σε μεγάλες ενημερώσεις βάρους κάνοντας το μοντέλο να "πηδά" πέρα από βέλτιστες λύσεις, ενώ ένας χαμηλός συντελεστής προκαλεί μικρότερες αλλαγές καθυστερώντας όμως την σύγκλιση. Επιπλέον, ένας κατάλληλα επιλεγμένος συντελεστής εκμάθησης μπορεί να βοηθήσει το μοντέλο να αποφύγει τον εγκλωβισμό σε τοπικά ελάχιστα της συνάρτησης σφάλματος. Συνήθεις τιμές για τον ρυθμό εκμάθησης κυμαίνονται από πολύ μικρές (π.χ., 0.01 ή 0.001) έως μεσαίες (π.χ., 0.1). Η επιλογή της κατάλληλης τιμής εξαρτάται από το συγκεκριμένο πρόβλημα και το σετ δεδομένων. Στο παρόν πρόβλημα επιλέχτηκε αυθαίρετα η τιμή 0,01.

Τέλος, οι εποχές (epochs) αναφέρονται στον αριθμό των πλήρων διαβάσεων των δεδομένων που εκτελεί ο αλγόριθμος κατά τη διάρκεια της μάθησης. Κάθε εποχή παρέχει στον αλγόριθμο μια ευκαιρία να "μάθει" και να βελτιώσει τα βάρη. Με την αύξηση του αριθμού των εποχών, συχνά βελτιώνεται η απόδοση του μοντέλου έως ένα σημείο, καθώς το μοντέλο προσαρμόζεται καλύτερα στα δεδομένα. Παρ' όλα αυτά, υπάρχει ο κίνδυνος υπερεκπαίδευσης (overfitting) αν το μοντέλο εκπαιδευτεί για πάρα πολλές εποχές, μαθαίνοντας να προβλέπει τέλεια τα δεδομένα εκπαίδευσης αλλά αποτυγχάνοντας να γενικεύσει σε νέα δεδομένα. Μία τεχνική ώστε να αποφύγω το overfitting είναι η early stopping, δηλαδή η διακοπή της εκπαίδευσης όταν δεν παρατηρείται περαιτέρω βελτίωση της απόδοσης. Ωστόσο, λόγω της απλότητας του προβλήματος προτίμησα να μην την υλοποιήσω. Η τιμή των εποχών έπειτα από αρκετές δοκιμές ορίστηκε 100.

Σημειώνω ότι η τιμή του ρυθμού μάθησης και των εποχών μπορεί να αλλάξουν καθώς αποτελούν ορίσματα της συνάρτησης που υλοποιεί τον αλγόριθμο του Batch Perceptron

Οι ετικέτες των δεδομένων χρησιμοποιούνται συχνά με τις τιμές 1 και -1 για να απλοποιήσουν τους υπολογισμούς και να καταστήσουν πιο άμεση τη διαδικασία ταξινόμησης. Η χρήση αυτών των τιμών διευκολύνει τον υπολογισμό του σφάλματος. Όταν το γινόμενο της πρόβλεψης με την ετικέτα είναι θετικό, αυτό σημαίνει ότι η ταξινόμηση είναι σωστή. Αντίθετα, αν είναι αρνητικό, σημαίνει ότι η ταξινόμηση είναι λανθασμένη. Αυτό καθιστά τον έλεγχο της ορθότητας ταξινόμησης πολύ απλό. Ως προς την άμεση ενημέρωση των βαρών και του κατωφλίου, αυτή γίνεται προς την κατεύθυνσης του δείγματος στην περίπτωση της σωστής ταξινόμησης ή προς την αντίθετη στην περίπτωση της λανθασμένης ταξινόμησης. Ακόμη, οι τιμές 1 και -1 είναι ιδανικές για την αναπαράσταση δύο αντίθετων κλάσεων σε ένα γραμμικό ταξινομητή

Οι περιοχές απόφασης μίας επανάληψης με βάση αυτόν τον ταξινομητή Batch Perceptron φαίνονται στην εικόνα 12.




*Εικόνα 12: Δεδομένα δεύτερης άσκησης και περιοχή απόφασης με Batch Perceptron.*

***Ερώτημα Β***

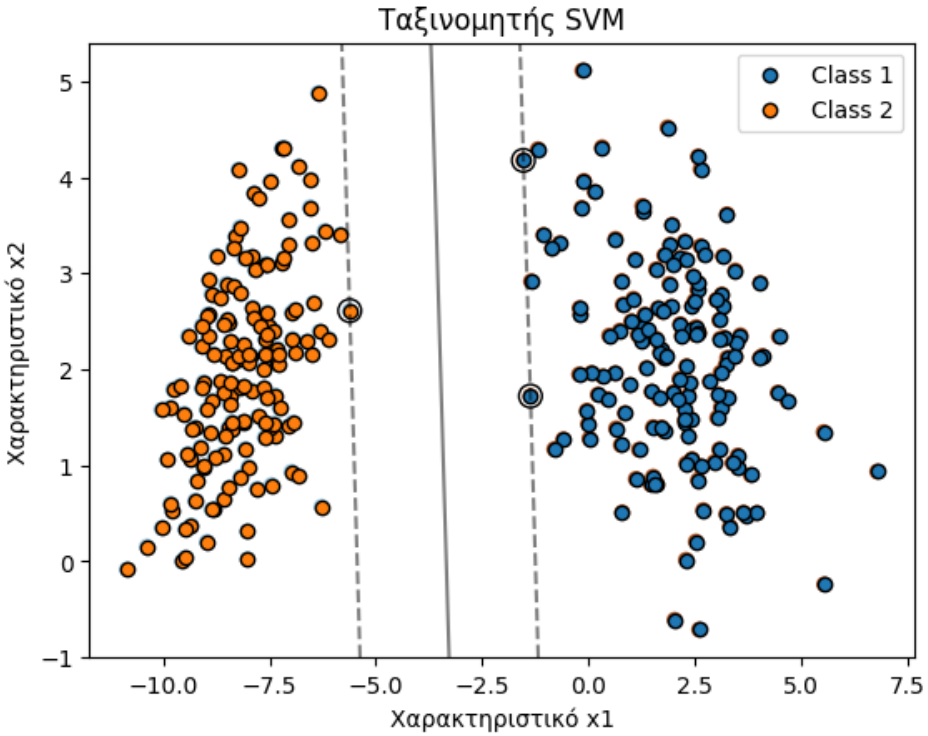
***Χρησιμοποιείστε γραμμικό SVM (από κατάλληλη βιβλιοθήκη της επιλογής σας) για να υπολογίσετε έναν νέο γραμμικό ταξινομητή για τα ίδια δεδομένα. Απεικονίστε τα δεδομένα, τα support vectors και το επίπεδο απόφασης σε νέο διάγραμμα***

Ο ταξινομητής SVM (Support Vector Machine) είναι ένας δημοφιλής αλγόριθμος μηχανικής μάθησης, ο οποίος χρησιμοποιείται για ταξινόμηση και παλινδρόμηση. Η κύρια ιδέα του γραμμικού SVM είναι να βρει το βέλτιστο επίπεδο που διαχωρίζει τα δεδομένα σε δύο κατηγορίες. Συνεπώς, είναι ιδιαίτερα αποτελεσματικός σε προβλήματα δυαδικής ταξινόμησης.

Ο πυρήνας (kernel) σε έναν ταξινομητή SVM καθορίζει τον τρόπο μετασχηματισμού των δεδομένων ώστε να γίνει δυνατός ο διαχωρισμός των δεδομένων σε κλάσεις. Υπάρχουν διάφοροι τύποι πυρήνων που μπορούν να χρησιμοποιηθούν ανάλογα με τον τύπο των δεδομένων. Στην παρούσα άσκηση επιλέγεται ο γραμμικός πυρήνας (`'linear'`) καθώς τα δεδομένα είναι γραμμικά διαχωρίσιμα, δηλαδή μπορούν να διαχωριστούν με μια ευθεία γραμμή (σε δύο διαστάσεις). Επιπλέον, ο γραμμικός πυρήνας απαιτεί λιγότερους υπολογιστικούς πόρους σε σύγκριση με τους μη γραμμικούς. Η χρήση πιο περίπλοκων πυρήνων σε γραμμικά διαχωρίσιμα δεδομένα άλλωστε, ενδέχεται να οδηγήσει σε υπερπροσαρμογή.

Ο SVM με γραμμικό πυρήνα στοχεύει στον εντοπισμό ενός γραμμικού υπερεπιπέδου που διαχωρίζει τα δεδομένα σε δύο κλάσεις. Στον διδιάστατο χώρο, αυτό το υπερεπίπεδο είναι μια γραμμή. Τα δεδομένα που βρίσκονται πλησιέστερα στο διαχωριστικό επίπεδο και καθορίζουν την θέση και τον προσανατολισμό του ονομάζονται διανύσματα υποστήριξης (support vectors). Αυτά τα σημεία είναι κρίσιμα στη διαμόρφωση του ταξινομητή, καθώς η τοποθεσία τους βοηθά στον προσδιορισμό του βέλτιστου υπερεπιπέδου που χωρίζει τις κατηγορίες. Ο SVM επιδιώκει να μεγιστοποιήσει το περιθώριο μεταξύ των δύο κλάσεων, δηλαδή την απόσταση μεταξύ του διαχωριστικού υπερεπιπέδου και των support vectors. Η εύρεση αυτού μπορεί να μορφοποιηθεί ως ένα πρόβλημα βελτιστοποίησης, όπου μεγιστοποιούμε το περιθώριο, διατηρώντας παράλληλα την σωστή ταξινόμηση των δεδομένων.

Στην εικόνα 13 φαίνονται οι περιοχές απόφασης με τον γραμμικό ταξινομητή SVM καθώς και τα support vectors.




*Εικόνα 13: Δεδομένα δεύτερης άσκησης και περιοχή απόφασης με SVM*

***Ερώτημα Γ***
***Σχολιάστε το αποτέλεσμα των δύο τεχνικών. Ποιες οι διαφορές και που οφείλονται?***

Για την ταξινόμηση γραμμικά διαχωρίσιμων δεδομένων, τόσο ο Batch Perceptron όσο και ο SVM επιτυγχάνουν καλά αποτελέσματα. Παρ' όλα αυτά, παρατηρώντας και τις εικόνες 12 και 13, υπάρχουν ορισμένες κυρίαρχες διαφορές μεταξύ των δύο προσεγγίσεων που αξίζει να σχολιαστούν.

| | **Batch Perceptron** | **SVM** |
|---|---|---|
| **Συνάρτηση Απόφασης** | Διαχωρίζει τις κλάσεις με ένα υπερεπίπεδο, τροποποιώντας τα βάρη με βάση τα σφάλματα ταξινόμησης. | Το υπερεπίπεδο που επιλέγει δεν διαχωρίζει απλά τα δείγματα, αλλά επιδιώκει να μεγιστοποιήσει το περιθώριο μεταξύ των κλάσεων. |
| **Αντοχή σε Θόρυβο** | Επηρεάζεται πολύ από θορυβώδη δεδομένα ή ακραίες τιμές (outliers). | Ανθεκτικός σε θόρυβο λόγω της μεγιστοποίησης του περιθωρίου και της ανάλυσης βασισμένης στα δείγματα υποστήριξης (support vectors). |
| **Σύγκλιση** | Η σύγκλιση δεν είναι εγγυημένη εάν τα δεδομένα δεν είναι γραμμικά διαχωρίσιμα. Η ταχύτητα σύγκλισης μπορεί να είναι αργή για μεγάλα σετ δεδομένων. | Έχει θεωρητικές εγγυήσεις σύγκλισης και συνήθως συγκλίνει πιο γρήγορα από τον perceptron για γραμμικά διαχωρίσιμα δεδομένα. |
| **Υπολογιστική Πολυπλοκότητα** | Μπορεί να είναι πιο απλός υπολογιστικά σε σχέση με τον SVM, αλλά αυτό μπορεί να μην είναι προτέρημα εάν η γραμμική σύγκλιση είναι αργή. | Η εκπαίδευση μπορεί να είναι πιο απαιτητική υπολογιστικά, αλλά οι σύγχρονες βελτιστοποιημένες εκδοχές του τον κάνουν πολύ αποτελεσματικό ακόμη και για μεγάλα σετ δεδομένων. |
| **Ευελιξία** | Είναι πιο περιορισμένος στην εφαρμογή του, καθώς κατά βάση ενδείκνυται για γραμμικά διαχωρίσιμα προβλήματα. | Προσφέρει μεγαλύτερη ευελιξία μέσω της επιλογής διάφορων πυρήνων για μη-γραμμική ταξινόμηση. |

Συνοψίζοντας, ενώ και οι δύο μέθοδοι είναι κατάλληλες για γραμμικά διαχωρίσιμα δεδομένα, ο SVM συνήθως προτιμάται λόγω της γενικότερης αποδοτικότητας, της αντοχής στον θόρυβο και της ευελιξίας του στην επιλογή πυρήνων για μη-γραμμικές εφαρμογές.

## ΑΣΚΗΣΗ 3

***Κατεβάστε το Wine Dataset. Τα δεδομένα αυτά αποτελούν τα αποτελέσματα μιας χημικής ανάλυσης κρασιών από τρεις διαφορετικές καλλιεργητικές ποικιλίες {c1,c2,c3}, και περιλαμβάνουν τιμές για 13 χημικά συστατικά που μετρήθηκαν σε κάθε κρασί. Η πρώτη στήλη του αρχείου των δεδομένων περιλαμβάνει την ετικέτα της ποικιλίας του κάθε κρασιού, και οι επόμενες στήλες τις τιμές των συστατικών που μετρήθηκαν. Στόχος μας είναι να διερευνήσουμε το πρόβλημα της πρόβλεψης της ποικιλίας από τα αποτελέσματα της χημική ανάλυσης.***

Ακολουθώ την ίδια διαδικασία για την φόρτωση και ανάγνωση των δεδομένων με την άσκηση 1.

***Ερώτημα Α***

***Θεωρείστε το υποσύνολο των δεδομένων που περιέχει τιμές μόνο για τα 5 πρώτα συστατικά και για τα κρασιά από τις ποικιλίες c2 και c3.***

Από τα αρχικά μου δεδομένα κρατάω εκείνα που ανήκουν στην κλάση c2 και c3. Αυτά στην πρώτη στήλη του πίνακα των δεδομένων η οποία προσδιορίζει την κλάση έχουν τιμή 2 ή 3. Έπειτα, κρατάω μονάχα τα 5 πρώτα χαρακτηριστικά αυτών. Δεδομένου ότι η πρώτη στήλη είναι οι ετικέτες, το πρώτο χαρακτηριστικό βρίσκεται στην στήλη 1, σύμφωνα με την αρίθμηση του προγράμματος.

***Να χωρίσετε το παραπάνω σε σύνολο εκπαίδευσης, επικύρωσης και δοκιμής (training, validation & test sets) με αναλογία 50%, 25% και 25% αντίστοιχα, με τυχαία επιλογή δεδομένων και ίδια αναλογία μεταξύ των κλάσεων σε κάθε σύνολο.***

Για τον διαχωρισμό των δεδομένων σε training, validation και test set χρησιμοποιώ την συνάρτηση `train_test_split` της βιβλιοθήκης `sklearn`. Η συνάρτηση αυτή, όπως λέει και το όνομά της, διαχωρίζει τα δεδομένα μονάχα σε train και test set. Για την δημιουργία του validation set καλώ πάλι την ίδια συνάρτηση με σκοπό να χωρίσω το train set της προηγούμενης κλίσης.

Προκείμενου να διατηρηθεί η αναλογία της εκφώνησης ως προς τα δεδομένα κάθε set, το size του validation set θα πρέπει να είναι το 33.33% του 75% του αρχικού train set. Η τιμή αυτή προκύπτει εύκολα με απλή αναλογία. Τα μεγέθη των set φαίνονται στον Πίνακα 1 και συμφωνούν με τα δεδομένα της εκφώνησης.

| | | |
|---|---|---|
| testing | ndarray | (30, 5) |
| training | ndarray | (59, 5) |
| validation | ndarray | (30, 5) |

*Πίνακας 1: Μεγέθη των train, validation και test set.*

***Ερώτημα Β***

***Χρησιμοποιήστε γραμμικό SVM για να εκπαιδεύσετε ταξινομητή που να διαχωρίζει την κλάση c2 από τη c3. Χρησιμοποιήστε το validation set για να ρυθμίσετε την παράμετρο C (box constraint) κατάλληλα.***

Η παράμετρος C στον γραμμικό ταξινομητή SVM είναι κρίσιμης σημασίας, καθώς ρυθμίζει την ισορροπία μεταξύ της επίτευξης ενός μεγάλου περιθωρίου διαχωρισμού και της μείωσης των σφαλμάτων ταξινόμησης.

Για χαμηλή τιμή του C, το μοντέλο τείνει να ευνοεί την μεγιστοποίηση του περιθωρίου, ακόμη και εάν αυτό σημαίνει την αύξηση των σφαλμάτων ταξινόμησης. Αυτή η προσέγγιση μπορεί να οδηγήσει σε μοντέλο γενικότερης μορφή, ικανού δηλαδή να ανταποκρίνεται καλά σε δεδομένα που δεν έχει δει. Αντίθετα, η επιλογή υψηλής τιμής οδηγεί σε μεγαλύτερη έμφαση στη σωστή ταξινόμηση των δεδομένων, μειώνοντας το σφάλμα ταξινόμησης αλλά αυξάνοντας τον κίνδυνο υπερπροσαρμογής (overfitting). Το μοντέλο είναι υπερβολικά εξειδικευμένο στα δεδομένα εκπαίδευσης και μπορεί να μην επιδεικνύει την ίδια απόδοση σε νέα δεδομένα.

Συνεπώς, η επιλογή της τιμής του C αποτελεί ένα σημαντικό βήμα στη διαδικασία δημιουργίας ενός SVM μοντέλου. Η βέλτιστη τιμή του καθορίζεται με προσεκτική ανάλυση και δοκιμές, συνήθως μέσω διαδικασιών όπως η διασταυρωμένη επικύρωση (cross-validation), οι οποίες επιτρέπουν την αξιολόγηση της επίδρασης διαφορετικών τιμών του C στην απόδοση του μοντέλου. Η σωστή ρύθμιση της παραμέτρου αυτής μπορεί να οδηγήσει σε ένα ισορροπημένο μοντέλο, το οποίο επιτυγχάνει έναν αποδοτικό συμβιβασμό μεταξύ της γενίκευσης και της ακρίβειας της ταξινόμησης. Η ρύθμιση της παραμέτρου C με το validation set βοηθάει στην εύρεση της ισορροπίας μεταξύ της ακρίβειας και της γενίκευσης.

Εφόσον η διαδικασία της εύρεσης της καλύτερης τιμής της παραμέτρου C πρόκειται να επαναληφθεί στα επόμενα ερωτήματα, ορίζω μία συνάρτηση με όνομα `find_best_C` που θα υλοποιεί αυτήν την διαδικασία.

Για τον ορισμό του SVM καλώ την αντίστοιχη συνάρτηση της βιβλιοθήκης `sklearn`. Ο πυρήνας του ταξινομητή είναι όρισμα της συνάρτησης προκειμένου να εξυπηρετούνται οι ανάγκες των επόμενων ερωτημάτων. Για να βρω την καλύτερη τιμή για το C χρησιμοποιώ την συνάρτηση `GridSearch` της βιβλιοθήκης. Μετά την αξιολόγηση όλων των πιθανών συνδυασμών παραμέτρων, η `GridSearchCV` επιλέγει τον συνδυασμό που παρέχει την καλύτερη απόδοση σύμφωνα με έναν συγκεκριμένο μετρικό αξιολόγησης (όπως η ακρίβεια). Σύμφωνα με το documentation, οι παράμετροι πρέπει να είναι σε μορφή λεξικού. Προκειμένου η ρύθμιση να γίνει με βάση το validation set, δίνω αυτό σαν όρισμα της συνάρτησης `fit` μαζί με τα αντίστοιχα labels. Μία διαφορετική προσέγγιση είναι η υλοποίηση κώδικα που θα κάνει την αναζήτηση. Ο κώδικας αυτός βρίσκεται σε σχόλια εντός της `find_best_C`. Από τις δύο αυτές προσεγγίσεις αποφάσισα να κρατήσω την πρώτη.

Όπως είχε αναφερθεί και στα πλαίσια του μαθήματος, δεν μας ενδιαφέρει η απόλυτη τιμή του C, αλλά η τάξη μεγέθους του. Επομένως, οι αρχικές τιμές στις οποίες γίνεται η αναζήτηση της βέλτιστης τιμής για το C είναι 0.01, 0.1, 1, 10, 100, 1000.

***Για την καλύτερη τιμή εφαρμόστε τον ταξινομητή που εκπαιδεύσατε στο test set. Τι σφάλμα ταξινόμησης πετύχατε?***

Έχοντας βρει την καλύτερη τιμή για την μεταβλητή C, ορίζω νέο γραμμικό ταξινομητή SVM με αυτή την τιμή και τον εκπαιδεύω στα δεδομένα του training set. Προκειμένου να αξιολογήσω το μοντέλο μου, το εφαρμόζω σε νέα δεδομένα που δεν έχει δει (test set) και υπολογίζω το σφάλμα που παρουσιάζει. Για μία εκτέλεση το λάθος ταξινόμησης για την καλύτερη τιμή του C που βρέθηκε είναι:

```
Σφάλμα Ταξινόμησης: 0.30000000000000004 με C = 0.1
```

***Ερώτημα Γ***

***Επαναλάβετε το προηγούμενο για 5 νέους τυχαιοποιημένους διαμερισμούς των δεδομένων, και υπολογίστε τη μέση τιμή και την τυπική απόκλιση του σφάλματος ταξινόμησης στο test set.***

Η διαδικασία αυτή επαναλαμβάνεται για διαφορετικούς τύπους πυρήνα στο επόμενο ερώτημα, οπότε υλοποίησα μία συνάρτηση που να την εκτελεί. Η εξήγηση της συνάρτησης γίνεται παρακάτω, εδώ παρουσιάζονται μονάχα τα αποτελέσματα του γραμμικού ταξινομητή.

```
 KERNEL     ΜΕΣΗ ΤΙΜΗ ΣΦΑΛΜΑΤΟΣ    ΑΠΟΚΛΙΣΗ
-------------------------------------------------
 linear          0.14667             0.05812
```

***Ερώτημα Δ***

***Επαναλάβετε το Γ για μη-γραμμικό SVM δοκιμάζοντας διάφορες συναρτήσεις πυρήνα (RBF, polynomial κλπ). Τι σφάλμα πετύχατε? Ποιος είναι ο καλύτερος ταξινομητής για το πρόβλημα? Σχολιάστε.***

Εφόσον θα πρέπει να επαναλάβω μία διαδικασία αλλάζοντας κάποια μεγέθη δημιουργώ μία συνάρτηση με τα αντίστοιχα ορίσματα. Η συνάρτηση αυτή αποτελείται από τον κώδικα των προηγούμενων ερωτημάτων με την διαφορά ότι η διαδικασία επαναλαμβάνεται για κάθε στοιχείο της λίστας των πυρήνων (`kernels`).

Δημιουργώ ένα λεξικό με κλειδί τον τύπο κάθε πυρήνα και τιμές σε μορφή λίστας το σφάλμα κάθε επανάληψης. Συνολικά έχω 5 επαναλήψεις. Προκειμένου τα συμπεράσματά μου να έχουν βάση θα πρέπει για κάθε τυχαίο διαχωρισμό των δεδομένων να εφαρμόζονται όλοι οι πυρήνες του ταξινομητή και να υπολογίζεται ξεχωριστά η μέση τιμή σφάλματος και η απόκλιση αφού πρώτα βρω την καλύτερη τιμή της παραμέτρου C. Οι τιμές αυτές αποθηκεύονται σε ξεχωριστό λεξικό και εκτυπώνονται στο τέλος. Επομένως, η επανάληψη που διατρέχει τους πίνακες είναι εμφωλευμένη εντός εκείνης που επαναλαμβάνει ολόκληρη την διαδικασία 5 φορές.

Για τον υπολογισμό της μέσης τιμής και της απόκλισης, μετατρέπω την λίστα των τιμών για κάθε κελί σε πίνακα numpy και χρησιμοποιώ τις αντίστοιχες συναρτήσεις της βιβλιοθήκης. Τέλος, εκτυπώνω τα αποτελέσματα. Τα αποτελέσματα μίας κλήσης της συνάρτησης για όλους τους πυρήνες είναι:

| KERNEL | ΜΕΣΗ ΤΙΜΗ ΣΦΑΛΜΑΤΟΣ | ΑΠΟΚΛΙΣΗ |
|---|---|---|
| linear | 0.17333 | 0.05735 |
| poly | 0.10667 | 0.05333 |
| rbf | 0.24000 | 0.03266 |
| sigmoid | 0.58000 | 0.09798 |

Η απόφαση του καλύτερου ταξινομητή προκύπτει από τους δύο προαναφερόμενους δείκτες: την μέση τιμή σφάλματος και την απόκλιση. Η μέση τιμή δείχνει το πόσο καλά ταξινομούνται τα δεδομένα, άρα όσο χαμηλότερη τόσο το καλύτερο. Η απόκλιση προσδιορίζει την σταθερότητα του ταξινομητή στις διάφορες εκτελέσεις. Μικρή απόκλιση σημαίνει ότι ο ταξινομητής έχει συνεπή αποτελέσματα ανεξάρτητα από το δείγμα των δεδομένων που χρησιμοποιεί.

Με βάση τα παραπάνω αποτελέσματα, ο πολυωνυμικός ταξινομητής έχει την χαμηλότερη μέση τιμή σφάλματος, δηλαδή κατά μέσο όρο ταξινομεί καλύτερα τα δεδομένα, και μικρή απόκλιση, δηλαδή παρουσιάζει συνεπή αποτελέσματα. Οι υπόλοιποι πυρήνες έχουν υψηλότερη μέση τιμή σφάλματος, δηλαδή παρουσιάζουν περισσότερα λάθη στην ταξινόμηση. Συνεπώς, δεδομένου της ακρίβειας και της συνέπειας ο πολυωνυμικός ταξινομητής είναι καλύτερος για τα συγκεκριμένα δεδομένα. Από αυτό συμπεραίνω ότι η δομή των δεδομένων είναι πολύπλοκη.

***Ερώτημα Ε***

***Χρησιμοποιήστε γραμμικό SVM για να εκπαιδεύσετε ταξινομητές για το πλήρες πρόβλημα των 3 κλάσεων, με την προσέγγιση ένας-εναντίον-ενός (one-vs-one) και καταμέτρηση ψήφων. Μπορείτε να αξιοποιήσετε τις σχετικές αυτοματοποιημένες λειτουργίες που έχουν κάποιες βιβλιοθήκες SVM.***

Η προσέγγιση one-vs-one είναι μία τεχνική που χρησιμοποιείται σε προβλήματα ταξινόμησης πολλών κλάσεων. Για $N > 2$ κλάσεις δημιουργούνται $\frac{N*(N-1)}{2}$ ταξινομητές. Έκαστος εκπαιδεύεται σε δεδομένα δύο κλάσεων, αγνοώντας τα δεδομένα από τις υπόλοιπες. Όταν πρόκειται να γίνει πρόβλεψη για νέο δείγμα, αυτό παρουσιάζεται σε όλους τους ταξινομητές. Κάθε ταξινομητής ψηφίζει για μία κλάση. Η τελική πρόβλεψη είναι η κλάση με τις περισσότερες ψήφους.

Η υλοποίηση της τεχνικής αυτής γίνεται με την βοήθεια της έτοιμης συνάρτησης `OneVsOneClassifier` από την βιβλιοθήκη `sklearn`.

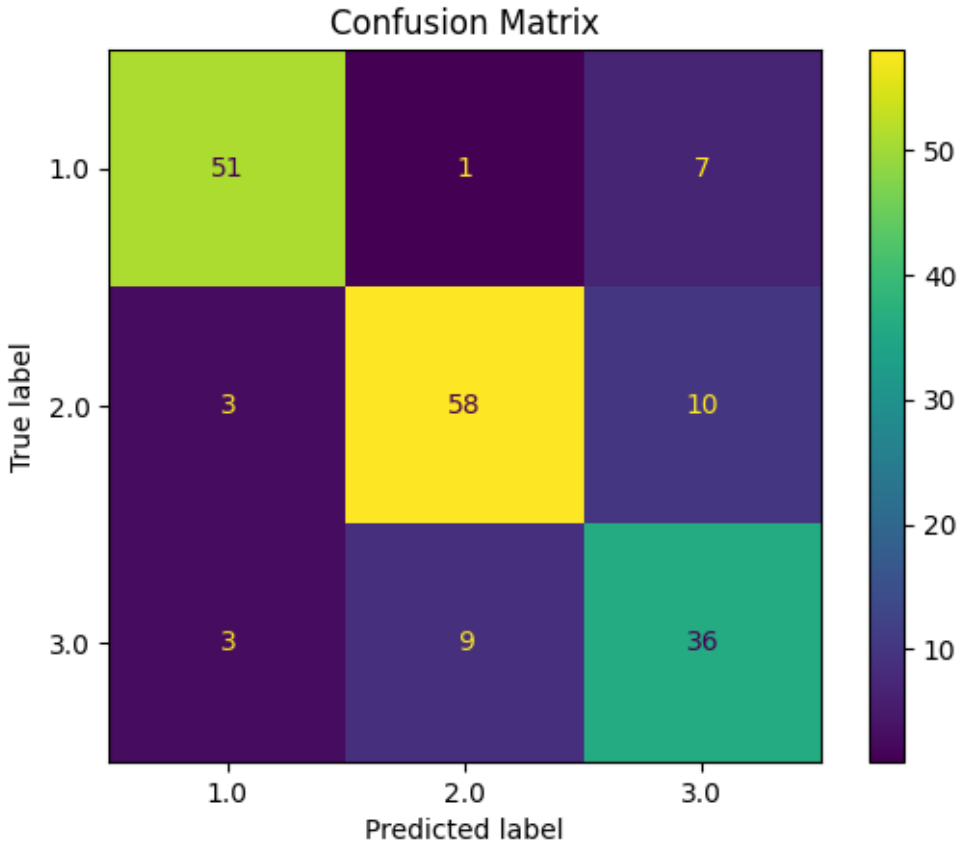
***Θέτοντας το C=1 και ακολουθώντας πρωτόκολλο 5-fold cross validation, να υπολογίστε τη μέση τιμή του σφάλματος ταξινόμησης χρησιμοποιώντας α) τα 5 πρώτα χαρακτηριστικά όπως και παραπάνω, και β) όλα τα διαθέσιμα χαρακτηριστικά. Σχολιάστε τα αποτελέσματα.***
***Υπολογίστε για κάθε περίπτωση τον πίνακα σύγχυσης (confusion matrix) της ταξινόμησης και σχολιάστε ποιες κλάσεις ομοιάζουν περισσότερο.***

Το πρωτόκολλο 5-fold cross validation είναι μία μέθοδο αξιολόγησης της απόδοσης ενός μοντέλου μηχανικής μάθησης με βάση το πόσο καλά γενικεύει σε νέα δεδομένα. Το σύνολο των δεδομένων διαιρείται τυχαία σε 5 ίσα σύνολα (folds). Κάθε υποσύνολο πρέπει να είναι όσο πιο αντιπροσωπευτικό του αρχικού γίνεται, δηλαδή να περιέχει μία καλή αναλογία δειγμάτων από κάθε κλάση. Σε κάθε γύρο, ένα από τα πέντε υποσύνολα λειτουργεί ως test set και τα υπόλοιπα ως training. Το μοντέλο εκπαιδεύεται με τα δεδομένα εκπαίδευσης και αξιολογείται στο test set. Αυτή η διαδικασία επαναλαμβάνεται πέντε φορές, με κάθε υποσύνολο να χρησιμοποιείται ακριβώς μία φορά ως test set. Μετά την ολοκλήρωση των πέντε γύρων, έχουμε πέντε μετρήσεις απόδοσης. Η τελική απόδοση του μοντέλου εκτιμάται συνήθως ως ο μέσος όρος αυτών των μετρήσεων.

Το γεγονός ότι όλα τα δεδομένα χρησιμοποιείται τόσο για εκπαίδευση όσο και για αξιολόγηση αυξάνει την αξιοπιστία. Αντί για την διαίρεση των δεδομένων σε ξεχωριστά σετ training και test, η τεχνική cross validation επιτρέπει την πιο αποτελεσματική χρήση του συνόλου των διαθέσιμων δεδομένων. Ωστόσο, η διαδικασία αυτή μπορεί να απαιτεί σημαντικά περισσότερο χρόνο σε σύγκριση με μια απλή διαίρεση σε training και test set. Επιπλέον, σε περιπτώσεις όπου τα δεδομένα δεν είναι ισορροπημένα ή όταν οι διαφορές μεταξύ των δειγμάτων είναι μεγάλες, η cross-validation μπορεί να μην είναι η βέλτιστη επιλογή.

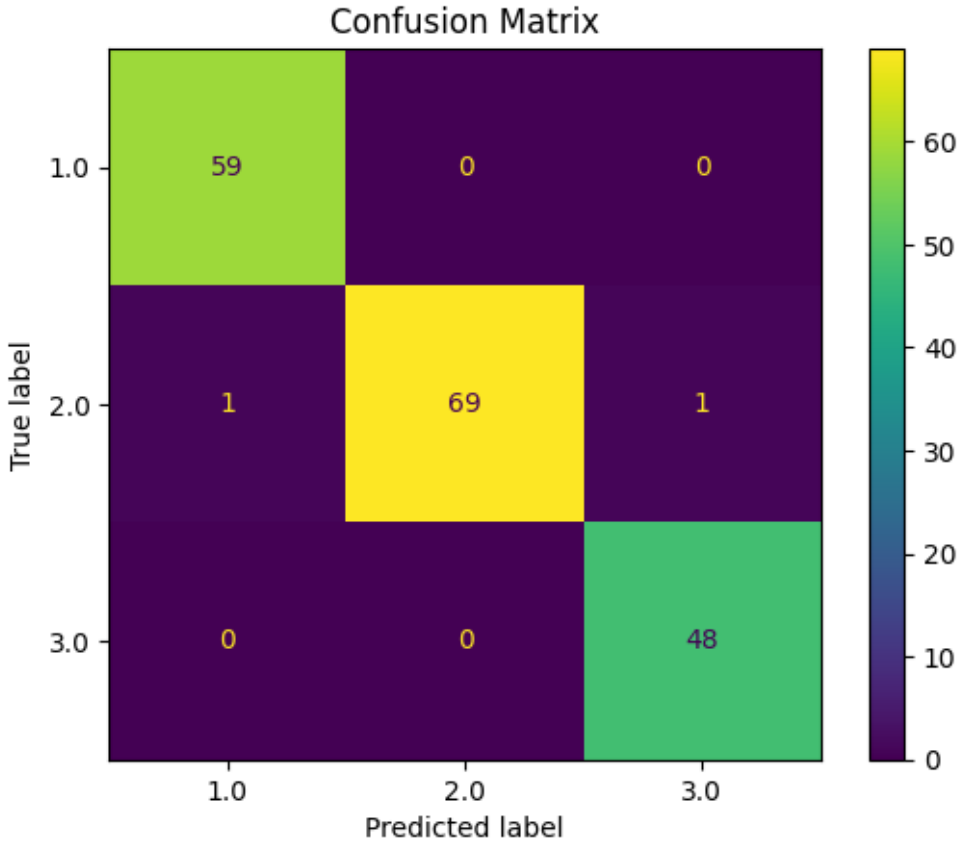
Ο πίνακας σύγχυσης (confusion matrix) είναι ένα εργαλείο που χρησιμοποιείται στην ανάλυση ταξινόμησης για να κατανοήσουμε την απόδοση ενός αλγορίθμου ταξινόμησης. Ο πίνακας αυτός παρουσιάζει τις πληροφορίες για τις πραγματικές κλάσεις σε σχέση με τις προβλεπόμενες κλάσεις από τον αλγόριθμο. Η κύρια διαγώνιος δείχνει τις περιπτώσεις όπου οι πραγματικές κλάσεις έχουν προβλεφθεί σωστά, ενώ οι εκτός

διαγωνίου τιμές αντιπροσωπεύουν λανθασμένες προβλέψεις. Όσο υψηλότερες είναι οι τιμές στη διαγώνιο και όσο χαμηλότερες είναι οι τιμές εκτός διαγώνιου, τόσο καλύτερη είναι η απόδοση του μοντέλου.




*Πίνακας 1: Πίνακας σύγχυσης για όλα τα δεδομένα με 5 χαρακτηριστικά.*

Η κλάση 1 έχει προβλεφθεί σωστά 51 φορές και λανθασμένα 1 φορά ως κλάση 2 και 7 φορές ως κλάση 3.
Η κλάση 2 έχει προβλεφθεί σωστά 58 φορές και λανθασμένα 3 φορές ως κλάση 1 και 10 φορές ως κλάση 3.
Η κλάση 3 έχει προβλεφθεί σωστά 36 φορές και λανθασμένα 3 φορές ως κλάση 1 και 9 φορές ως κλάση 2.




*Πίνακας 2: Πίνακας σύγχυσης για όλα τα δεδομένα με όλα τα χαρακτηριστικά.*

Η κλάση 1 έχει προβλεφθεί σωστά 59 φορές και καμία φορά λανθασμένη ως κλάση 2 ή 3.
Η κλάση 2 έχει προβλεφθεί σωστά 69 φορές και λανθασμένα 1 φορές ως κλάση 1 και 1 φορά ως κλάση 3.
Η κλάση 3 έχει προβλεφθεί σωστά 48 φορές και λανθασμένα καμία φορά λανθασμένη ως κλάση 1 ή 2.

Συγκρίνοντας τους δύο πίνακες σύγχυσης συμπεράνουμε ότι η μείωση του αριθμού των χαρακτηριστικών (διαστάσεων) οδήγησε σε χειρότερη απόδοση του μοντέλου, με αυξημένα λάθη ταξινόμησης σε όλες τις κλάσεις. Αυτό συμβαίνει συνήθως όταν τα χαρακτηριστικά που αφαιρέθηκαν περιέχουν σημαντικές πληροφορίες που βοηθούν το μοντέλο να διακρίνει τις κλάσεις.