**Μέρος Β**

**Ονοματεπώνυμο: Καραΐσκος Κωνσταντίνος**

**Α.Μ. : 1072636**

**Ερώτημα 4.** Με χρήση της μεθόδου 5-fold cross validation και του matlab, εκπαιδεύσετε τους παρακάτω ταξινομητές, να παρουσιάστε και να σχολιάσετε την απόδοσή τους:

* Support Vector Machines (με Radial Basis Function kernel function):
  + Ρυθμίστε την παράμετρο C με διαδοχική αναζήτηση του βέλτιστου C στο διάστημα 1-200 με βήμα 5 και χρήση γραμμικών SVM. Στη συνέχεια, ρυθμίστε την παράμετρο γ με χρήση του βέλτιστου C που βρέθηκε από πριν, και διαδοχική αναζήτηση του βέλτιστου γ στο διάστημα 0-10 με βήμα 0.5 και χρήση RBFSVM.
* Ταξινομητής K-Κοντινότερου Γείτονα
  + Ρυθμίστε την παράμετρο Κ με διαδοχική αναζήτηση της βέλτιστης τιμής στο διάστημα 3-15.

Ποιο συγκεκριμένα ζητείται να παρουσιάσετε για κάθε ταξινομητή την μέση απόδοση του με χρήση 5 fold cross validation σε σχέση με την μετρική του γεωμετρικού μέσου (Geometric Mean) της ευαισθησίας (Sensitivity) και της ειδίκευσής (Specificity) του:

Geometric Mean = sqrt (Sensitivity \* Specificity)

Η μετρική αυτή χρησιμοποιείται για προβλήματα ταξινόμησης όπου παραδείγματα εκπαίδευσης της μίας κλάσης είναι περισσότερα από τα παραδείγματα εκπαίδευσης της άλλης κλάσης.

Στη συνέχεια, παρουσιάστε τα ενδιάμεσα αποτελέσματα που πήρατε από τα πειράματα για την ρύθμιση των παραμέτρων των αλγορίθμων. Γιατί η κάθε μέθοδος ταξινόμησης δίνει διαφορετικά αποτελέσματα; Με βάση τα παραπάνω αποτελέσματα ποιά από τις μεθόδους προτείνετε εσείς να χρησιμοποιηθεί για το παραπάνω πρόβλημα και γιατί;

Απάντηση:

* Ενδιάμεσα αποτελέσματα ρύθμισης των παραμέτρων, πρώτα της C και μετά της γ, και έξοδος του ταξινομητή Support Vector Machines για τα βέλτιστα C και γ:

C:\Users\ν\AppData\Local\Programs\Python\Python38-32\python.exe D:/pythonProjects/Python/main.py

C= 1 \scores: [0.71551724 0.74137931 0.69827586 0.73275862 0.70434783] \scores.mean: 0.718455772113943 \previous max\_score: -1 \previous optimal\_c 0

C= 6 \scores: [0.72413793 0.74137931 0.72413793 0.75 0.72173913] \scores.mean: 0.7322788605697153 \previous max\_score: 0.718455772113943 \previous optimal\_c 1

C= 11 \scores: [0.72413793 0.74137931 0.73275862 0.75862069 0.73043478] \scores.mean: 0.7374662668665668 \previous max\_score: 0.7322788605697153 \previous optimal\_c 6

C= 16 \scores: [0.71551724 0.73275862 0.74137931 0.76724138 0.73913043] \scores.mean: 0.7392053973013493 \previous max\_score: 0.7374662668665668 \previous optimal\_c 11

C= 21 \scores: [0.71551724 0.73275862 0.74137931 0.75862069 0.73913043] \scores.mean: 0.7374812593703147 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 26 \scores: [0.71551724 0.73275862 0.74137931 0.75 0.73913043] \scores.mean: 0.7357571214392804 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 31 \scores: [0.71551724 0.73275862 0.74137931 0.75 0.73913043] \scores.mean: 0.7357571214392804 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 36 \scores: [0.71551724 0.73275862 0.74137931 0.75 0.73913043] \scores.mean: 0.7357571214392804 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 41 \scores: [0.70689655 0.73275862 0.74137931 0.75 0.73913043] \scores.mean: 0.7340329835082459 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 46 \scores: [0.70689655 0.73275862 0.74137931 0.75 0.73913043] \scores.mean: 0.7340329835082459 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 51 \scores: [0.70689655 0.73275862 0.74137931 0.75 0.73913043] \scores.mean: 0.7340329835082459 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 56 \scores: [0.70689655 0.73275862 0.74137931 0.75 0.73913043] \scores.mean: 0.7340329835082459 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 61 \scores: [0.70689655 0.73275862 0.74137931 0.75 0.73913043] \scores.mean: 0.7340329835082459 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 66 \scores: [0.70689655 0.73275862 0.73275862 0.75 0.73913043] \scores.mean: 0.7323088455772113 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 71 \scores: [0.70689655 0.73275862 0.73275862 0.75 0.73913043] \scores.mean: 0.7323088455772113 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 76 \scores: [0.70689655 0.73275862 0.73275862 0.75 0.73913043] \scores.mean: 0.7323088455772113 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 81 \scores: [0.70689655 0.73275862 0.73275862 0.75 0.73913043] \scores.mean: 0.7323088455772113 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 86 \scores: [0.69827586 0.73275862 0.74137931 0.75 0.73913043] \scores.mean: 0.7323088455772113 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 91 \scores: [0.69827586 0.73275862 0.74137931 0.75 0.73913043] \scores.mean: 0.7323088455772113 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 96 \scores: [0.69827586 0.73275862 0.74137931 0.75 0.73913043] \scores.mean: 0.7323088455772113 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 101 \scores: [0.69827586 0.73275862 0.74137931 0.75 0.73913043] \scores.mean: 0.7323088455772113 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 106 \scores: [0.69827586 0.73275862 0.74137931 0.75 0.73913043] \scores.mean: 0.7323088455772113 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 111 \scores: [0.69827586 0.73275862 0.74137931 0.75 0.73913043] \scores.mean: 0.7323088455772113 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 116 \scores: [0.69827586 0.73275862 0.74137931 0.75 0.73913043] \scores.mean: 0.7323088455772113 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 121 \scores: [0.69827586 0.73275862 0.74137931 0.75 0.73913043] \scores.mean: 0.7323088455772113 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 126 \scores: [0.69827586 0.73275862 0.74137931 0.75 0.73913043] \scores.mean: 0.7323088455772113 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 131 \scores: [0.69827586 0.73275862 0.74137931 0.74137931 0.73913043] \scores.mean: 0.7305847076461769 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 136 \scores: [0.69827586 0.73275862 0.74137931 0.74137931 0.73913043] \scores.mean: 0.7305847076461769 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 141 \scores: [0.69827586 0.73275862 0.74137931 0.74137931 0.73913043] \scores.mean: 0.7305847076461769 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 146 \scores: [0.69827586 0.73275862 0.74137931 0.74137931 0.73913043] \scores.mean: 0.7305847076461769 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 151 \scores: [0.69827586 0.73275862 0.74137931 0.74137931 0.73913043] \scores.mean: 0.7305847076461769 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 156 \scores: [0.69827586 0.73275862 0.74137931 0.74137931 0.73913043] \scores.mean: 0.7305847076461769 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 161 \scores: [0.69827586 0.73275862 0.74137931 0.74137931 0.73913043] \scores.mean: 0.7305847076461769 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 166 \scores: [0.69827586 0.73275862 0.74137931 0.74137931 0.73913043] \scores.mean: 0.7305847076461769 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 171 \scores: [0.69827586 0.73275862 0.74137931 0.74137931 0.73913043] \scores.mean: 0.7305847076461769 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 176 \scores: [0.69827586 0.73275862 0.74137931 0.74137931 0.73913043] \scores.mean: 0.7305847076461769 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 181 \scores: [0.69827586 0.73275862 0.74137931 0.74137931 0.73913043] \scores.mean: 0.7305847076461769 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 186 \scores: [0.69827586 0.73275862 0.74137931 0.74137931 0.73913043] \scores.mean: 0.7305847076461769 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 191 \scores: [0.69827586 0.73275862 0.74137931 0.74137931 0.73913043] \scores.mean: 0.7305847076461769 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

C= 196 \scores: [0.69827586 0.73275862 0.74137931 0.74137931 0.73913043] \scores.mean: 0.7305847076461769 \previous max\_score: 0.7392053973013493 \previous optimal\_c 16

gamma= 0.5 \scores: [0.71551724 0.71551724 0.71551724 0.70689655 0.71304348] \scores.mean: 0.7132983508245877 \previous max\_score: -1 \previous optimal\_g: 0

gamma= 1.0 \scores: [0.71551724 0.72413793 0.70689655 0.70689655 0.71304348] \scores.mean: 0.7132983508245877 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

gamma= 1.5 \scores: [0.71551724 0.69827586 0.70689655 0.70689655 0.71304348] \scores.mean: 0.7081259370314842 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

gamma= 2.0 \scores: [0.70689655 0.70689655 0.69827586 0.69827586 0.73043478] \scores.mean: 0.7081559220389805 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

gamma= 2.5 \scores: [0.68965517 0.71551724 0.67241379 0.72413793 0.71304348] \scores.mean: 0.7029535232383808 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

gamma= 3.0 \scores: [0.68965517 0.72413793 0.6637931 0.73275862 0.71304348] \scores.mean: 0.7046776611694152 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

gamma= 3.5 \scores: [0.68103448 0.72413793 0.65517241 0.71551724 0.72173913] \scores.mean: 0.6995202398800601 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

gamma= 4.0 \scores: [0.68103448 0.72413793 0.64655172 0.69827586 0.70434783] \scores.mean: 0.6908695652173913 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

gamma= 4.5 \scores: [0.68103448 0.73275862 0.6637931 0.69827586 0.71304348] \scores.mean: 0.6977811094452773 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

gamma= 5.0 \scores: [0.68965517 0.73275862 0.67241379 0.68965517 0.71304348] \scores.mean: 0.6995052473763119 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

gamma= 5.5 \scores: [0.69827586 0.73275862 0.68103448 0.67241379 0.72173913] \scores.mean: 0.7012443778110944 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

gamma= 6.0 \scores: [0.69827586 0.73275862 0.68103448 0.65517241 0.72173913] \scores.mean: 0.6977961019490253 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

gamma= 6.5 \scores: [0.69827586 0.73275862 0.68965517 0.64655172 0.71304348] \scores.mean: 0.6960569715142428 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

gamma= 7.0 \scores: [0.69827586 0.75 0.70689655 0.63793103 0.71304348] \scores.mean: 0.7012293853073464 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

gamma= 7.5 \scores: [0.70689655 0.73275862 0.69827586 0.63793103 0.71304348] \scores.mean: 0.6977811094452775 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

gamma= 8.0 \scores: [0.68103448 0.72413793 0.68965517 0.62931034 0.71304348] \scores.mean: 0.6874362818590705 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

gamma= 8.5 \scores: [0.64655172 0.69827586 0.68965517 0.62931034 0.71304348] \scores.mean: 0.675367316341829 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

gamma= 9.0 \scores: [0.63793103 0.68965517 0.68965517 0.62931034 0.71304348] \scores.mean: 0.6719190404797601 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

gamma= 9.5 \scores: [0.62931034 0.67241379 0.68103448 0.62931034 0.70434783] \scores.mean: 0.6632833583208396 \previous max\_score: 0.7132983508245877 \previous optimal\_g: 0.5

Support Vector Machines scores:

[0.62931034 0.67241379 0.68103448 0.62931034 0.70434783]

0.66 accuracy with standard deviation of 0.03

Optimal C: 16 and optimal gamma: 0.5

---------------------------------------------------------

Specificity scores:

[0.24242424 0.24242424 0.3030303 0.27272727 0.15151515]

0.24 accuracy with standard deviation of 0.05

---------------------------------------------------------

Sensitivity scores:

[0.78313253 0.84337349 0.8313253 0.77108434 0.92682927]

0.83 accuracy with standard deviation of 0.06

---------------------------------------------------------

Geometric mean scores:

[0.43571815 0.4521661 0.5019131 0.45858012 0.37473814]

0.44 accuracy with standard deviation of 0.04

---------------------------------------------------------

Process finished with exit code 0

* Ενδιάμεσα αποτελέσματα ρύθμισης της παραμέτρου k και έξοδος του ταξινομητή k-Nearest Neighbor για το βέλτιστο αριθμό γειτόνων k:

C:\Users\ν\AppData\Local\Programs\Python\Python38-32\python.exe D:/pythonProjects/Python/main.py

k= 3 \scores: [0.59482759 0.68103448 0.68103448 0.62068966 0.71304348] \scores.mean: 0.6581259370314843 \previous max\_score: -1 \previous optimal\_n: 0

k= 4 \scores: [0.67241379 0.70689655 0.75 0.62931034 0.72173913] \scores.mean: 0.696071964017991 \previous max\_score: 0.6581259370314843 \previous optimal\_n: 3

k= 5 \scores: [0.59482759 0.67241379 0.65517241 0.62931034 0.74782609] \scores.mean: 0.6599100449775113 \previous max\_score: 0.696071964017991 \previous optimal\_n: 4

k= 6 \scores: [0.60344828 0.68965517 0.71551724 0.6637931 0.72173913] \scores.mean: 0.6788305847076461 \previous max\_score: 0.696071964017991 \previous optimal\_n: 4

k= 7 \scores: [0.57758621 0.68965517 0.68965517 0.6637931 0.71304348] \scores.mean: 0.6667466266866566 \previous max\_score: 0.696071964017991 \previous optimal\_n: 4

k= 8 \scores: [0.62931034 0.68965517 0.69827586 0.6637931 0.72173913] \scores.mean: 0.6805547226386807 \previous max\_score: 0.696071964017991 \previous optimal\_n: 4

k= 9 \scores: [0.56896552 0.65517241 0.64655172 0.65517241 0.73913043] \scores.mean: 0.6529985007496251 \previous max\_score: 0.696071964017991 \previous optimal\_n: 4

k= 10 \scores: [0.60344828 0.70689655 0.71551724 0.67241379 0.73913043] \scores.mean: 0.6874812593703149 \previous max\_score: 0.696071964017991 \previous optimal\_n: 4

k= 11 \scores: [0.59482759 0.6637931 0.65517241 0.65517241 0.73043478] \scores.mean: 0.659880059970015 \previous max\_score: 0.696071964017991 \previous optimal\_n: 4

k= 12 \scores: [0.62931034 0.72413793 0.68103448 0.68965517 0.72173913] \scores.mean: 0.6891754122938532 \previous max\_score: 0.696071964017991 \previous optimal\_n: 4

k= 13 \scores: [0.62931034 0.69827586 0.65517241 0.67241379 0.70434783] \scores.mean: 0.6719040479760119 \previous max\_score: 0.696071964017991 \previous optimal\_n: 4

k= 14 \scores: [0.62931034 0.68103448 0.65517241 0.68965517 0.71304348] \scores.mean: 0.6736431784107946 \previous max\_score: 0.696071964017991 \previous optimal\_n: 4

k= 15 \scores: [0.62068966 0.68965517 0.6637931 0.68103448 0.70434783] \scores.mean: 0.6719040479760119 \previous max\_score: 0.696071964017991 \previous optimal\_n: 4

K-nearest neighbors scores for n=4:

[0.67241379 0.70689655 0.75 0.62931034 0.72173913]

0.70 accuracy with standard deviation of 0.04

---------------------------------------------------------

Specificity scores:

[0.18181818 0.27272727 0.33333333 0.15151515 0.12121212]

0.21 accuracy with standard deviation of 0.08

---------------------------------------------------------

Sensitivity scores:

[0.86746988 0.87951807 0.91566265 0.81927711 0.96341463]

0.89 accuracy with standard deviation of 0.05

---------------------------------------------------------

Geometric mean scores:

[0.39714204 0.48976379 0.55246799 0.35232499 0.34172728]

0.43 accuracy with standard deviation of 0.08

Process finished with exit code 0

Για του τρείς ταξινομητές πήραμε τις εξής αποδόσεις:

GaussianNB: 0.57 accuracy

Support Vector Machines: 0.66 accuracy

K-Nearest Neighbors: 0.70 accuracy

Παρατηρούμε σημαντική διαφορά στην ακρίβεια ταξινόμησης μεταξύ του GaussianNB και των άλλων δύο. Αυτό συμβαίνει γιατί ενώ ο GaussianNB θεωρεί τα χαρακτηριστικά ανεξάρτητα μεταξύ τους, ο SVM ψάχνει για συσχετίσεις μεταξύ των χαρακτηριστικών αφού χρησιμοποιούμε rbf kernel και όχι γραμμικό. Ακόμα αφού ο GaussianNB είναι γραμμικός, σε αντίθεση με τους άλλους δύο, και έτσι δεν μπορούμε να σχεδιάσουμε το ίδιο αποδοτικά όρια διαχωρισμού των δύο κλάσεων με αυτά των SVM και K-NN.

Μεταξύ του SVM και του k-NN παρατηρείται επίσης μία μικρή διαφορά στην απόδοση, η οποία είναι αναμενόμενη αφού το πλήθος των δεδομένων εκπαίδευσης είναι αρκετά μεγαλύτερο από το πλήθος των χαρακτηριστικών των δεδομένων και έτσι ο K-NN αποδίδει λίγο καλύτερα.

Καθώς ο ταξινομητής K-NN παρουσιάζει την καλύτερη απόδοση σε σχέση με του άλλους δύο και καθώς το πλήθος των δεδομένων εκπαίδευσης είναι αρκετά μεγαλύτερο από το πλήθος των χαρακτηριστικών των δεδομένων, όπως αναφέρθηκε και παραπάνω, προτείνεται αυτός για χρήση στο πρόβλημά μας.