**Εργαστηριακή Άσκηση**

**για το μάθημα Θεωρία Αποφάσεων**

**2022-2023**

**Ονοματεπώνυμο: Πύργας Αθανάσιος** **ΑΜ: 1041866 (παλαιός: 236189)**

**Ερώτημα 4.**

**a)** Support Vector Machines (με Radial Basis Function kernel function)
Το αρχείο κώδικα που αντιστοιχεί στο παραπάνω ερώτημα είναι το **e4a.py**. Πραγματοποιούμε grid search για **C από 1 έως 201 με βήμα 5** και **gamma από 0 έως 10 με βήμα 0.5**. Τα αποτελέσματα δείχουν ότι οι καλύτερες τιμές για τις υπερπαραμέτρους C και gamma είναι 86 και 7.5 αντίστοιχα (βασιζόμενοι στον καλύτερο γεωμετρικό μέσο Geometric mean) και τα αποτελέσματα φαίνονται στην παρακάτω εικόνα.

```
Best Geometric Mean for C = 86 and Gamma = 7.50
0.5344471629188312
Corresponding Sensitivity
0.7583098746026575
Corresponding Specificity
0.38372329472329475
Corresponding Accuracy
0.6485116416150898
```

Παρατηρούμε ότι ο αλγόριθμος για το συγκεκριμένο σύνολο δεδομένων, αντίθετα με τον Naive Bayes Classifier που εξετάσαμε στην προηγούμενη εργασία, έχει σχετικά υψηλό Sensitivity αλλά έχει πολύ χαμηλό Specificity. Δηλαδή αυτό σημαίνει ότι εντοπίζει αρκετά καλά τους ανθρώπους που αντιμετωπίζουν πρόβλημα αλλά δεν κάνει καλή δουλειά στον εντοπισμο των ανθρώπων που είναι υγιείς. Παρατηρούμε επίσης ότι, **παραδόξως**, έχει χαμηλότερο Geometric Mean από τον Naive Bayes Classifier.

**b)** Ταξινομητής K-Κοντινότερου Γείτονα
Το αρχείο κώδικα που αντιστοιχεί στο παραπάνω ερώτημα είναι το **e4b.py**. Εκτελούμε τον αλγόριθμο για αριθμό κοντινότερων γειτόνων από 3 έως 15 όπως μας ζητείται. Τα αποτελέσματα δειχνουν ότι η καλύτερη τιμή **n_neighbors** (βασιζόμενοι στον καλύτερο γεωμετρικό μέσο Geometric mean) είναι η τιμή **8**.

```
Best Geometric Mean for K = 8
0.5341575614382584
Corresponding Sensitivity
0.7210902241732225
Corresponding Specificity
0.3967001287001287
Corresponding Accuracy
0.6278809313292072
```

Αντίστοιχα με τον SVM αλγόριθμο, ο kNN αλγόριθμος έχει σχετικά υψηλό Sensitivity αλλά πολύ χαμηλό Specificity. Και αυτός ο αλγόριθμος, έχει χαμηλότερο Geometric Mean από τον Naive Bayes Classifier.

Ας εκτελέσουμε τα ιδια πειράματα ομως και με ισορροπημένο σετ δεδομένων. Το σετ δεδομένων που μας δίνεται δεν είναι ισορροπημένο καθώς αποτελείται από 416 περιπτώσεις ασθενών και από μόλις 167 υγειών ατόμων. Για την εξισορρόπηση των δεδομένων θα χρησιμοποιήσουμε τον **αλγόριθμο SMOTE**.

**a)** Support Vector Machines (με Radial Basis Function kernel function)
Το αρχείο κώδικα που αντιστοιχεί στο παραπάνω ερώτημα είναι το **e4a_bal.py**. Πραγματοποιούμε grid search για **C από 1 έως 201 με βήμα 5** και **gamma από 0 έως 10 με βήμα 0.5**. Τα αποτελέσματα δείχουν ότι οι καλύτερες τιμές για τις υπερπαραμέτρους C και gamma έιναι 76 και 5.00 (βασιζόμενοι στον καλύτερο Γεωμετρικό Μέσο Geometric Mean) και τα αποτελέσματα φαίνονται στην παρακάτω εικόνα.

```
Best Geometric Mean for C = 76 and Gamma = 5.00
0.8223827781594348
Corresponding Sensitivity
0.7449105995982604
Corresponding Specificity
0.9099001330672151
Corresponding Accuracy
0.8256835726138085
```

Οι νέες τιμές είναι φανερά βελτιωμένες σε σχέση με την περίπτωση που έχουμε μη ισορροπημένα δεδομένα.

**b)** Ταξινομητής k-Κοντινότερου Γείτονα
Το αρχείο κώδικα που αντιστοιχεί στο παραπάνω ερώτημα είναι το **e4b_bal.py**. Εκτελούμε τον αλγόριθμο για αριθμό κοντινότερων γειτόνων από 3 έως 15 όπως μας ζητείται. Τα αποτελέσματα δειχνουν ότι η καλύτερη τιμή **n_neighbors** (βασιζόμενοι στον καλύτερο γεωμετρικό μέσο Geometric mean) είναι η τιμή **3**.

```
Best Geometric Mean for K = 3
0.7152104043286162
Corresponding Sensitivity
0.584365291571174
Corresponding Specificity
0.8762725030643214
Corresponding Accuracy
0.7294783926123656
```

Επίσης πολύ βελτιωμένα αποτελέσματα σε σχέση με τα αποτελέσματα στην περίπτωση των μη ισορροπημένων δεδομένων. **Η εξισορρόπηση των δεδομένων παίζει σημαντικό ρόλο!**

**Ερώτημα 5.**

Το αρχείο κώδικα που αντιστοιχεί στο παραπάνω ερώτημα είναι το **e5.py**. Εκτελούμε t test μεταξύ των δύο κλάσεων (υγειείς και ασθενείς) για κάθε ξεχωριστό χαρακτηριστικό του προβλήματος μας. Αποθηκεύουμε τις τιμές του t statistic σε κάθε περίπτωση και διαλέγουμε τις 5 καλύτερες.

```
Best Features Indices and Names
[0 1 7 8 9]
['Age' 'Gender' 'SGPT' 'SGOT' 'Alkphos']
```

Με αυτά τα 5 τώρα χαρακτηριστικά που προέκυψαν παραπάνω, θα εκπαιδεύσουμε τον βέλτιστο ταξινομητή που βρήκαμε στο προηγούμενο ερώτημα **δηλαδή ενα SVM με υπερπαραμέτρους C=76 και gamma = 5.00**. Τα αποτελέσματα φαίνονται στην παρακάτω εικόνα.

```
Geometric Mean per fold
[0.68529966 0.72456884 0.73246671 0.64224889 0.73399093]

Mean Geometric Mean
0.7037150050673657
```

Παρατηρούμε ότι υπάρχει μια πτώση στην απόδοση του ταξινομητή της τάξεως του 10%. Παρόλα αυτά όμως, είναι σημαντικό ότι **έχουμε μειώσει την διάσταση του προβλήματος στο μισό** .