PROJECT DATA MINING

ΚΥΡΙΑΚΟΥΛΟΠΟΥΛΟΣ ΚΑΛΛΙΝΙΚΟΣ
ΑΜ:1084583

Contents

Καταγραφή του περιβάλλοντος υλοποίησης

Περιβάλλον Υλοποίησης:

CPU model	Intel(R) Core(TM) i7-1065G7
CPU clock speed	1.30GHz
Physical CPU cores	4
Logical CPU cores	8
RAM	8
Secondary Storage Type	SSD
Python version	3.12.2

Γενικές βιβλιοθήκες:

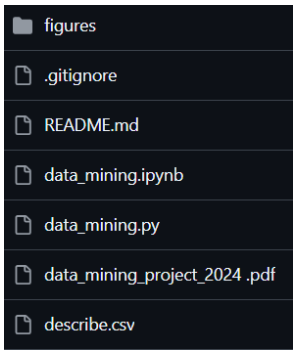
Pandas, numpy, os, matplotlib, seaborn, yellowbrick

Βιβλιοθήκη για classification και clustering:

Sklearn

Οι παραπάνω βιβλιοθήκες εγκαθίστανται με την εντολή **pip install library_name**

Περιγραφή Αρχείων



Data_mining.py: είναι το κύριο αρχείο στο οποίο υλοποιούνται όλα τα ερωτήματα(1,2,3).

Data_mining.ipynb: είναι notebook το οποίο υλοποιείται μόνο το πρώτο ερώτημα. Το έχω υλοποιήσει και σε notebook το πρώτο ερώτημα ώστε να εμφανίζονται καλύτερα τα αποτελέσματα.

Figures: είναι φάκελος στον οποίο αποθηκεύονται όλα τα γραφήματα που δημιουργούνται από το αρχείο data_minig.py .

Describe.csv: είναι αρχείο .csv το οποίο έχει το αποτέλεσμα της εντολής df.describe(), δηλαδή τα στατιστικά στοιχεία των δεδομένων .

Τον φάκελο harth (με τα δεδομένα) θα πρέπει να τον τοποθετήσετε στο ίδιο directory με τα παραπάνω αρχεία ώστε να εισάγει αρχείο data_mining.py τα δεδομένα. Δεν υπάρχει λόγω μεγέθους (815MB).

Περιγραφή του τρόπου υλοποίησης

Ερώτημα 1

Αφού εισάγουμε στο πρόγραμμα τα αρχεία .csv με τα δεδομένα δημιουργείται ένα dataframe που περιέχει τις μετρήσεις από όλους τους συμμετέχοντες. Επίσης για διαχωρισμό της κάθε εγγραφής που υπάρχει στο dataframe εισάγεται μια επιπλέον στήλη “userID” ώστε να γνωρίζουμε σε ποιόν ανήκει η μέτρηση.

Το dataframe που δημιουργείται έχει τις στήλες:

timestamp, back_x, back_y, back_z, thigh_x, thigh_y, thigh_z, label

Για αυτό το dataframe παίρνουμε τα παρακάτω αποτελέσματα:

Πληροφορίες dataframe: df.info()

```
Dataframe Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6461328 entries, 0 to 6461327
Data columns (total 8 columns):
 #   Column     Dtype
---  ------     -----
 0   timestamp  object
 1   back_x     float64
 2   back_y     float64
 3   back_z     float64
 4   thigh_x    float64
 5   thigh_y    float64
 6   thigh_z    float64
 7   label      int64
dtypes: float64(6), int64(1), object(1)
memory usage: 394.4+ MB
None
```

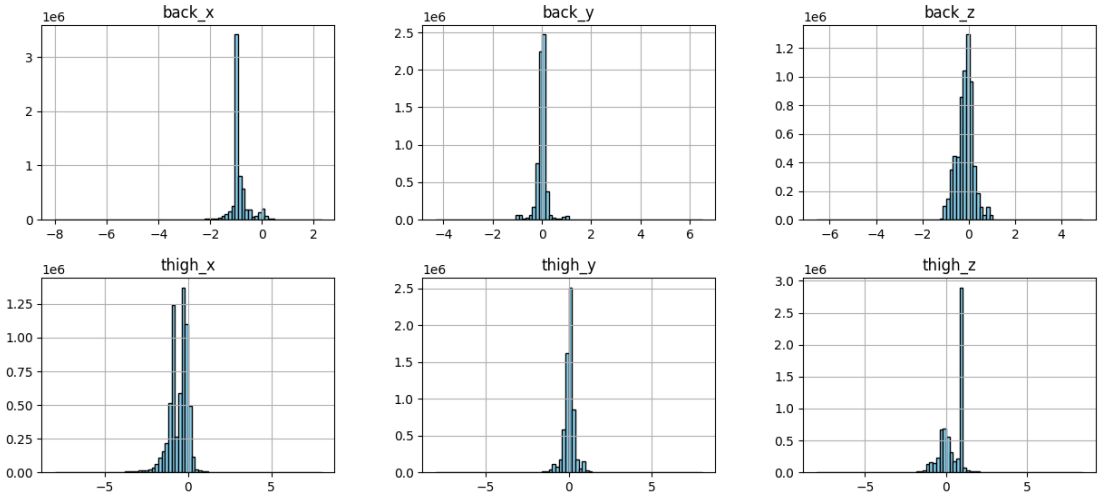
Έλεγχος για μη ύπαρξη τιμών στα δεδομένα:

```
Check empty data:
timestamp    0
back_x       0
back_y       0
back_z       0
thigh_x      0
thigh_y      0
thigh_z      0
label        0
dtype: int64
```

Στατιστικά δεδομένα: df.describe()

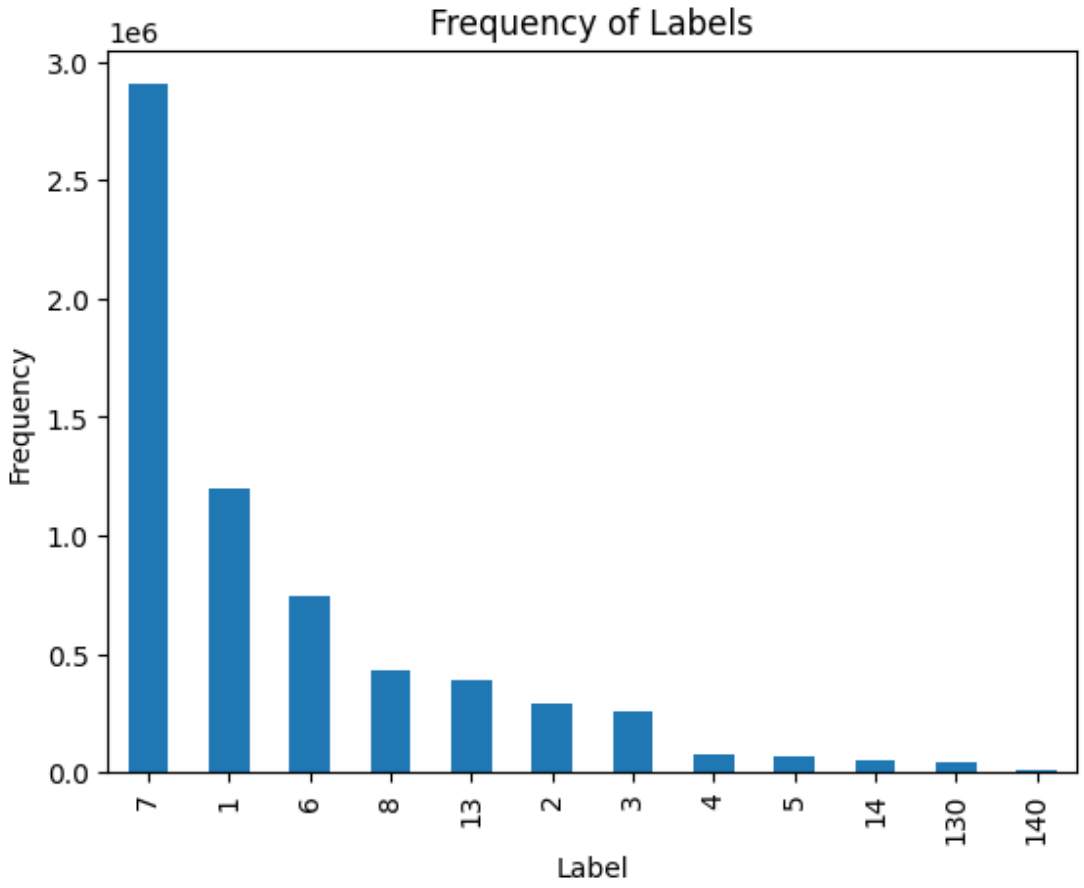
	back_x	back_y	back_z	thigh_x	thigh_y	thigh_z	label
count	6.461328e+06	6.461328e+06	6.461328e+06	6.461328e+06	6.461328e+06	6.461328e+06	6.461328e+06
mean	-8.849574e-01	-1.326128e-02	-1.693779e-01	-5.948883e-01	2.087665e-02	3.749160e-01	6.783833e+00
std	3.775916e-01	2.311709e-01	3.647385e-01	6.263466e-01	3.884511e-01	7.360983e-01	1.143238e+01
min	-8.000000e+00	-4.307617e+00	-6.574463e+00	-8.000000e+00	-7.997314e+00	-8.000000e+00	1.000000e+00
25%	-1.002393e+00	-8.312914e-02	-3.720700e-01	-9.742110e-01	-1.000873e-01	-1.557138e-01	3.000000e+00
50%	-9.748998e-01	2.593677e-03	-1.374510e-01	-4.217309e-01	3.262909e-02	7.004390e-01	7.000000e+00
75%	-8.123032e-01	7.251000e-02	4.647321e-02	-1.678755e-01	1.549512e-01	9.486747e-01	7.000000e+00
max	2.291708e+00	6.491943e+00	4.909483e+00	7.999756e+00	7.999756e+00	8.406235e+00	1.400000e+02

Ιστόγραμμα κάθε στήλης:

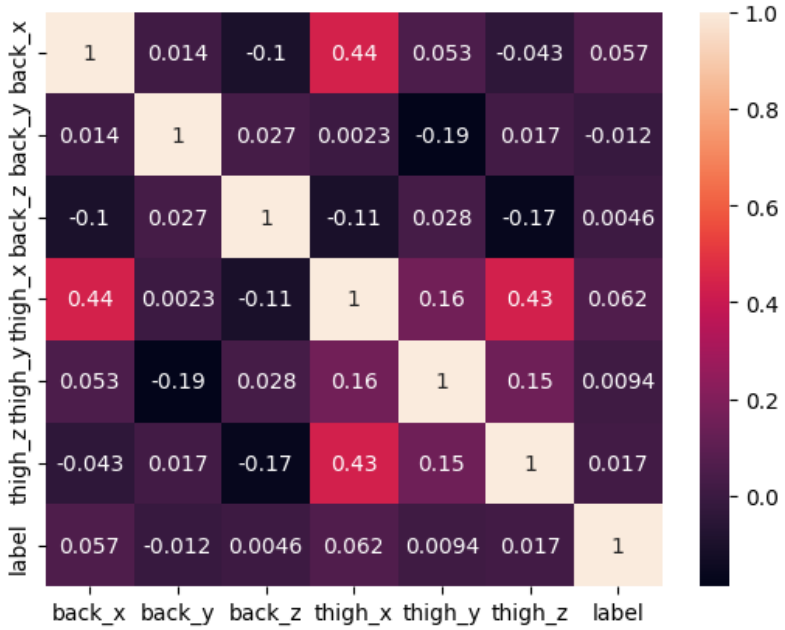


Σχολιασμός: οι γραφικές παραστάσεις των στηλών back_x, back_y, back_z, thigh_x, thigh_y, thigh_z φαίνεται να ακολουθούν κανονική κατανομή. Βρίσκονται γύρω από το 0 και παρουσιάζουν κορύφωση στο 0. Βέβαια παρατηρούνται διαφοροποιήσεις στις γραφικές παραστάσεις των back_x και thigh_x που δεν έχουν κέντρο στο 0 και στο διάγραμμα thigh_z κυρίως, όπου υπάρχουν τιμές που αποκλίνουν από τις υπόλοιπες.

Ιστόγραμμα ετικετών (πιο αναλυτικό):

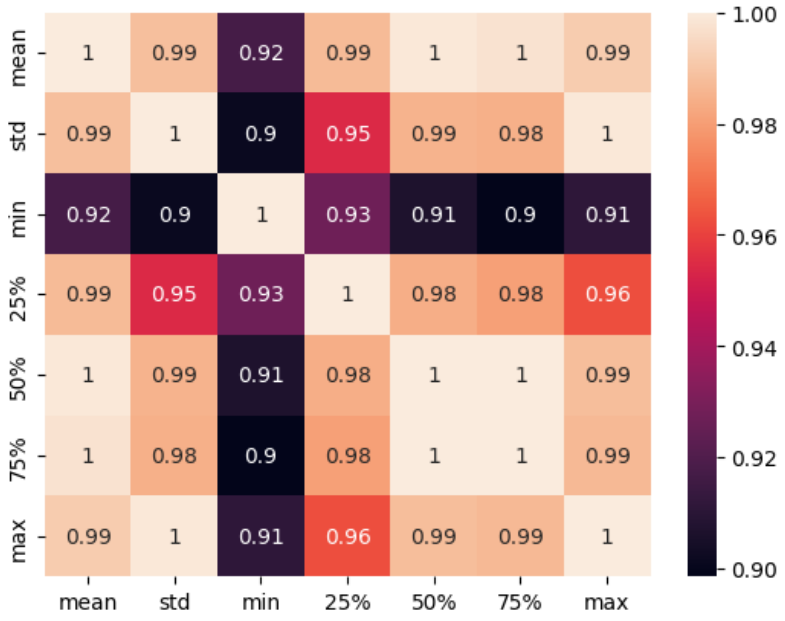


Διάγραμμα Συσχέτισης τιμών dataframe:



Σχολιασμός: συσχέτιση παρουσιάζεται μεταξύ των τιμών thigh_x, back_x και thigh_x, thigh_z. Οι τιμές αυτών των στηλών μοιάζουν μεταξύ τους. Για τις υπόλοιπες τιμές η συσχετίσεις είναι χαμηλές.

Διάγραμμα συσχέτισης στατιστικών δεδομένων:



Σχολιασμός: Από το διάγραμμα συμπεραίνεται ότι υπάρχουν μεγάλες συσχετίσεις μεταξύ των στατιστικών μεγεθών, εκτός από τις ελάχιστες τιμές που δεν έχουν συσχέτιση. Επομένως στα δεδομένα οι ελάχιστες τιμές έχουν μεγάλη απόσταση από τις υπόλοιπες τιμές.

Ερώτημα 2

Για το classifiacation χρησιμοποιήθηκαν Bayesian Networks, Neural Networks, Random Forests.

```
from sklearn.naive_bayes import GaussianNB
from sklearn.neural_network import MLPClassifier
from sklearn.ensemble import RandomForestClassifier
```

Αρχικά παράγω από το dataframe το σύνολο δεδομένων εισόδου και το σύνολο δεδομένων με τις ετικέτες. Για το σύνολο δεδομένων εισόδου διαγράφονται οι στήλες του dataframe ('label', 'timestamp', 'userID') και για τις ετικέτες κρατάω μόνο τη στήλη 'label'.

Έπειτα με τη συνάρτηση select_classifier(option=None) επιλέγεται ένας από τους 3 ταξινομητές και με τη συνάρτηση run_classification(X,Y,classifier = 1) γίνεται ο διαχωρισμός των δεδομένων σε σύνολο training και testing με αναλογία 0.3 και μετά πραγματοποιείται το training και το testing. Αφού εκτελεστούν και οι 3 ταξινομητές παράγεται διάγραμμα με τα αποτελέσματα ακρίβειας σε εκπαίδευση και δοκιμή.

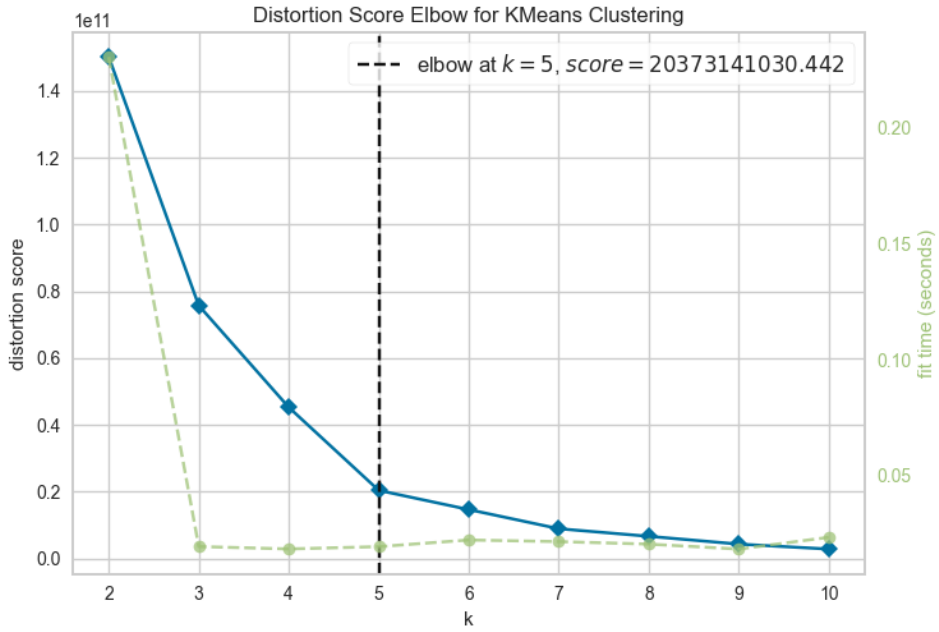
Τέλος με τη συνάρτηση classification(df, option=None) θα εκτελεστούν και οι 3 ταξινομητές, αλλά μπορεί με την παράμετρο option να εκτελεστεί μόνο 1 από τους 3.

Ερώτημα 3

Για το clustering προηγείται η διαδικασία επεξεργασίας των δεδομένων. Από το dataframe παράγω μια λίστα με 22 dataframes. Το καθένα αντιστοιχεί σε ένα συμμετέχοντα του πειράματος. Οι στήλες αυτών των dataframes είναι οι τιμές mean, std, min, 25%, 50%, 75%, max για κάθε sensor που χρησιμοποιείται, δηλαδή 7(στατιστικές μετρικές)*6(στήλες)=42 στήλες. Στο τέλος όταν δημιουργηθούν και τα 22 dataframes τα ενώνω σε ένα και κάθε γραμμή του νέο dataframe αντιστοιχεί σε έναν από τους 22 συμμετέχοντες.

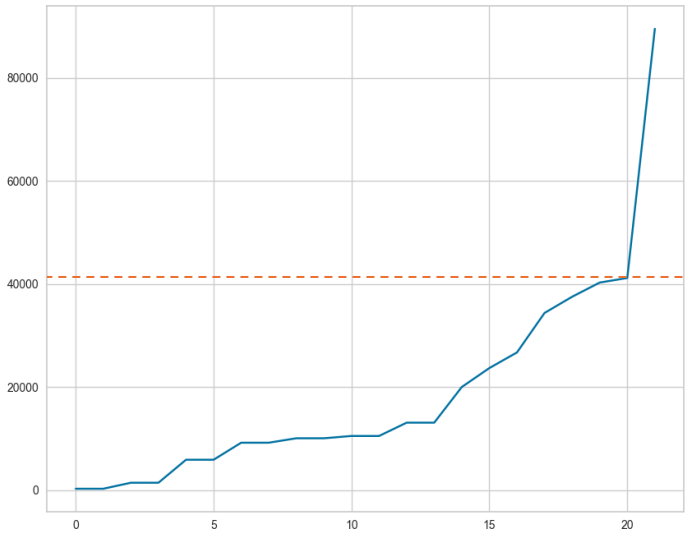
Μετά τη φάση της δημιουργίας του νέου dataframe το χρησιμοποιώ για τη διαδικασία του clustering. Η συσταδοποίηση γίνεται με Kmeans, MeanShift και DBscan. Με τη συνάρτηση select_clusterer(option) επιλέγεται ένας από τους 3 clusterers και εκτελείται η συσταδοποίηση. Ο κάθε αλγόριθμος συσταδοποίησης που χρησιμοποιείται εκτελείται με κλήση στην κατάλληλη συνάρτηση (kmeans, meanshift, dbscan).

Στη συνάρτηση kmeans() εκτελείται ο αλγόριθμος και μετά καλείται η συνάρτηση KElbowVisualizer ώστε να βρεθεί η βέλτιστη τιμή του αριθμού των συστάδων. Αφού βρεθεί εκτελείται ο αλγόριθμος με τη βέλτιστη τιμή του k.



Στη συνάρτηση meanshift() εκτελείται ο αλγόριθμος MeanShift. Ο αριθμός των συστάδων δεν δίνεται ως όρισμα διότι η συνάρτηση MeanShift() βρίσκει από μόνη της το βέλτιστο αριθμό συστάδων.

Στη συνάρτηση dbscan() πριν εκτελεστεί ο αλγόριθμος DBSCAN χρησιμοποιείται η μέθοδος εύρεσης των πλησιέστερων γειτόνων ώστε να υπολογιστεί η μέγιστη απόσταση που θα έχουν 2 σημεία ώστε να θεωρηθούν ότι ανήκουν στην ίδια γειτονιά. Από το διάγραμμα εντοπίζεται το σημείο elbow και η τιμή του σημείου θα είναι η παράμετρος eps που θα χρησιμοποιηθεί στον dbscan.



Καλώντας τη συνάρτηση clustering(X, option=None) θα εκτελεστούν και οι 3 αλγόριθμοι συσταδοποίησης, αλλά μπορεί με την παράμετρο option να εκτελεστεί μόνο 1 από τους 3.

Σχολιασμό των τελικών αποτελεσμάτων

Για το πρώτο ερώτημα ο σχολιασμός έχει γίνει στην προηγούμενη ενότητα.

Ερώτημα 2

Αποτελέσματα ταξινόμησης:

```
------------------------------------------------------------------------
Classifier RandomForestClassifier yeilds training accuracy of 0.9999909350776897
with a testing accuracy of 0.9016317074038936

------------------------------------------------------------------------
Classifier MLPClassifier yeilds training accuracy of 0.8760699095652397
with a testing accuracy of 0.8756102329809291

------------------------------------------------------------------------
Classifier GaussianNB yeilds training accuracy of 0.775219995715166
with a testing accuracy of 0.7749601604210485
```

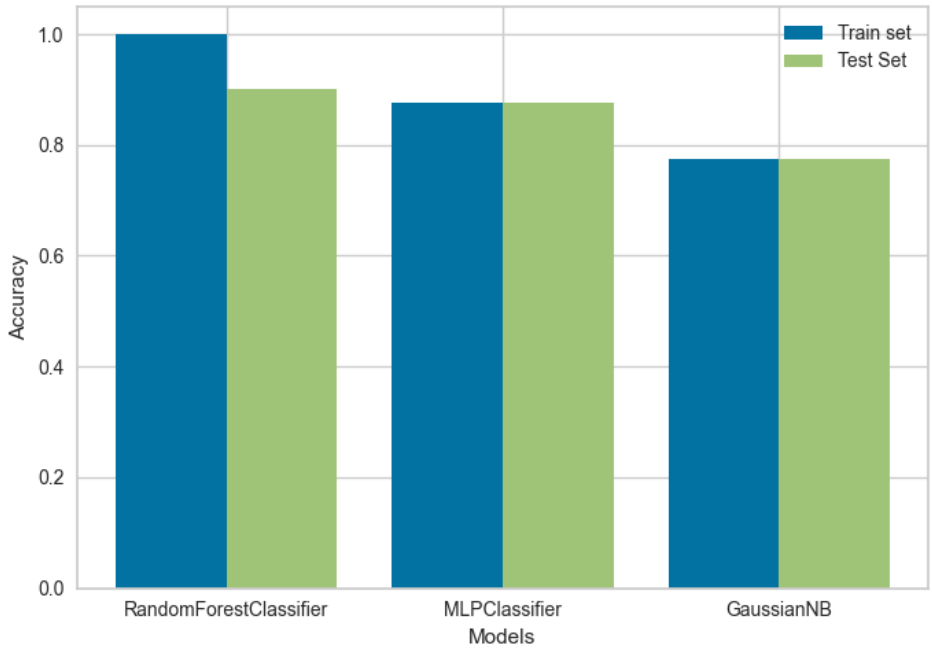
Από τους 3 ταξινομητές ο randomforest έχει την μεγαλύτερη ακρίβεια, μετά το νευρωνικό δίκτυο και τέλος το Bayesian δίκτυο. Επίσης από τους 3 ταξινομητές το Bayesian δίκτυο έχει το μικρότερο χρόνο εκτέλεσης με διαφορά από τους άλλους 2.

Ο randomforest στο στάδιο εκπαίδευσης και δοκιμής έχει τη μεγαλύτερη, αλλά οι δύο τιμές μεταξύ τους έχουν διαφορά. Επομένως γίνεται υπερεκπαίδευση στο μοντέλο και για αυτό όταν του δίνεται το σύνολο δοκιμής δεν πετυχαίνει την ίδια ακρίβεια αφού τα δεδομένα είναι διαφορετικά από αυτά με τα οποία εκπαιδεύτηκε.

Το νευρωνικό δίκτυο έχει την αμέσως επόμενη μεγαλύτερη ακρίβεια και οι τιμές στην ακρίβεια μεταξύ εκπαίδευσης και testing είναι σχεδόν ίδιες. Άρα το μοντέλο πετυχαίνει γενίκευση.

Το bayesian δίκτυο πετυχαίνει τη μικρότερη ακρίβεια από τους άλλους ταξινομητές. Παρόλα αυτά παρουσιάζει πολύ καλή γενίκευση όπως και το νευρωνικό δίκτυο και δίνει αποτέλεσμα πολύ πιο γρήγορα.

Η χαμηλή ακρίβεια που δίνει αυτός ο ταξινομητής προκύπτει από την κατανομή των δεδομένων, ο ταξινομητής υποθέτει ότι τα δεδομένα ακολουθούν κανονική κατανομή. Αλλά όπως αναφέρθηκε νωρίτερα η κατανομή των δεδομένων μοιάζει με κανονική διότι υπάρχουν τιμές οι οποίες δεν υπακούν στην κατανομή και έτσι μειώνουν την ακρίβεια του μοντέλου.



Ερώτημα 3

```
*** KMEANS ***
Silhouette Score: 0.6198553100092677
Labels: [4 4 1 0 0 0 0 4 0 0 2 2 0 2 2 1 1 3 3 1 1 1]
Number of clusters: 5

*** MEANSHIFT ***
Silhouette Score: 0.6192423290019132
Labels: [3 3 1 0 0 0 0 3 0 0 2 2 0 2 2 1 1 4 1 1 1 1]
Number of clusters: 5

*** DBSCAN ***
Silhouette Score: 0.5814032443155533
Labels: [ 0  0  1  2  2  2  2  0  2  2  2  3  2  3  2  1  1 -1  1  1  1  1]
Number of clusters: 5
```

Στην συσταδοποίηση χρησιμοποιήθηκαν οι αλγόριθμοι kmeans, meanshift, dbscan. Η απόδοσή τους αξιολογείται από την μετρική silhouette που βρίσκει πόσο μοιάζει κάθε σημείο στο cluster που τοποθετήθηκε σε σχέση με τα άλλα clusters.

Το καλύτερο score το πετυχαίνει ο αλγόριθμος kmeans, μετά ο meanshift και τέλος ο dbscan. Οι συστάδες που δημιουργούνται είναι οι ίδιες για κάθε αλγόριθμο.

Η ομαδοποίηση στους 2 πρώτους αλγορίθμους είναι σχεδόν ίδια, μόνο ένα σημείο διαφέρει, ενώ στον dbscan λόγω της μορφής των δεδομένων που δεν έχουν πυκνούς σχηματισμούς υπάρχει outlier.