ΕΡΓΑΣΙΑ ΣΤΑΤΙΣΤΙΚΗ ΜΑΘΗΣΗ

Η εργασία περιλαμβάνει την ανάλυση δεδομένων για την ποιότητα του λευκού κρασιού (winequality-white.csv). Συγκεκριμένα:

1. Θα πρέπει να εκπαιδεύσετε μοντέλα στατιστικής/μηχανικής μάθησης για να προβλέπουν την ποιότητα του κρασιού. Η ποιότητα θα καθορίζεται με βάση το χαρακτηριστικό *quality* με τον εξής τρόπο:
    - **Κακή**: βαθμολογία κάτω ή ίση του 4
    - **Μέτρια**: βαθμολογία 5 ή 6
    - **Καλή**: βαθμολογία μεγαλύτερη ή ίση του 7
2. Θα πρέπει να εκπαιδεύσετε μοντέλα στατιστικής/μηχανικής μάθησης για να προβλέπουν την ποιότητα του κρασιού όπως αυτή δίνεται από τα δεδομένα (*quality*), δηλ. χωρίς την ταξινόμηση στην 1.
3. Στα δύο παραπάνω θα πρέπει να ελέγξετε για την παρουσία τυχών ανώμαλων σημείων.
4. Μπορείτε να συμπεράνετε ποια χαρακτηριστικά έχουν τη μεγαλύτερη επιρροή στην ποιότητα του κρασιού σύμφωνα με τα μοντέλα στατιστικής/μηχανικής μάθησης;

Θα πρέπει να γράψετε σε ένα αρχείο docx/latex τα αποτελέσματα σας και στη συνέχεια να το μετατρέψετε σε pdf, στο όποιο θα βάλετε ως όνομα τα στοιχεία σας, π.χ. *Λουμπόνιας_Κώστας_ΑΕΜ.pdf*. Ακριβώς τα ίδια στοιχεία θα φέρει και ο φάκελος με το κώδικα σας*. Στο αρχείο pdf θα πρέπει να περιγράφετε-δικαιολογείτε τη διαδικασία που ακολουθήσατε και την ερμηνεία των αποτελεσμάτων. Μπορείτε να χρησιμοποιήσετε και screenshots από το κώδικα. Οποιαδήποτε μέθοδο χρησιμοποιήσετε θα πρέπει να γνωρίζετε την λειτουργίας της. **Η παρουσίαση των κύριων ευρημάτων και ερμηνεία των αποτελεσμάτων θα γίνει προφορικά εντός 5 λεπτών.**

*Στο φάκελο με το κώδικα **μη** συμπεριλάβετε το αρχείο csv με τα δεδομένα.

Οδηγίες:

- Η είσοδος των δεδομένων θα γίνει βάση των εντολών:
  import pandas as pd
  df = pd.read_csv(' winequality-white.csv', delimiter=';')
- Για την είσοδο στα μοντέλα σας μπορείτε να χρησιμοποιήσετε οποιαδήποτε χαρακτηριστικό από τη βάση (winequality-white.csv) με όποιο τρόπο θέλετε (γραμμικούς και μη συνδυασμούς) εκτός από το χαρακτηριστικό *quality.*
- Να χρησιμοποιήσετε τουλάχιστον 3 διαφορετικά μοντέλα για το 1^{ο} κομμάτι της εργασίας.
- Να χρησιμοποιήσετε τουλάχιστον 2 διαφορετικά μοντέλα για το 2^{ο} κομμάτι της εργασίας.
- Σε περίπτωση προβλήματος ταξινόμησης να χρησιμοποιήσετε ως μετρική το

  f1_score(πραγματική τιμή, προβλεπόμενη τιμή, average = 'micro')

  ενώ σε περίπτωση παλινδρόμησης την **ρίζα του μέσου τετραγωνικού σφάλματος** (rmse).

- Ως training, validation και testing συνόλου δεδομένων να χρησιμοποιήσετε το 70%, 10% και 20% της αρχικής βάσης. Προτιμήστε τον τρόπο που έχει χρησιμοποιηθεί κατά την διάρκεια των μαθημάτων.
- Η τελική αξιολόγηση των μοντέλων πρέπει να γίνει μόνο στα δεδομένα **testing**.
- Για την καλύτερη αξιολόγηση των αποτελεσμάτων να εκπαιδευτούν όλα τα μοντέλα από 10 φορές. Κάθε φορά να υπάρχει και διαφορετικός διαχωρισμός (χωρίς συγκεκριμένο seed ή random_state) σε training, validation και testing, δηλ. **train_test_split(x, y, test_size=0.2).**
  Στο τέλος να υπολογιστεί η μέση τιμή και τυπική απόκλιση των f1_score / rmse (για το testing dataset) για τις 10 επαναλήψεις.

Για τα Support Vector Machine (SVM), Decision trees μπορείτε να χρησιμοποιήσετε:

A. SVM

```
from sklearn.svm import SVC

model = SVC(decision_function_shape='ovo')

model.fit(x_train, y_train)

y_pred = model.predict(x_test)
```

B. Decision trees

```
from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier()

model.fit(x_train, y_train)

y_pred = model.predict(x_test)
```

Από το sklearn επίσης μπορείτε να χρησιμοποιήσετε το linear και logistic regression επίσης.

Προσοχή: Το y_train (οι κλάσεις) δεν πρέπει να είναι σε one-hot-vector μορφή, αλλά σε απλή μορφή, π.χ. y_train = [0,1,1,2,0,2,…].

Σε περίπτωση που χρησιμοποιήσετε τη **Principal Component Analysis (PCA),** να κάνετε χρήση της παρακάτω βιβλιοθήκη:

```
from sklearn.decomposition import PCA

pca = PCA(n_components=5)            # for 5 components

X_pca = pca.fit_transform(X)
```

ή αντί για pca = PCA(n_components=12), το

```
pca = PCA(n_components=0.95)         #  for retain 95% of total variances
```