ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΙ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΑΘΗΜΑ: ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

ΚΑΘΗΓΗΤΕΣ : ΚΩΣΤΑΣ ΔΙΑΜΑΝΤΑΡΑΣ, ΚΩΣΤΑΣ ΓΟΥΛΙΑΝΑΣ

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 8

ΕΠΙΛΟΓΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΜΕ PRINCIPAL COMPONENT ANALYSIS (PCA)

Σκοπός της άσκησης: η χρήση της **Ανάλυσης Κυρίων Συνιστωσών** (**Principal Component Analysis - PCA**) για την συμπίεση των δεδομένων εισόδου και η μελέτη της επίπτωσης αυτής της ανάλυσης στην επίδοση ενός μοντέλου Μηχανικής Μάθησης. Ως παράδειγμα θα χρησιμοποιηθεί ένα μοντέλο *Naïve Bayes* με *Γκαουσσιανή* συνάρτηση πυκνότητας πιθανότητας.

Βήματα υλοποίησης:

1. Χρησιμοποιήστε το σύνολο δεδομένων `mnist_49.npz` το οποίο δίνεται. Το αρχείο είναι τύπου numpy zip και διαβάζεται με τη χρήση της συνάρτησης

   `data = np.load(<file_name>)`

   Η δομή δεδομένων data είναι ουσιαστικά ένα dictionary που περιέχει δύο πίνακες:

   (α) τον πίνακα `x` των δεδομένων με 11791 πρότυπα διάστασης 784. Κάθε γραμμή του πίνακα `x` είναι μια εικόνα διάστασης 28×28 με 784 pixels (784=28×28). Οι εικόνες περιέχουν χειρόγραφα ψηφία και συγκεκριμένα τα ψηφία "4" και "9". Παίρνουμε τον πίνακα x από τη δομή data με την εντολή

   `x = data['x']`

   (β) τον πίνακα `t` των στόχων με 11791 δυαδικούς αριθμούς 0/1. Αν ο στόχος είναι 0 τότε το πρότυπο είναι το ψηφίο "4" αλλιώς είναι το ψηφίο "9". Παίρνουμε τον πίνακα `t` από τη δομή `data` με την εντολή

   `t = data['t']`

2. Θα γίνει ταξινόμηση των προτύπων με την μέθοδο Naïve Bayes/Gaussian χρησιμοποιώντας Cross-Validation για K=10 folds διαιρώντας τα πρότυπα σε train και test set με τη μέθοδο `train_test_split()`. Η μέθοδος Naïve Bayes/Gaussian καλείται χρησιμοποιώντας την κλάση `GaussianNB` του Scikit-Learn.

   - Δημιουργήστε ένα μοντέλο `GaussianNB()` χωρίς καμία παράμετρο
   - Εκπαιδεύστε το μοντέλο με την συνάρτηση `fit()` χρησιμοποιώντας φυσικά το train set.
   - Αξιολογήστε το μοντέλο με τη συνάρτηση `score()` η οποία υπολογίζει το accuracy. Κάνετε δύο αξιολογήσεις του μοντέλου χρησιμοποιώντας ξεχωριστά το train set και ξεχωριστά το test set. Τυπώστε στην οθόνη το μέσο accuracy από όλα τα folds τόσο για το train set όσο και για το test set.

3. Εφαρμόστε τη μέθοδο PCA για εξαγωγή των πιο σημαντικών χαρακτηριστικών για κάθε εικόνα εισόδου. Το πλήθος των χαρακτηριστικών θα το ονομάσετε `num_components` (number of components). Θα τρέξετε ένα loop για διαφορετικές τιμές του `num_components`, πχ για τις τιμές [1, 2, 5, 10, 20, 30, 40, 50, 100, 200]

   `Για κάθε` *num_components*

- Εφαρμόστε PCA για πλήθος χαρακτηριστικών = `num_components` ως εξής:
  - Δημιουργήστε ένα μοντέλο PCA καλώντας την κλάση PCA

    ```
    pca = PCA(n_components = num_components)
    ```

  - Δημιουργήστε τον πίνακα συμπιεσμένων δεδομένων `x_pca` από τα αρχικά δεδομένα `x` με τη μέθοδο `fit_transform()`. Ο πίνακας `x_pca` πρέπει να έχει διάσταση 11791× `num_components`.
- Κάνετε ταξινόμηση των προτύπων `x_pca` με το μοντέλο Naïve Bayes/Gaussian χρησιμοποιώντας Cross-Validation για K=10 folds διαιρώντας τα πρότυπα σε train και test set με τη μέθοδο `train_test_split()`.
- Αξιολογήστε το μοντέλο με τη συνάρτηση `score()`. Κάνετε δύο αξιολογήσεις του μοντέλου χρησιμοποιώντας ξεχωριστά το train set και ξεχωριστά το test set. Σώστε το μέσο accuracy για το train set και το μέσο accuracy για το test set σε διαφορετικά array (ή λίστες ή dictionaries). Πχ.
  - acc_train[] = array με το μέσο accuracy στο train set για όλα τα `num_components`
  - acc_test[] = array με το μέσο accuracy στο test set για όλα τα `num_components`

```
end    #for
```

4. Μετά το τέλος του loop δημιουργήστε ένα γράφημα όπου θα δείχνει το μέσο accuracy για το train set και το μέσο accuracy για το test set σαν συνάρτηση του `num_components`. Δηλαδή θα δείξετε δύο καμπύλες όπου ο άξονας y θα δείχνει το μέσο accuracy και ο άξονας x θα έχει το πλήθος των components.
5. Τι παρατηρείτε; Εξηγήστε.