ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΙ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΑΘΗΜΑ: ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

ΚΑΘΗΓΗΤΕΣ : ΚΩΣΤΑΣ ΔΙΑΜΑΝΤΑΡΑΣ, ΚΩΣΤΑΣ ΓΟΥΛΙΑΝΑΣ

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 1

ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ – ΔΙΑΧΩΡΙΣΜΟΣ CROSS-VALIDATION

Σκοπός της άσκησης: Η ανάγνωση των δεδομένων από ένα αρχείο και η κατανόηση και η υλοποίηση της μεθόδου διασταύρωσης (Cross-Validation). Σύμφωνα με τη μέθοδο αυτή τα δεδομένα που διαθέτουμε χωρίζονται σε δύο υποσύνολα:

1. Το υποσύνολο εκπαίδευσης (train set) το οποίο θα χρησιμοποιηθεί για την εκπαίδευση του μοντέλου μηχανικής μάθησης.
2. Το υποσύνολο ελέγχου (test set) το οποίο θα χρησιμοποιηθεί για τον έλεγχο της ικανότητας γενίκευσης του μοντέλου.

Εκτελείται μια σειρά από πειράματα που καλούνται "*folds*". Σε κάθε fold:

- δημιουργούνται διαφορετικά train set και test set χωρίζοντας τα δεδομένα με τυχαίο τρόπο
- το μοντέλο εκπαιδεύεται χρησιμοποιώντας το αντίστοιχο train set
- υπολογίζεται το σφάλμα (ή η επιτυχία) του αλγορίθμου στο test set. Ανάλογα με το πρόβλημα το κριτήριο επίδοσης μπορεί να είναι διαφορετικό.

Αφού εκτελεστούν K folds συλλέγεται ο μέσος όρος της επίδοσης του αλγορίθμου στα K folds. Αυτός ο μέσος όρος αποτελεί την εκτίμησή μας για την επίδοση του μοντέλου σε άγνωστα δεδομένα (ικανότητα γενίκευσης).

Βήματα υλοποίησης:

1. Κατεβάστε το σύνολο δεδομένων (data set) IRIS dataset από την παρακάτω ιστοσελίδα:

   http://archive.ics.uci.edu/ml/machine-learning-databases/iris/

   Αυτό είναι ίσως το πιο γνωστό σύνολο δεδομένων που χρησιμοποιείται στη βιβλιογραφία της αναγνώρισης προτύπων. Αφορά την αναγνώριση του τύπου λουλουδιού του γένους "ίρις". Περιέχει 3 κλάσεις λουλουδιών: "Iris-setosa","Iris-versicolor" και "Iris-virginica", με 50 δείγματα από κάθε μια κλάση (σύνολο 150 δείγματα).

   Το data set αποτελείται από δύο αρχεία:

   i. `iris.data` : περιέχει τα δεδομένα. Αποτελείται από 150 γραμμές, όπου κάθε γραμμή αντιστοιχεί σε ένα δείγμα. Κάθε δείγμα περιέχει 4 χαρακτηριστικά συν τον τύπο του λουλουδιού σε μορφή text-string. χωρισμένα με κόμματα.
   ii. `iris.names` : ενημερωτικό κείμενο το οποίο περιέχει την περιγραφή των δεδομένων.

2. Διαβάστε το αρχείο δεδομένων `iris.data` στο MATLAB. Χρησιμοποιήστε τις παρακάτω εντολές
   - `fopen()` : ανοίγει ένα αρχείο για ανάγνωση.

```
% Παράδειγμα:
fid = fopen('όνομα αρχείου', 'r');
```

- `textscan()` : διαβάζει ένα ολόκληρο αρχείο κειμένου (όσες γραμμές κι αν έχει) αρκεί κάθε γραμμή να περιέχει n στοιχεία χωρισμένα με κάποιο διαχωριστικό χαρακτήρα. Για παράδειγμα, η παρακάτω εντολή διαβάζει το περιεχόμενο ενός ολόκληρου αρχείου όπου κάθε γραμμή περιέχει 3 ακέραιους αριθμούς (format '%d') χωρισμένους με κόμμα.

```
% Παράδειγμα:
data = textscan(fid,'%d %d %d', 'Delimiter',',');
```

  Επιστρέφει το cell array `data` το οποίο περιέχει 3 στοιχεία:

  - data{1} : array n×1 που περιέχει τους πρώτους αριθμούς από κάθε γραμμή
  - data{2} : array n×1 που περιέχει τους δεύτερους αριθμούς από κάθε γραμμή
  - data{3} : array n×1 που περιέχει τους τρίτους αριθμούς από κάθε γραμμή

- `fclose()` : κλείνει το αρχείο που άνοιξε η `fopen()`.

```
% Παράδειγμα:
fclose(fid);
```

Cell arrays: είναι arrays τα οποία μπορούν αν περιέχουν ετερόκλητα στοιχεία, πχ ακεραίους, text-strings, πίνακες, κλπ. Για παράδειγμα η εντολή:

```
>>   z = {'one', 'two', 'three', 1, 2, [10,20,30]}

z =

  'one'    'two'    'three'    [1]    [2]    [1x3 double]
```

δημιουργεί ένα cell array με 6 στοιχεία:

`z{1}` είναι `'one'` (string)

`z{2}` είναι `'two'` (string)

`z{3}` είναι `'three'` (string)

`z{4}` είναι `1` (αριθμός)

`z{5}` είναι `2` (αριθμός)

`z{6}` είναι `[10,20,30]` (array 1x3)

Τα στοιχεία ενός cell array προσπελαύνονται χρησιμοποιώντας τις αγκύλες `{}`.

Στο παραπάνω παράδειγμα, αφού το `z{6}` είναι array, έχουμε:

`z{6}(1)` είναι `10`

`z{6}(2)` είναι `20`

`z{6}(3)` είναι `30`

3. Υπολογίστε τα εξής:

   - Πλήθος των attributes: `NumberOfAttributes` (στη συγκεκριμένη περίπτωση = 5) χρησιμοποιώντας την συνάρτηση `length(data)`.
   - Πλήθος των δειγμάτων: `NumberOfPatterns` (στη συγκεκριμένη περίπτωση = 150) χρησιμοποιώντας τη συνάρτηση `length(data{1})`.
   - Αρχικοποιήστε τον πίνακα των προτύπων `x()` σε μηδέν. Πρέπει να έχει διαστάσεις `(NumberOfAttributes-1) × NumberOfPatterns`
   - Αρχικοποιήστε το διάνυσμα στόχων `t()` σε μηδέν. Πρέπει να έχει διαστάσεις `1 × NumberOfPatterns`

Θα χρησιμοποιήσετε τη συνάρτηση `zeros()`.

4. Χρησιμοποιώντας loop, γεμίστε τους πίνακες `x(), t()` ως εξής:

   - για τα 4 πρώτα attributes `i=1,2,3,4,` και για όλα τα `pattern`:

     ```
     x(i,pattern) = data{i}(pattern);
     ```

   - για κάθε πρότυπο `pattern`, το 5ο attribute (`data{NumberOfAttributes}(pattern)`) είναι το όνομα της κλάσης που ανήκει το πρότυπο (τύπου text string). Ο στόχος `t(pattern)` για το πρότυπο αυτό θα πρέπει να είναι:

     `t(pattern) = 1;` αν το 5ο attribute είναι `'Iris-versicolor'`

     `t(pattern) = 0;` σε διαφορετική περίπτωση

   Θα χρησιμοποιήσετε τις συναρτήσεις:

   - `char()` για να μετατρέψετε το cell `data{NumberOfAttributes}(pattern)` σε text string.
   - `strcmp()` για να συγκρίνετε strings.

5. Δοκιμή της μεθόδου `crossvalind('Kfold',...)`

   Τεμαχίστε τα δεδομένα σε 9 cross-validation folds (K=9) χρησιμοποιώντας τη συνάρτηση `crossvalind()` με την παράμετρο `'Kfold'`.

   Σύμφωνα με τη μέθοδο αυτή, επιστρέφονται οι δείκτες των folds μέσα στα οποία κάθε πρότυπο συμμετέχει στο test set.

```
        % Παράδειγμα:
>> indices = crossvalind('Kfold', 20, 5)
indices =
     3
     5
     5
     4
     4
     3
     4
     2
     5
     1
     3
     5
     2
     1
     4
     1
     3
     2
     2
     1
```

Το πρότυπο 1 ανήκει στο test set του fold **3**
Το πρότυπο 2 ανήκει στο test set του fold **5**
Το πρότυπο 3 ανήκει στο test set του fold **5**
Το πρότυπο 4 ανήκει στο test set του fold **4**

...κλπ

Πλήθος folds
Συνολικό πλήθος προτύπων

Η μέθοδος φροντίζει έτσι ώστε όλα τα πρότυπα να συμμετέχουν σε κάποιο test set. Επίσης όλα τα test sets περιέχουν κατά το δυνατόν το ίδιο πλήθος προτύπων. Στο συγκεκριμένο παράδειγμα,

Στο test set…	Ανήκουν τα πρότυπα με αύξ. αριθμό:
…του fold 1	10, 14, 16, 20
…του fold 2	8, 13, 18, 19
…του fold 3	1, 6, 11, 17
…του fold 4	4, 5, 7, 15
…του fold 5	2, 3, 9, 12

βλέπουμε ότι όλα τα test sets περιέχουν ίσο πλήθος προτύπων (4 πρότυπα).

Θα πρέπει να κάνετε τα εξής:

Για κάθε *fold*

- Βρείτε το σύνολο των δεικτών που ανήκουν στο test set για το συγκεκριμένο fold. Ονομάστε το σύνολο αυτό `testidx`
- Βρείτε το σύνολο των δεικτών που ανήκουν στο train set για το συγκεκριμένο fold. Ονομάστε το σύνολο αυτό `trainidx`
- Με τη συνάρτηση `fprintf()` τυπώστε ένα μήνυμα πόσα πρότυπα ανήκουν στο train set και πόσα στο test set για το συγκεκριμένο fold
- Δημιουργήστε τον πίνακα `xtrain()` επιλέγοντας από τον πίνακα `x()` τις στήλες που ανήκουν στο `trainidx`
- Δημιουργήστε το διάνυσμα `ttrain()` επιλέγοντας από το διάνυσμα στόχων `t()` τα στοιχεία που ανήκουν στο `trainidx`
- Δημιουργήστε τον πίνακα `xtest()` επιλέγοντας από τον πίνακα `x()` τις στήλες που ανήκουν στο `testidx`
- Δημιουργήστε το διάνυσμα `ttest()` επιλέγοντας από το διάνυσμα στόχων `t()` τα στοιχεία που ανήκουν στο `testidx`
- Σχεδιάστε με `plot` τα διανύσματα `xtrain(1,:)` → άξονας x, `xtrain(3,:)` → άξονας y, χρησιμοποιώντας τελείες με μπλε χρώμα και τα διανύσματα `xtest(1,:)` → άξονας x, `xtest(3,:)` → άξονας y, χρησιμοποιώντας τελείες με κόκκινο χρώμα
- Χρησιμοποιήστε την εντολή `subplot` έτσι ώστε όλα τα γραφήματα να εμφανιστούν στο Figure 1.

end

Θα χρησιμοποιήσετε τις συναρτήσεις:

- `find()`: βρίσκει ποιοι δείκτες σε ένα array ικανοποιούν μια συνθήκη

```
% Παράδειγμα:
>> find([1,3,5,4,2,3,2,4,3] == 3)
ans =
     2     6     9
```

- `figure()`: δηλώνει ποιο θα είναι το επόμενο παράθυρο τύπου figure

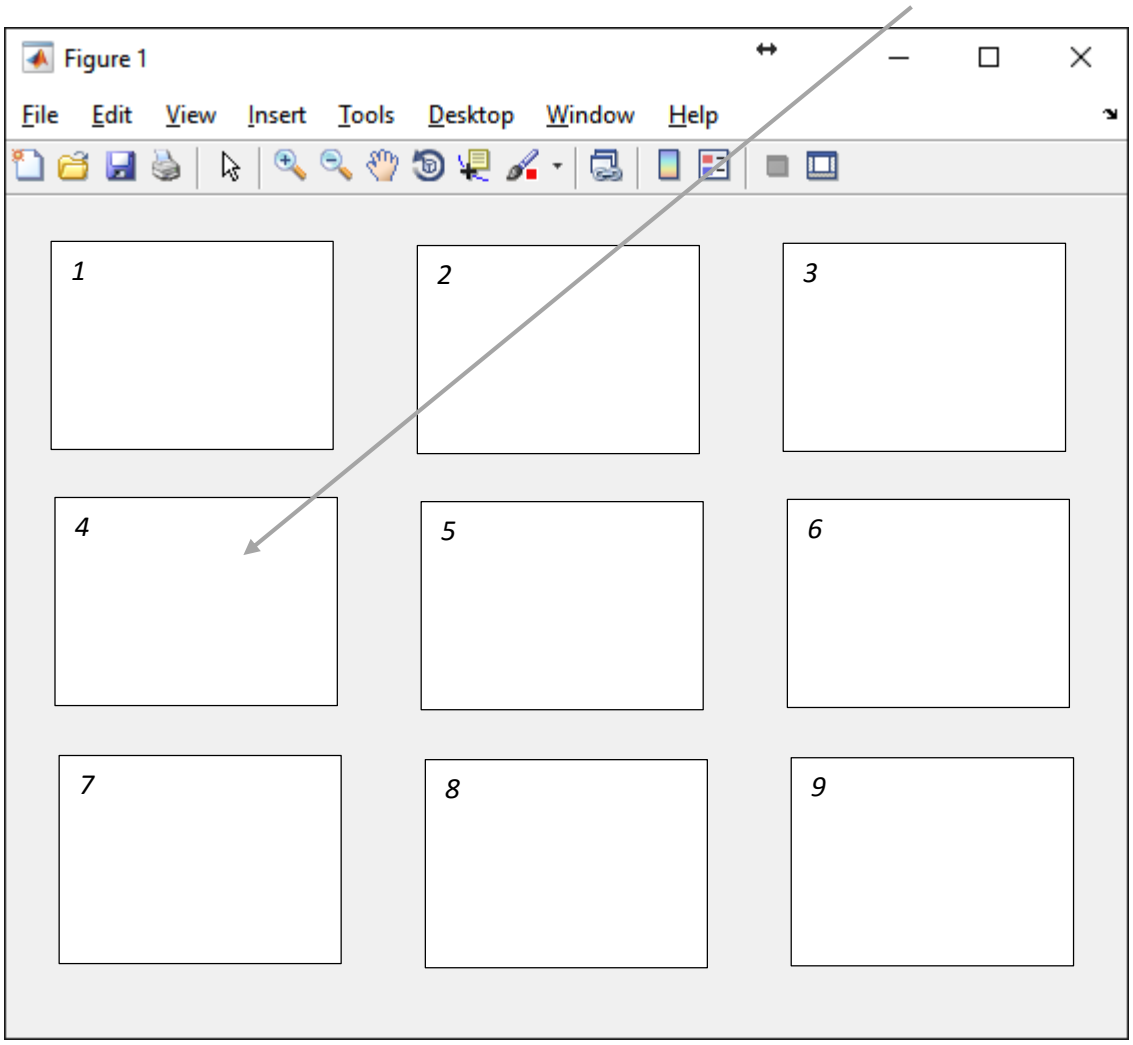
```
% Παράδειγμα:
>> figure(3)

% Ανοίγει το παράθυρο figure Αρ.3
% Αν το παράθυρο είναι ήδη ανοιχτό το φέρνει μπροστά
```

```
% Κάθε επόμενο plot θα εμφανιστεί εκεί
```

- `subplot()`: δηλώνει σε ποιο υπο-plot μέσα σε ένα figure θα γίνει το επόμενο plot

```
% Παράδειγμα:
>> subplot(3, 3, 4)
% Στο current figure δημιούργησε ένα grid 3x3 με υπο-plots
% και αρίθμησε τα όπως στο παρακάτω σχήμα.
% Το επόμενο plot θα εμφανιστεί μέσα στο υπο-plot 4.
```



- `plot()`: η βασική εντολή εμφάνισης γραφημάτων (βλ. Matlab help)
- `hold on, hold off`: διατηρεί ανέπαφο (`hold on`) το υπάρχον γράφημα ώστε το επόμενο plot στο ίδιο figure να πέσει «από πάνω του». Διαφορετικά (`hold off`) το επόμενο plot πάνω στο ίδιο figure σβήνει το προηγούμενο γράφημα.

6. Δοκιμή της μεθόδου `crossvalind('LeaveMOut',...)`

Τεμαχίστε τα δεδομένα σε 9 cross-validation folds (K=9) χρησιμοποιώντας τη συνάρτηση `crossvalind()` με την παράμετρο `'LeaveMOut'`.

Η μέθοδος αυτή δέχεται σαν παραμέτρους

- το συνολικό πλήθος των προτύπων
- το πλήθος των προτύπων στο test set για το συγκεκριμένο fold

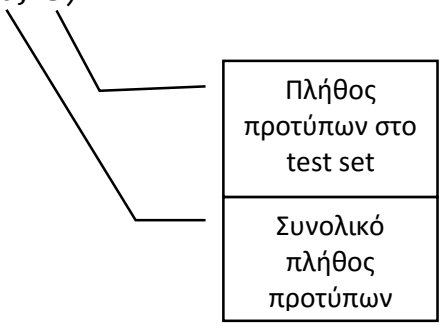
και επιστρέφει δύο arrays:

- ένα boolean array που δείχνει ποια πρότυπα ανήκουν στο train set (0=false, 1=true)
- ένα boolean array που δείχνει ποια πρότυπα ανήκουν στο test set (0=false, 1=true)

Τα δύο arrays δημιουργούνται με τυχαίο τρόπο. Την επόμενη φορά που θα ξανατρέξετε τη συνάρτηση θα βγάλει άλλο αποτέλεσμα.

```
        % Παράδειγμα:
>> [trainidx, testidx] = crossvalind('LeaveMOut', 10, 3)
trainidx =
     0
     1
     0
     0
     1
     1
     1
     1
     1
     1

testidx =
     1
     0
     1
     1
     0
     0
     0
     0
     0
     0
```



Θα πρέπει να κάνετε τα εξής:

Για κάθε *fold*

- Δημιουργήστε τα boolean arrays `trainidx, testidx` με τη μέθοδο `crossvalind()`
- Με τη συνάρτηση `fprintf()` τυπώστε ένα μήνυμα πόσα πρότυπα ανήκουν στο train set και πόσα στο test set για το συγκεκριμένο fold
- Δημιουργήστε τον πίνακα `xtrain()` επιλέγοντας από τον πίνακα `x()` τις στήλες που ανήκουν στο `trainidx`
- Δημιουργήστε το διάνυσμα `ttrain()` επιλέγοντας από το διάνυσμα στόχων `t()` τα στοιχεία που ανήκουν στο `trainidx`
- Δημιουργήστε τον πίνακα `xtest()` επιλέγοντας από τον πίνακα `x()` τις στήλες που ανήκουν στο `testidx`

- Δημιουργήστε το διάνυσμα `ttest()` επιλέγοντας από το διάνυσμα στόχων `t()` τα στοιχεία που ανήκουν στο `testidx`
- Σχεδιάστε με `plot` τα γραφήματα:
  - `xtrain(1,:)` → άξονας x, `xtrain(3,:)` → άξονας y, χρησιμοποιώντας τελείες με μπλε χρώμα
  - `xtest(1,:)` → άξονας x, `xtest(3,:)` → άξονας y, χρησιμοποιώντας τελείες με κόκκινο χρώμα
- Χρησιμοποιήστε την εντολή `subplot` έτσι ώστε όλα τα γραφήματα να εμφανιστούν στο Figure 2.

end