ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΙ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΑΘΗΜΑ: ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

ΚΑΘΗΓΗΤΕΣ : ΚΩΣΤΑΣ ΔΙΑΜΑΝΤΑΡΑΣ, ΚΩΣΤΑΣ ΓΟΥΛΙΑΝΑΣ

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 2

ΓΡΑΜΜΙΚΟΣ ΤΑΞΙΝΟΜΗΤΗΣ

Σκοπός της άσκησης: Η εκτίμηση της επίδοσης ενός <u>γραμμικού ταξινομητή</u> δύο κλάσεων σε άγνωστα δεδομένα. Θα γίνει χρήση της μεθόδου διασταύρωσης (Cross-Validation).

Θα χρησιμοποιηθούν τα εξής κριτήρια επίδοσης:

1. Ακρίβεια (accuracy)
2. Ευστοχία (precision)
3. Ανάκληση (recall)
4. F-Measure
5. Ευαισθησία (Sensitivity)
6. Προσδιοριστικότητα (Specificity)

Δείτε τη σημασία αυτών των κριτηρίων στο επισυναπτόμενο κείμενο <u>Κριτήρια επίδοσης ταξινομητών.pdf</u>

Βήματα υλοποίησης:

1. Χρησιμοποιήστε το σύνολο δεδομένων IRIS από το προηγούμενο εργαστήριο, καθώς και τον κώδικα από το εργαστήριο αυτό. Θυμίζουμε ότι τα πρότυπα χωρίστηκαν σε δύο κλάσεις ως εξής:
   - Κλάση 0 (στόχος t=0): αποτελείται από τα πρότυπα των κατηγοριών "<u>Iris-setosa</u>"+"<u>Iris-virginica</u>",
   - Κλάση 1 (στόχος t=1): αποτελείται από τα πρότυπα της κατηγορίας "<u>Iris-versicolor</u>"

2. Θα εφαρμοστεί <u>μόνο</u> η μέθοδος `crossvalind('Kfold',...)` για K=9 folds. Στο αντίστοιχο loop θα πρέπει να κάνετε τα εξής:

   `Για κάθε` *fold*

   - Έχετε ήδη δημιουργήσει τους πίνακες `xtrain(), xtest()` καθώς και τα διανύσματα στόχων `ttrain(), ttest()`
   - Βρείτε το πλήθος των προτύπων στο train set (P_{train}) και στο test set (P_{test}) για παράδειγμα χρησιμοποιώντας τη συνάρτηση `length()`.
   - Επαυξήστε τον πίνακα των προτύπων προσθέτοντας σε κάθε πρότυπο τον αριθμό 1, δηλαδή προσθέστε μια γραμμή με 1 στους πίνακες `xtrain(), xtest()`.
   - Μετατρέψτε τα διανύσματα στόχων `ttrain(), ttest()` έτσι ώστε
     - Αν `ttrain(pattern) == 1` → `ttrain1(pattern) = 1`
     - Αν `ttrain(pattern) == 0` → `ttrain1(pattern) = -1`
     - Αν `ttest(pattern) == 1` → `ttest1(pattern) = 1`
     - Αν `ttest(pattern) == 0` → `ttest1(pattern) = -1`

- Βρείτε το διάνυσμα βαρών $\tilde{\mathbf{w}}$ του γραμμικού ταξινομητή

$$y = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}}$$

$$\tilde{\mathbf{w}} = \begin{bmatrix} \mathbf{w} \\ w_0 \end{bmatrix}, \qquad \tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$$

όπου $\mathbf{w}$ είναι το διάνυσμα βαρών, w_0 είναι η πόλωση, και $\mathbf{x}$ είναι το πρότυπο εισόδου.

- Υπολογίστε το βέλτιστο διάνυσμα βαρών $\tilde{\mathbf{w}}$. Σύμφωνα με τη θεωρία των γραμμικών ταξινομητών, το βέλτιστο διάνυσμα είναι

$$\tilde{\mathbf{w}}^T = \mathbf{t}_{train}^T \tilde{\mathbf{X}}_{train}^+$$

όπου

$\mathbf{t}_{train}^T = [t_1 \quad t_2 \quad \cdots \quad t_P]$ είναι το διάνυσμα των τροποποιημένων στόχων (-1/1),

$\tilde{\mathbf{X}}_{train} = [\tilde{\mathbf{x}}_1 \quad \tilde{\mathbf{x}}_2 \quad \cdots \quad \tilde{\mathbf{x}}_P] = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_P \\ 1 & 1 & \cdots & 1 \end{bmatrix}$ είναι ο πίνακας των επαυξημένων προτύπων του train set,

$\mathbf{X}^+$ είναι ο τελεστής του ψευδο-αντίστροφου του πίνακα $\mathbf{X}$. Στο MATLAB ο ψευδο-αντίστροφος υλοποιείται με τη συνάρτηση `pinv()`.

- Υπολογίστε την έξοδο του ταξινομητή για όλα τα πρότυπα του test set:

$$\mathbf{y}_{test}^T = \tilde{\mathbf{w}}^T \tilde{\mathbf{X}}_{test}$$

- Υπολογίστε την εκτίμηση που κάνει ο ταξινομητής για την κλάση στην οποία ανήκουν τα πρότυπα του test set:

$$predict_{test}(i) = \begin{cases} 0, & y_{test}(i) < 0 \\ 1, & y_{test}(i) \geq 0 \end{cases}$$

- Υλοποιήστε τη συνάρτηση `evaluate()` με τρεις εισόδους και μια έξοδο ως εξής:

```
function value = evaluate( t, predict, criterion )
% Είσοδος t : διάνυσμα με τους πραγματικούς στόχους (0/1)
% Είσοδος predict : διάνυσμα με τους εκτιμώμενους στόχους (0/1)
% Είσοδος criterion : text-string με τις εξής πιθανές τιμές:
%   'accuracy'
%   'precision'
%   'recall'
%   'fmeasure'
%   'sensitivity'
%   'specificity'
% Έξοδος value : η τιμή του κριτηρίου που επιλέξαμε.
```

- Πρώτα υπολογίστε τα true-negatives, false-negatives, true-positives, false positives, τα οποία ορίζονται ως εξής:

  (α) **true negatives** (πραγματικά αρνητικά) οι περιπτώσεις όπου:
  το πρότυπο βγήκε *αρνητικό*, δηλ. $y_i = 0$
  και όντως *ανήκει στην κλάση 0*, δηλ. $t_i = 0$

  (β) **false negatives** (εσφαλμένα αρνητικά) οι περιπτώσεις όπου:
  το πρότυπο βγήκε *αρνητικό*, δηλ. $y_i = 0$
  αλλά *ανήκει στην κλάση 1*, δηλ. $t_i = 1$

(γ) **true positives** (πραγματικά θετικά) οι περιπτώσεις όπου:
το πρότυπο βγήκε ***θετικό***, δηλ. $y_\iota = 1$
και όντως *ανήκει στην κλάση 1*, δηλ. $t_i = 1$

(δ) **false positives** (εσφαλμένα θετικά) οι περιπτώσεις όπου:
το πρότυπο βγήκε ***θετικό***, δηλ. $y_\iota = 1$
αλλά *ανήκει στην κλάση 0*, δηλ. $t_i = 0$

Κατόπιν

- Αν επιλεγεί `criterion = 'accuracy'` τότε η συνάρτηση επιστρέφει

$$value = Accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

- Αν επιλεγεί `criterion = 'precision'` τότε η συνάρτηση επιστρέφει

$$value = Precision = \frac{tp}{tp + fp}$$

- Αν επιλεγεί `criterion = 'recall'` τότε η συνάρτηση επιστρέφει

$$value = Recall = \frac{tp}{tp + fn}$$

- Αν επιλεγεί `criterion = 'fmeasure'` τότε η συνάρτηση επιστρέφει

$$value = Fmeasure = \frac{Precision \cdot Recall}{(Precision + Recall)/2}$$

- Αν επιλεγεί `criterion = 'sensitivity'` τότε η συνάρτηση επιστρέφει

$$value = Sensitivity = \frac{tp}{tp + fn}$$

- Αν επιλεγεί `criterion = 'specificity'` τότε η συνάρτηση επιστρέφει

$$value = Specificity = \frac{tn}{tn + fp}$$

- Καλέστε τη συνάρτηση `evaluate()` όσες φορές χρειάζεται έτσι ώστε για το συγκεκριμένο fold να υπολογίσετε το Accuracy, Precision, Recall, F-measure, Sensitivity και Specificity.
- Χρησιμοποιώντας κατάλληλο subplot σε grid 3x3 στο figure(1) τυπώστε το εξής γράφημα:
  - δείξτε με μπλε τελείες τους πραγματικούς στόχους $t_{test}(i)$ για όλα τα πρότυπα του test set
  - δείξτε με κόκκινους κύκλους τους εκτιμώμενους στόχους $predict_{test}(i)$ για όλα τα πρότυπα του test set

```
end    %for
```

Μετά το τέλος του loop υπολογίστε και τυπώστε στην οθόνη τα εξής:

1. τη μέση τιμή του Accuracy για όλα τα folds
2. τη μέση τιμή του Precision για όλα τα folds
3. τη μέση τιμή του Recall για όλα τα folds
4. τη μέση τιμή του F-Measure για όλα τα folds
5. τη μέση τιμή του Sensitivity για όλα τα folds
6. τη μέση τιμή του Specificity για όλα τα folds

Θα χρησιμοποιήσετε τις παρακάτω εντολές ή συναρτήσεις:

- `switch`: έλεγχος πολλαπλών επιλογών

```
% Παράδειγμα:
switch μεταβλητή
    case τιμή1
        κάνε κάτι…;
    case τιμή2
        κάνε κάτι άλλο…;
    . . .
    case τιμήN
        κάνε κάτι άλλο…;
    otherwise
        κάνε κάτι άλλο…;
end
```

- `pinv()`: υπολογισμός ψευδο-αντίστροφου ενός πίνακα

```
% Παράδειγμα:
>> A = [0.3,    0.1,    0.2; ...
      0.5,    1,    0.2]
A =
    0.3000    0.1000    0.2000
    0.5000    1.0000    0.2000
>> A1 = pinv(A)
A1 =
    2.5078   -0.1762
   -1.6684    1.1503
    2.0725   -0.3109
>> A*A1
ans =
    1.0000    0.0000
    0.0000    1.0000
```

- Λογικοί τελεστές με διανύσματα:

```
% Παράδειγμα:
>> z = [0.1,   -0.5,   0,    0.5,   -0.1,    0.3,    0.2];
>> y1 = (z>0)
y1 =
     1     0     0     1     0     1     1
```

```
>> y2 = (z==0)
y2 =
     0     0     1     0     0     0     0
>> y3 = (z<0)
y3 =
     0     1     0     0     1     0     0
```

Στο MATLAB δεν υπάρχουν οι σταθερές *true* και *false*, απλά η τιμή 1 θεωρείται true και η τιμή 0 false. Συνεπώς μπορούμε να κάνουμε λογικές πράξεις μεταξύ binary διανυσμάτων ως εξής:

```
% Παράδειγμα λογικού OR:
>> y1 | y2
ans =
     1     0     1     1     0     1     1
% Παράδειγμα λογικού AND:
>> y3 & [1,1,1,1,0,0,0]
ans =
     0     1     0     0     0     0     0
% Παράδειγμα λογικού NOT:
>> ~y3
ans =
     1     0     1     1     0     1     1
```

- Η συνάρτηση `sum()`: αν το όρισμά της είναι διάνυσμα, επιστρέφει ένα αριθμό που είναι το άθροισμα των στοιχείων του διανύσματος. Αν το όρισμα είναι πίνακας υπολογίζει το άθροισμα των στοιχείων κάθε στήλης και επιστρέφει διάνυσμα.

```
% Παράδειγμα:
>> z = [0.1,   -0.5,   0,    0.5,   -0.1,    0.3,    0.2];
>> y = (z>0)
y =
     1     0     0     1     0     1     1
>> sum(y)
ans =
     4

>> A = [0.3,    0.1,    0.2; ...
0.5,    1,    0.2]
A =
    0.3000    0.1000    0.2000
    0.5000    1.0000    0.2000
>> sum(A)
ans =
    0.8000    1.1000    0.4000
```