

ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΙ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΑΘΗΜΑ: ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

ΚΑΘΗΓΗΤΕΣ : ΚΩΣΤΑΣ ΔΙΑΜΑΝΤΑΡΑΣ, ΚΩΣΤΑΣ ΓΟΥΛΙΑΝΑΣ

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 3

ΜΟΝΤΕΛΟ PERCEPTRON

Σκοπός της άσκησης: Η εκτίμηση της επίδοσης ενός ταξινομητή τύπου Perceptron 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 [Κριτήρια επίδοσης ταξινομητών.pdf](#)

Βήματα υλοποίησης:

1. Χρησιμοποιήστε το σύνολο δεδομένων IRIS από το προηγούμενο εργαστήριο, καθώς και τον κώδικα από το εργαστήριο αυτό.
2. Με τη χρήση της εντολής plot δημιουργήστε τη **γραφική παράσταση** των προτύπων **των 3 κλάσεων** με **διαφορετικό σύμβολο και χρώμα για την κάθε κλάση** χρησιμοποιώντας την 1^η και 3^η στήλη του πίνακα x, ώστε να τα απεικονίσετε στο χώρο των 2 διαστάσεων και εμφανίστε τα στο ίδιο γράφημα, ώστε να πάρετε μια ιδέα για το πώς είναι η διασπορά των προτύπων στο χώρο των 4 διαστάσεων. Μη ξεχνάτε ότι η αρίθμηση ξεκινάει απ' το 0.
3. Επαυξήστε τον πίνακα των προτύπων προσθέτοντας σε κάθε πρότυπο τον αριθμό 1, δηλαδή προσθέστε μια στήλη με 1 στον πίνακα x. Χρησιμοποιήστε τη συνάρτηση `hstack()` του `numpy`.
4. Χρησιμοποιώντας τη συνάρτηση `zeros` από τη βιβλιοθήκη `numpy` αρχικοποιήστε τον πίνακα `t` ώστε να είναι γεμάτος μηδενικά και να έχει διάσταση `NumberOfPatterns`.
5. **Εκχωρήστε** στη μεταβλητή `ans` την τιμή "γ".
6. Για όσο (`ans = "γ"`)
 - **Εμφανίστε** το παρακάτω **menu** επιλογών :
 - 1 Διαχωρισμός **Iris-setosa** από **Iris-versicolor** και **Iris-virginica**
 - 2 Διαχωρισμός **Iris-virginica** από **Iris-setosa** και **Iris-versicolor**
 - 3 Διαχωρισμός **Iris-versicolor** από **Iris-setosa** και **Iris-virginica****Διαβάστε** την επιλογή (1/2/3)
Αν επιλογή = 1

Δημιουργήστε ένα dictionary `map_dict` με τα εξής ζευγάρια `key/values`:

- "Iris-setosa": 1
- "Iris-versicolor": 0
- "Iris-virginica": 0

Κατόπιν, χρησιμοποιώντας `loop` θέστε για κάθε `pattern` την τιμή στόχου `t[pattern]` ως εξής:

`t[pattern] = 1` αν η 5^ο στήλη για το `pattern` είναι "Iris-setosa"

`t[pattern] = 0` σε διαφορετική περίπτωση

Αν επιλογή = 2

Δημιουργήστε ένα dictionary `map_dict` με τα εξής ζευγάρια `key/values`:

- "Iris-setosa": 0
- "Iris-versicolor": 0
- "Iris-virginica": 1

Κατόπιν, χρησιμοποιώντας `loop` θέστε για κάθε `pattern` την τιμή στόχου `t[pattern]` ως εξής:

`t[pattern] = 1` αν η 5^ο στήλη για το `pattern` είναι "Iris-virginica"

`t[pattern] = 0` σε διαφορετική περίπτωση

Αν επιλογή = 3

Δημιουργήστε ένα dictionary `map_dict` με τα εξής ζευγάρια `key/values`:

- "Iris-setosa": 0
- "Iris-versicolor": 1
- "Iris-virginica": 0

Κατόπιν, χρησιμοποιώντας `loop` θέστε για κάθε `pattern` την τιμή στόχου `t[pattern]` ως εξής:

`t[pattern] = 1` αν η 5^ο στήλη για το `pattern` είναι "Iris-versicolor"

`t[pattern] = 0` σε διαφορετική περίπτωση

Μπορείτε να το κάνετε αυτό χρησιμοποιώντας το `map_dict` και να αποφύγετε εντολή `if-else`.

7. Χωρισμός προτύπων σε πρότυπα εκπαίδευσης και ανάκλησης

Τεμαχίστε τα δεδομένα των πινάκων `x` και `t` σε 4 πίνακες:

- `xtrain` πίνακας με τα πρότυπα που θα χρησιμοποιηθούν στην εκπαίδευση, τα 40 πρώτα πρότυπα της κάθε κλάσης.
- `xtest` πίνακας με τα πρότυπα που θα χρησιμοποιηθούν στον έλεγχο, τα 10 τελευταία πρότυπα της κάθε κλάσης.
- `ttrain` διάνυσμα με τους στόχους που θα χρησιμοποιηθούν στην εκπαίδευση, οι 40 πρώτοι στόχοι της κάθε κλάσης.
- `ttest` διάνυσμα με τους στόχους που θα χρησιμοποιηθούν στον έλεγχο, οι 10 τελευταίοι στόχοι της κάθε κλάσης.

- Χρησιμοποιώντας τη συνάρτηση `plot` από τη βιβλιοθήκη `matplotlib.pyplot` σχεδιάστε

- ο τα διανύσματα `xtrain[:,0]` → άξονας `x`, `xtrain[:,2]` → άξονας `y`, χρησιμοποιώντας τελείες με μπλε χρώμα και
- ο τα διανύσματα `xtest[:,0]` → άξονας `x`, `xtest[:,2]` → άξονας `y`, χρησιμοποιώντας τελείες με κόκκινο χρώμα.

- Μετατρέψτε τα διανύσματα στόχων `ttrain()`, `ttest()` έτσι ώστε
 - Av `ttrain(pattern) == 1` → `ttrain1(pattern) = 1`
 - Av `ttrain(pattern) == 0` → `ttrain1(pattern) = -1`
 - Av `ttest(pattern) == 1` → `ttest1(pattern) = 1`
 - Av `ttest(pattern) == 0` → `ttest1(pattern) = -1`
- Διαβάστε μια ακέραια τιμή για το Μέγιστο Αριθμό Εποχών (`MAXEPOCHS`)
- Διαβάστε μια πραγματική τιμή για το Συντελεστή Εκπαίδευσης (`beta`)
- Υπολογίστε το βέλτιστο διάνυσμα βαρών \tilde{w} εκπαιδεύοντας το μοντέλο Perceptron με τα πρότυπα και τους στόχους εκπαίδευσης: `xtrain`, `ttrain`. Θα χρησιμοποιήσετε τη function `perceptron()` την οποία θα πρέπει να γράψετε εσείς.

```
def perceptron( x, t, MAXEPOCHS, beta ):
# Είσοδος x : πίνακας με τα επαυξημένα πρότυπα
# Είσοδος t : διάνυσμα με τους πραγματικούς στόχους (0/1 ή -1/1)
# Είσοδοι MAXEPOCHS : μέγιστο πλήθος εποχών
# Είσοδοι beta : βήμα εκπαίδευσης
# Επιστρέφει w : τελικό επαυξημένο διάνυσμα βαρών.
```

- Ο αλγόριθμος Perceptron λειτουργεί ως εξής:

Δώσε τυχαία αρχική τιμή στο διάνυσμα βαρών

Για κάθε εποχή εκπαίδευσης

Για κάθε πρότυπο p

Υπολόγισε την διέγερση του νευρώνα $u = \tilde{x}[p, :] \tilde{w}$

Υπολόγισε την έξοδο του νευρώνα

- Εφόσον οι στόχοι έχουν τιμές 0/1 τότε

$$y = \begin{cases} 0, & \text{αν } u < 0 \\ 1, & \text{αν } u \geq 0 \end{cases}$$

- Εφόσον οι στόχοι έχουν τιμές -1/1 τότε

$$y = \begin{cases} -1, & \text{αν } u < 0 \\ 1, & \text{αν } u \geq 0 \end{cases}$$

Διόρθωσε το διάνυσμα βαρών:

- Av $t[p] = y$ τότε μην κάνεις τίποτε
- Av $t[p] \neq y$ τότε $\tilde{w} \leftarrow \tilde{w} + \beta (t[p] - y) \tilde{x}[p, :]$

end

Τερμάτισε εφόσον

- δεν έγινε καμία διόρθωση των βαρών για κανένα πρότυπο
- είτε φτάσαμε στο μέγιστο αριθμό εποχών

end

Μετά το τέλος του αλγορίθμου τυπώστε στην οθόνη το πλήθος των εποχών που εκτελέστηκαν.

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

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

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

- Εφόσον οι στόχοι έχουν τιμές 0/1 τότε

$$predict_{test}[i] = \begin{cases} 0, & \text{αν } u_{test}[i] < 0 \\ 1, & \text{αν } u_{test}[i] \geq 0 \end{cases}$$

- Εφόσον οι στόχοι έχουν τιμές -1/1 τότε

$$predict_{test}[i] = \begin{cases} -1, & \text{αν } u_{test}[i] < 0 \\ 1, & \text{αν } u_{test}[i] \geq 0 \end{cases}$$

- Τυπώστε το εξής γράφημα:

- δείξτε με μπλε τελείες τους πραγματικούς στόχους $t_{train}[i]$ για όλα τα πρότυπα του *train* set
- δείξτε με κόκκινους κύκλους τους εκτιμώμενους στόχους $predict_{train}[i]$ για όλα τα πρότυπα του *train* set.

- 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:

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

- 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

8. Θα εφαρμοστεί η μέθοδος `train_test_split(...)` 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`. Φροντίστε να είναι `numpy arrays` τύπου `float`.
- Βρείτε το πλήθος των προτύπων στο *train* set (P_{train}) και στο *test* set (P_{test}) για παράδειγμα χρησιμοποιώντας τη συνάρτηση `len()`.
- Μετατρέψτε τα διανύσματα στόχων `ttrain()`, `ttest()` έτσι ώστε
 - Av `ttrain(pattern) == 1` → `ttrain1(pattern) = 1`
 - Av `ttrain(pattern) == 0` → `ttrain1(pattern) = -1`
 - Av `ttest(pattern) == 1` → `ttest1(pattern) = 1`
 - Av `ttest(pattern) == 0` → `ttest1(pattern) = -1`

- Υπολογίστε το βέλτιστο διάνυσμα βαρών $\tilde{\mathbf{w}}$ εκπαιδεύοντας το μοντέλο Perceptron με τα πρότυπα και τους στόχους εκπαίδευσης: `xtrain`, `ttrain`.

- 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:

- Εφόσον οι στόχοι έχουν τιμές 0/1 τότε

$$predict_{test}[i] = \begin{cases} 0, & \text{αν } u_{test}[i] < 0 \\ 1, & \text{αν } u_{test}[i] \geq 0 \end{cases}$$

- Εφόσον οι στόχοι έχουν τιμές -1/1 τότε

$$predict_{test}[i] = \begin{cases} -1, & \text{αν } u_{test}[i] < 0 \\ 1, & \text{αν } u_{test}[i] \geq 0 \end{cases}$$

- 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

[Διαβάστε](#) την απάντηση `ans` του χρήστη, αν θέλετε να συνεχίσετε.

Σχόλια:

- **Η εντολή `break`.** Η πρόωρη έξοδος από ένα loop στην Python, όπως και στις περισσότερες γλώσσες, γίνεται με την εντολή `break`.
- **Άνοιγμα figure για plot.** Μπορούμε να ανοίξουμε ένα συγκεκριμένο figure, πχ το figure 1, με την εντολή

```
import matplotlib.pyplot as plt
plt.figure(1)
```

Κατόπιν όλες οι εντολές plot, κλπ θα εκτελεστούν σε αυτό το figure. Αν αποφασίσουμε ότι θέλουμε τώρα να τυπώσουμε σε άλλο figure, πχ το figure 2, απλά καλούμε την εντολή

```
plt.figure(2)
```

και τώρα όλες οι εντολές plot θα εμφανίζονται στο figure 2. Μπορούμε ανά πάσα στιγμή να επιστρέψουμε στο figure 1 ξανακαλώντας

```
plt.figure(1)
```

Οδηγίες κατάθεσης ασκήσεων

1. Συνδεθείτε στο URL: <http://aetos.it.teithe.gr/s>
1. Επιλέξτε το μάθημα “Μηχανική Μάθηση – Εργαστήριο X” (Όπου X ο αριθμός του εργαστηρίου του οποίου τις ασκήσεις πρόκειται να καταθέσετε) και πατήστε επόμενο.
2. Συμπληρώστε τα στοιχεία σας. Πληκτρολογήστε USERNAME 00003 και PASSWORD 30000 (Επώνυμο και Όνομα με ΛΑΤΙΝΙΚΟΥΣ ΧΑΡΑΚΤΗΡΕΣ).
3. Αν θέλετε να καταθέσετε μόνο ένα αρχείο μη το βάζετε σε zip file. Αντίθετα, αν θέλετε να καταθέσετε περισσότερα από ένα αρχεία, τοποθετήστε τα σε ένα zip ή rar file.
4. Επιλέξτε το αρχείο που θέλετε να στείλετε επιλέγοντας “choose file” στο πεδίο FILE1 και πατήστε “Παράδοση”