ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΙ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΑΘΗΜΑ: ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

ΚΑΘΗΓΗΤΕΣ : ΚΩΣΤΑΣ ΔΙΑΜΑΝΤΑΡΑΣ, ΚΩΣΤΑΣ ΓΟΥΛΙΑΝΑΣ

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 6

ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΤΟ ΜΟΝΤΕΛΟ NAIVE BAYES

Σκοπός της άσκησης: Η εκτίμηση της επίδοσης ενός ταξινομητή τύπου **Naïve Bayes** χρησιμοποιώντας την **Γκαουσσιανή κατανομή**. Θα γίνει χρήση της μεθόδου διασταύρωσης (Cross-Validation) και τα κριτήρια επίδοσης:

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

Δείτε πώς λειτουργεί το μοντέλο Naïve Bayes διαβάζοντας τα παρακάτω:

Η μέθοδος ταξινόμησης Naïve Bayes.pdf

Μηχανική Μάθηση - 05 Bayes.pptx

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. Χρησιμοποιώντας τη συνάρτηση zeros από τη βιβλιοθήκη numpy αρχικοποιήστε τον πίνακα `t` ώστε να είναι γεμάτος μηδενικά και να έχει διάσταση `NumberOfPatterns`.
4. **Εκχωρήστε** στη μεταβλητή ans την τιμή "y'.
5. Για όσο (ans = "y')
   - **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 5ο 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 5ο 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 5ο 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`.

6. 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

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

- Εκπαιδεύστε ένα μοντέλο Naive Bayes κάνοντας την υπόθεση ότι τα χαρακτηριστικά ακολουθούν την Γκαουσσιανή κατανομή. Θα χρησιμοποιήσετε την συνάρτηση `nbtrain(xtrain,ttrain)` την οποία θα πρέπει να γράψετε εσείς.

```
def nbtrain( x, t ):
# Είσοδος x : Pxn πίνακας με τα πρότυπα (P=πλήθος προτύπων, n=διάσταση)
# Είσοδος t : διάνυσμα με τους στόχους (0/1)
# Έξοδος model : dictionary που θα περιέχει τις παραμέτρους του μοντέλου
```

Ο αλγόριθμος εκπαίδευσης του μοντέλου NB λειτουργεί ως εξής:

- Χωρίστε τα πρότυπα στην κλάση 0 και στην κλάση 1 (Χρησιμοποιήστε στον πίνακα x κατάλληλα δείκτες `t==0` και `t==1`)
- Βρείτε το πλήθος των προτύπων σε κάθε κλάση
- Υπολογίστε τις εκ των προτέρων πιθανότητες (prior) των δύο κλάσεων (δηλ. πλήθος προτύπων στην κλάση δια το συνολικό πλήθος των προτύπων)
- Για κάθε χαρακτηριστικό `i` (στήλη του πίνακα x) υπολογίστε
  - μ`[0,i]` = μέση τιμή του χαρακτηριστικού `i` για την κλάση 0 (Χρησιμοποιήστε τη συνάρτηση numpy.`mean`)
  - σ`[0,i]` = διασπορά του χαρακτηριστικού `i` για την κλάση 0 (Χρησιμοποιήστε τη συνάρτηση numpy.`std`)
  - μ`[1,i]` = μέση τιμή του χαρακτηριστικού `i` για την κλάση 1
  - σ`[1,i]` = διασπορά του χαρακτηριστικού `i` για την κλάση 1
- `# end for`

Δημιουργήστε το dictionary "`model`" που θα περιέχει τα εξής πεδία:

- όνομα '`prior`', τιμή `prior`: array 2 στοιχείων με τις εκ των προτέρων πιθανότητες των 2 κλάσεων
- όνομα '`mu`', τιμή μ: array 2xn με τις μέσες τιμές των n χαρακτηριστικών για τις 2 κλάσεις
- όνομα '`sigma`', τιμή σ: array 2xn με τις διασπορές των n χαρακτηριστικών για τις 2 κλάσεις

- Αφού εκπαιδεύσατε το μοντέλο με την παραπάνω συνάρτηση κάνετε ανάκληση χρησιμοποιώντας τη συνάρτηση `nbpredict(xtest, model)` την οποία επίσης πρέπει να γράψετε.

```
def nbpredict( x, model ):
# Είσοδος x : Pxn πίνακας με τα πρότυπα
# Είσοδος model : dictionary με τις παραμέτρους του μοντέλου NB
# Έξοδος predict : διάνυσμα με τις εκτιμώμενες τιμές στόχου
```

- Για κάθε πρότυπο p (γραμμή του πίνακα x)
  - Υπολογίζουμε το λόγο των πιθανοτήτων L. Αρχικά θέτουμε

$$L = \frac{\text{prior}[1]}{\text{prior}[0]}$$

  - Για κάθε χαρακτηριστικό `i` (στήλη του πίνακα x)
    - Ενημερώνουμε το L:

$$L \leftarrow L * \frac{G(x[p,i], mu[1,i], sigma[1,i])}{G(x[p,i], mu[0,i], sigma[0,i])}$$

    - Όπου $G(x, \mu, \sigma)$ είναι η συνάρτηση της Γκαουσσιανής κατανομής με μέση τιμή μ και διασπορά σ. (Χρησιμοποιήστε τη συνάρτηση `norm.pdf(x, loc=μ, scale=σ)` αφού πρώτα την κάνετε `import: from scipy.stats import norm`)
  - `# end for`
  - Αν $L < 1$ τότε εκτιμάμε ότι το πρότυπο p ανήκει στην κλάση 0
  - Αν $L > 1$ τότε εκτιμάμε ότι το πρότυπο p ανήκει στην κλάση 1
- `# end for`

- Το διάνυσμα που πήρατε στην έξοδο είναι το $predict_{test}$.
- Καλέστε τη συνάρτηση `evaluate()` από το προηγούμενο εργαστήριο όσες φορές χρειάζεται έτσι ώστε να υπολογίσετε το Accuracy, Precision, Recall, F-measure, Sensitivity και Specificity.
- 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

- Θα εφαρμοστεί η μέθοδος `train_test_split()` για K=9 folds.

2. Στο Cross-Validation loop θα πρέπει να κάνετε τα εξής:

   Για κάθε *fold*

   - Έχετε ήδη δημιουργήσει τους αρχικούς πίνακες προτύπων `xtrain` και `xtest` (χωρίς επαύξηση) καθώς και τα διανύσματα στόχων `ttrain` και `ttest`. Χρησιμοποιήστε τιμές των στόχων 0/1.
   - Εκπαιδεύστε ένα μοντέλο Naive Bayes κάνοντας την υπόθεση ότι τα χαρακτηριστικά ακολουθούν την Γκαουσσιανή κατανομή. Θα χρησιμοποιήσετε την συνάρτηση `nbtrain(xtrain,ttrain)` την οποία θα πρέπει να γράψετε εσείς.
   - Αφού εκπαιδεύσατε το μοντέλο με την παραπάνω συνάρτηση κάνετε ανάκληση χρησιμοποιώντας τη συνάρτηση `nbpredict(xtest, model)` την οποία επίσης πρέπει να γράψετε.
   - Το διάνυσμα που πήρατε στην έξοδο είναι το $predict_{test}$.
   - Καλέστε τη συνάρτηση `evaluate()` από το προηγούμενο εργαστήριο 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(2) 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

3. 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.

Οδηγίες κατάθεσης ασκήσεων

1. Συνδεθείτε στο URL: http://aetos.it.teithe.gr/s
1. Επιλέξτε το μάθημα “Μηχανική Μάθηση – Εργαστήριο Χ” (Όπου Χ ο αριθμός του εργαστηρίου του οποίου τις ασκήσεις πρόκειται να καταθέσετε) και πατήστε επόμενο.
2. Συμπληρώστε τα στοιχεία σας. Πληκτρολογείστε 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”