

ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΙ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΑΘΗΜΑ: ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

ΚΑΘΗΓΗΤΕΣ : ΚΩΣΤΑΣ ΔΙΑΜΑΝΤΑΡΑΣ, ΚΩΣΤΑΣ ΓΟΥΛΙΑΝΑΣ

## ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 5 ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΤΟ ΜΟΝΤΕΛΟ SVM

**Σκοπός της άσκησης:** Η εκτίμηση της επίδοσης ενός ταξινομητή τύπου **Support Vector Machine (SVM)** χρησιμοποιώντας τον πυρήνα **Gaussian (RBF)**. 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)

### 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 "Iris-setosa" + "Iris-virginica",
  - Κλάση 1 (στόχος  $t=1$ ): αποτελείται από τα πρότυπα της κατηγορίας "Iris-versicolor".
2. Θα εφαρμοστεί η μέθοδος `train_test_split()` για  $K=9$  folds.
3. Πριν από το Cross-Validation loop ζητήστε από το χρήστη να δώσει τις παραμέτρους `gamma` και `C` από το πληκτρολόγιο χρησιμοποιώντας την εντολή `input`. (Πχ. δοκιμάστε `gamma=0.01`, `C=10`).
4. 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`. Οι τιμές των στόχων θα είναι 0/1.
- Δημιουργήστε ένα δίκτυο SVM με πυρήνα RBF χρησιμοποιώντας την κλάση SVC ( <http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html> )
- `SVC(C, kernel='rbf', gamma, degree, coef0, ...)` όπου:
  - `C` = παράμετρος  $C$  του SVM
  - `kernel` = επιλογή συνάρτησης πυρήνα μεταξύ 'linear', 'poly', 'rbf', 'sigmoid'. Επιλέξτε 'rbf'
  - `gamma` = η παράμετρος  $\gamma$  του πυρήνα RBF
$$k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$$
  - `degree` = η παράμετρος  $d$  του πολυωνυμικού πυρήνα  $k(\mathbf{x}, \mathbf{y}) = (\gamma \mathbf{x}^T \mathbf{y} + r)^d$

- $coef0$  = είτε η παράμετρος  $r$  του πολυωνυμικού πυρήνα  $k(\mathbf{x}, \mathbf{y}) = (\gamma \mathbf{x}^T \mathbf{y} + r)^d$ , είτε η παράμετρος  $r$  του σιγμοειδούς πυρήνα  $k(\mathbf{x}, \mathbf{y}) = \tanh(\gamma \mathbf{x}^T \mathbf{y} + r)$
- ...και άλλες λουπές παράμετροι (βλ. documentation)
- Εκπαιδεύστε το δίκτυο που φτιάξατε χρησιμοποιώντας τη συνάρτηση
  - `fit()` : με εισόδους το μοντέλο, τον πίνακα των προτύπων εκπαίδευσης (`xtrain`), και το διάνυσμα των στόχων εκπαίδευσης (`ttrain`)
- Αφού εκπαιδεύσετε το μοντέλο κάνετε ανάκληση χρησιμοποιώντας τη συνάρτηση
  - `predict()` : με εισόδους το εκπαιδευμένο μοντέλο και τον πίνακα των προτύπων ελέγχου (`xtest`)
- Ονομάστε  $predict_{test}$  το διάνυσμα που πήρατε.
- 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(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`

5. 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

6. Βάλτε το παραπάνω cross-validation loop μέσα σε ένα διπλό loop για το gamma και το C. Δοκιμάστε τιμές

- `gamma = 0.01, 0.03, 0.1, 0.3, 1`
- `C = 1, 10, 100, 1000`

Βρείτε για ποιο συνδυασμό gamma και C πετυχαίνετε την καλύτερη μέση τιμή test-accuracy και τυπώστε την στην οθόνη.

Παρατηρήστε αν εμφανίζονται διαφορές στην επίδοση του αλγορίθμου.