Θέμα: <u>Αλγόριθμοι ταξινόμησης (ή κατηγοριοποίησης) δεδομένων</u>

Η ιστοσελίδα http://archive.ics.uci.edu/ml/index.html αποτελεί μια πολύ γνωστή βάση πειραματικών δεδομένων που χρησιμοποιείται για τη μελέτη της επίδοσης μεθόδων αναγνώρισης προτύπων και μηχανικής μάθησης σε διάφορα προβλήματα.

Τα παρακάτω πειραματικά σύνολα δεδομένων αφορούν προβλήματα δυαδικής ταξινόμησης (*binary classification*), δηλ. περιέχουν δεδομένα δύο (2) κατηγοριών:

- spam dataset https://archive.ics.uci.edu/ml/machine-learning-databases/spambase/
- credit card clients dataset https://archive.ics.uci.edu/ml/machine-learning-databases/00350/

Κάθε μέθοδο ταξινόμησης που θα υλοποιήσετε:

- Θα την αξιολογήσετε χρησιμοποιώντας ως μέτρα αξιολόγησης τα ακόλουθα:

  - *Accuracy* – δηλ. το (%) **ποσοστό επιτυχίας των αποφάσεων** του ταξινομητή, και

  - *F1 score*

$$ACC = \frac{TP + TN}{P + N}$$

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

$$F1 = 2\frac{Precision * Recall}{Precision\ + Recall}$$

  ,

  όπου *TP: true positives, TN: true negative, FN: false negative, FP: false positives, P: positives, N: negatives.*

- Θα ακολουθήσετε την στρατηγική ***10-folds cross validation:*** συγκεκριμένα, θα χωρίσετε με τυχαίο τρόπο το αρχικό σύνολο δεδομένων σε δέκα (10) ξένα μεταξύ τους υποσύνολα (*10 folds*) και σε κάθε ένα θα μετρήσετε την επίδοση της μεθόδου (*testing set*) κάνοντας εκπαίδευση στα υπόλοιπα 9 υποσύνολα (*training set)*. Είναι δικαιότερος ο διαχωρισμός να γίνει αναλογικά για κάθε κατηγορία, έτσι ώστε σε κάθε *fold* η αναλογία των δεδομένων ανά κατηγορία να είναι η ίδια. Η συνολική επίδοση κάθε μεθόδου ταξινόμησης θα προκύψει από την ποσοστιαία **μέση τιμή στα *10 folds***.

Θα κατασκευάσετε την παρακάτω παραλλαγή της μεθόδου *LVQ* κατά την οποία προσπαθούμε να χωρίσουμε δυναμικά τον χώρο των δεδομένων σε σφαιρικές - Γκαουσιανές (*Gaussian*) περιοχές της ίδιας κατηγορίας. Τότε η διαδικασία της απόφασης ταξινόμησης για ένα «άγνωστο» σημείο θα είναι ανάλογη της κατηγορίας

της κοντινότερης Γκαουσιανής περιοχής. Τα βήματα του **αλγορίθμου**[1] που θα πρέπει να προγραμματίσετε είναι τα παρακάτω:

- Αρχικά[2] έχετε *K* περιοχές, μία περιοχή για κάθε κατηγορία *j*, που περιγράφονται από το μέσον $\boldsymbol{\mu_j}$ και την ακτίνα της $\boldsymbol{\sigma_j}$. Σε κάθε κατηγορία αντιστοιχεί και μία κατηγορία, έστω $\boldsymbol{C_j}$.

- *Επαναληπτικά για M*N φορές, όπου N ο αριθμός των παραδειγμάτων του συνόλου εκπαίδευσης και M ένας ακέραιος αριθμός > 1 (π.χ. M=10) :*

  1. **Επιλογή** ενός *τυχαίου* παραδείγματος από το σύνολο εκπαίδευσης, έστω ***x*** τα χαρακτηριστικά του και ***y*** η πραγματική του κατηγορία.

  2. **Εντοπισμός** της Γκαουσιανής περιοχής (από το τρέχον σύνολο των *K* περιοχών) που ανήκει το παράδειγμα σύμφωνα με την Γκαουσιανή συνάρτηση ομοιότητας:

     $$e^{-\frac{|x-\mu_j|^2}{2\sigma_j^2}}$$

     Έστω ***j**** η «νικήτρια» περιοχή (***winner region***), δηλ. αυτή με την μεγαλύτερη ομοιότητα.

  3. **Προσαρμογή** των παραμέτρων της νικήτριας περιοχής ανάλογα με την ορθότητα της απόφαση:

     a. ***Αν y = C_j*** τότε (***success***) μεταβάλλουμε[3] τις παραμέτρους της περιοχής, *επιβραβεύοντας* (***reward***) την απόφαση, ως εξής:

     $$\mu_{j*} = (1-a)\mu_{j*} + a\,x \qquad and \qquad \sigma_{j*} = \sigma_{j*} + a\,dist(x,\mu_{j*})$$

     b. ***Αν y ≠ C_j*** τότε (***failure***) τότε μεταβάλλουμε τις παραμέτρους της περιοχής, *τιμωρώντας* (***penalty***) την απόφαση:

     $$\mu_{j*} = (1+a)\,\mu_{j*} - ax \qquad and \qquad \sigma_{j*} = \sigma_{j*} - a\,dist(x,\mu_{j*})$$

     δημιουργούμε μία ***καινούργια*** Γκαουσιανή περιοχή (***K=K+1***) με κέντρο το σημείο ***x*** και ακτίνα μία *default* μικρή τιμή $\boldsymbol{\sigma_{init}}$.

     $$\sigma_K^2 = \sigma_{init}^2 \qquad and\ \mu_K = x$$

[1] Ο αλγόριθμος αυτός είναι μία παραλλαγή που στοχεύει στον διαμερισμό του χώρου σε σφαιρικές περιοχές όσο περισσότερο μεγαλύτερες και διαυγείς ως προς την κατηγορία των δεδομένων. Δεν υπάρχει στην επίσημη βιβλιογραφία και πιθανώς να παρουσιάζει κάποιες ατέλειες. Παρόλ' αυτά ο στόχος είναι απλά να κατασκευάσουμε την δική μας μέθοδο ταξινόμησης και να την συγκρίνουμε με ήδη υπάρχουσες.

[2] Μπορείτε ως αρχικές τιμές να επιλέξετε για κάθε μία κατηγορία το δειγματικό μέσον τους για το μ_j και το την δειγματική διασπορά τους για το σ_j^2.

[3] Η παράμετρος *a* εκφράζει το ποσό μεταβολής των παραμέτρων, δηλ. είναι ο ρυθμός μάθησης (*learning rate*). Η τιμή του θα πρέπει να είναι μικρή, π.χ. *a=0.001*. Ως *dist* να χρησιμοποιήστε την *Ευκλείδια απόσταση*, ενώ ως $\sigma^2_{in\ it}$ να έχετε ένα μικρό ποσοστό (π.χ. 10%) της συνολικής διασποράς όλων των δεδομένων του συνόλου εκπαίδευσης.

Παράλληλα με τον παραπάνω αλγόριθμο, θα πρέπει να προγραμματίσετε[4] και να μελετήσετε τις εξής **μεθόδους ταξινόμησης (που έχουμε διδαχτεί)**:

[Method 1]. ***Nearest Neighbor k-NN με Ευκλείδια απόσταση*** υποθέτοντας ***k*** κοντινότερους γείτονες (μεταβλητό αριθμό *k*).

[Method 2]. ***Neural Networks:*** Νευρωνικά δίκτυα με σιγμοειδή συνάρτηση ενεργοποίησης σε κάθε νευρώνα (*α*) με ένα (1) κρυμμένο επίπεδο δοκιμάζοντας διαφορετικό αριθμό *K* νευρώνων, και (*β*) με δύο (2) κρυμμένους νευρώνες δοκιμάζοντας διαφορετικούς αριθμούς *K1* και *K2* νευρώνων ανά επίπεδο. Για την εκπαίδευσή τους χρησιμοποιήστε *2* εναλλακτικά σχήματα: (i) *Gradient Descent* και (ii) *Stochastic Gradient Descent*.

[Method 3]. **Support Vector Machines (SVM):** Μηχανές διανυσματικής στήριξης, χρησιμοποιώντας (α) γραμμική συνάρτηση πυρήνα (*linear kernel*) και (β) *Gaussian* συνάρτηση πυρήνα (*kernel*) δοκιμάζοντας διάφορες τιμές της παραμέτρου ακρίβειας (*accuracy – inverse variance*).

[Method 4]. ***Naïve Bayes[5] classifier*** υποθέτοντας (ανεξάρτητη) κανονική κατανομή (*normal distribution*) για κάθε χαρακτηριστικό.

Δώστε ένα **σύντομο *report* (*pdf αρχείο το οποίο και θα στείλετε*)** με τον τρόπο κατασκευής των μεθόδων, τα αποτελέσματα των δοκιμών ανά μέθοδο, όπως επίσης την βέλτιστη μέθοδο που θα προκύψει από την σύγκριση. Να δοθεί επίσης και ο κώδικας που κατασκευάσατε (*Matlab ή Python ή C/C++*).

[4] Επιτρέπεται να χρησιμοποιήσετε έτοιμες ρουτίνες για τις μεθόδους από τις βιβλιοθήκες της γλώσσας προγραμματισμού που θα επιλέξετε (π.χ. *sklearn της Python*).

[5] Την μέθοδο του *Naïve Bayes* δεν την έχουμε διδαχτεί ακόμα. Ελπίζω να γίνει.