Assignment 7A

Computer Science Department, University of Crete

MACHINE LEARNING - CS 577, Fall 2022

Deadline: Friday, 8/1/2023, 23:55 on e-learn (`https://elearn.uoc.gr`).

Deliverable files: Submit a zip/tar etc. file containing:

- a report in PDF with ALL answers to theoretical tasks, observations and figures produced by scripts.
- **all** Python script files written by you in the scope of the assignment.

NOTE: 10% penalty on your assignment grade will be applied in failing to follow the above instructions for deliverables or not explaining your code/ having comments!

Exercise 1 - Support Vector Machine (Theoretical) [50 points]

In the table below you are provided with the training data of a 1-norm, soft-margin SVM as well as the (fictional) Lagrange multipliers α that stem from training the model with cost C = 10. The data are fictional, with values that make easy the computations. With lower x are denoted the input training vectors and capital X the variables, while y is the class.

1. [**10 points**] Write the decision function $f(x_{test})$ to classify a new input vector x_{test} (thus, you must also compute the value of b in the SVM equations).
2. [**25 points**] Show how the provided Lagrange multipliers cannot really be the solution to the proposed problem.
3. [**15 points**] Now assume a Gaussian kernel ($\sigma = 1$) and, as above, write the decision function $f(x_{test})$ to classify a new input vector x_{test} (again, you must also compute the value of b in the SVM equations).

Sample	a	y	X1	X2	X3
x1	1	-1	1	0	1
x2	1	1	0	-1	0
x3	10	-1	1	1	-1
x4	0	-1	1	0	1
x5	0	1	0	-1	0

Exercise 2 (Programming) [50 points]

In this exercise you will use the SVM[1] classifier as provided by scikit-learn. For this exercise you will implement the following in one script, when we run this script the results will be visible in the console output and the required plots will be produced.

Data: the data in the file *Dataset7_XY.csv* (the last column contains the label, the other columns contain the features).

Specifically, you will need to implement the following:

- You will split your data into 70% train and the remaining 30% will be the hold-out set. Perform a stratified split.
- You will then apply a *Stratified 5-Fold Cross Validation* in the 70% (reuse the code of the previous exercise[2]). Your performance metric will be *ROC AUC*.
- With the cross validation procedure you will tune and select the best SVM configuration, searching for the best combination of the following hyper-parameters: **C**: {0.01, 1, 10}, **kernel**: {'linear', 'rbf'} and **gamma**: {0.1, 1, 10}.
- Report the best configuration found (reporting the hyper-parameters) and its cross validation performance (the average AUC over the 5 test folds).
- Finally, build the final model and test it on the hold-out set. Report the performance on the hold-out set.

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`

[2] In case that you did not deliver the previous exercise, you should use the StratifiedKFold of scikit-learn