# MACHINE LEARNING - CS 577

**Computer Science Department, University of Crete**

October 2022

## Assignment 3

**Deadline**: Thursday, 31/10/2022, 23:59 on e-learn (`https://elearn.uoc.gr`).

**Deliverable files**: Submit a zip/tar etc. file containing a report in PDF with the answers **AND all** Python(.py) files written by you in the scope of the assignment. The final grade will be the result of the quality of your submitted results in your report, together with the correctness of your submitted code.

**Python Version**: Use python **3.6**

**Regarding the programming exercises: DO NOT alter the function calls or the variable names. Keep them as stated by the tasks!**

## Exercise 1 - Logistic Regression (Theoretical) [15 points]

Prove that if the data are linearly separable, the logistic regression weights ($\boldsymbol{w}$) go to infinity (very large values).

***Hint**: What happens when you have linearly separable data? (A figure can help you explain). You can suppose that your data are centered around 0 (and if not, you can just translate them to be centered, without losing generality), i.e. that $x = 0$ perfectly separates your data. You will find that you can conveniently split the likelihood function as well, and use this trick to prove the theorem.*

## Exercise 2 - Evaluation of Classifiers (Programming) [70 points]

In this exercise you will apply some other classifiers on the given data-sets. The data-sets consist of two parts:

- Dataset3.2_*_Y: the class label

- Dataset3.2_*_X: input data
- Dataset A contains only categorical. Dataset B containts only continuous, and Dataset C [**optional 10 points bonus**] contains a mix

You will apply the following classification algorithms on the data-sets (general case):

- Trivial: Classification to the most frequent class (every new sample is predicted to belong to the most frequent class encountered in the training set).

  ***Note**: This classifier is not provided: you will have to implement a trivial_train(X, Y) and trivial_predict(X) function*

- NBC: Naive Bayes Classifier with L = 1.
  ***Hint**: Use your implementation of NBC [bonus 5 points] or existing versions:*
  `https://pypi.org/project/mixed-naive-bayes/`

  ***Note**: You will have to split your data in categorical (less than 15 values) and continuous!*

- Logistic regression: `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html`
  (C = $\frac{1}{L}$)

  ***Note**: Because some of your data are categorical (for simplicity you have numbers not strings but each categorical value could have not been a number), you will have to use one hot encoding to encode them. What this does is produce one new feature for each value of each variable. Eg for a variable X={1,2,3} it will create 3 columns each having 1 when 1,2, or 3 respectively appears or 0 otherwise. Then merge this dataset with the rest of the continuous to use for training: sklearn.preprocessing.OneHotEncoder*

The analysis consists of the following steps:

1. Randomly split each dataset into two parts, one containing 75% of the samples (training set) and one containing the remaining 25% (test set)
2. For K = 50, 60, 70, 80, 90, 100
   a) Keep only the first K% samples of the training set
   b) Apply all of the aforementioned classifiers on that subset
   c) Evaluate them on the test set (the 25% you left out in step 1)
   d) Compute their accuracy (the percentage of correctly classified samples)
3. Repeat steps 1 and 2 100 times and compute the average accuracy for each algorithm and each K. In the end, you should have 6 values for each dataset and algorithm.

4. For each dataset, create a plot showing how the accuracy of each algorithm varies with increasing sample size.
   a) x axis: the sample size of the training set (50%, 60%, ..., 100%)
   b) y axis: the average accuracy of each classifier
   c) Use titles, axis labels and legends.
5. Write a report containing one plot for each dataset as well as any observations/comments you have about the results. Some possible observations are the following:
   a) How does NBC compare to the trivial and LR classifiers?
   b) How does different sample size affect the results?
   c) Were the results expected? Why?

Study the results and make your observations.

***NOTE:*** *The whole analysis procedure should be implemented in a script called* "run_analysis". ***Running this script should reproduce all results and plots.***

## Exercise 3 - Regularization of Logistic Regression (Programming) [15 points]

You will explore what happens to the logistic regression weights ($\boldsymbol{w}$) when the hyper-parameter $\lambda$ ruling the regularization is varied.

Using existing implementations of the logistic regression (see previous exercise), train the classification using *a*) no penalization, and a *b*) "Lasso" regularization. For the Lasso, apply a regularization strength $\lambda$ of 0.5, 10, and 100 (hence you will have 1 no penalty + 3 Lasso different models).

Then, for each method, display the values of the weights. The $x$-axis of the plot shall just be an indexing of the weights $w_0$, $w_1$, ... $w_n$, while the $y$-axis shall report the values of the corresponding weight. Remember to use the same $y$-axis range when plotting, in order to better visualize the effect (zoom accordingly to include all data points).

Discuss the plots by explaining why the regularization causes the weights to obtain the displayed values. Explain the differences (if any) with respect to the model without penalization, and extreme behaviours (if any).

The necessary files (uploaded) are:

a. $Dataset3.3\_X.txt$: sample values (samples along rows and features along columns)

b. $Dataset3.3\_Y.txt$: sample labels (samples along rows)