Assignment 3

Computer Science Department, University of Crete

MACHINE LEARNING - CS 577, Fall 2024

Assignment 3

Deadline: Wednesday, 6/11/2024, 23:59 on e-learn (`https://elearn.uoc.gr`).

Deliverable files: Submit a folder containing a report in PDF with any answers or comments **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: Ideally, use python **3.6**. If you have any issues with this version of Python feel free to use a more recent version.

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

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

Hint: *What happens when you have linearly separable data? You can suppose that your data are centered around 0 since you can translate them, ie. "move" every data point by the same distance, without losing generality, such 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 classifiers to the given data-sets and evaluate their performance. The data-sets consist of two parts:

- Dataset3.2_*_Y: Class Labels aka Targets
- Dataset3.2_*_X: Features aka Predictors

- Dataset A contains categorical data, Dataset B contains continuous data and Dataset C [***Optional 10 Points (Bonus)***] contains a mix of both categorical and continuous data.

You will apply the following classification algorithms to the data-sets:

- Trivial (baseline): Classification to the most frequent class where every input sample is predicted to belong in the most frequent class of the training set.

  This classifier is not provided, you will have to implement a trivial_train(X,Y) and trivial_predict(X) function.
  Note*: It is always helpful to compare a classifier's results with a baseline model.*

- NBC: Naive Bayes Classifier with L = 1.
  You are allowed to use this implementation of NBC:
  `https://pypi.org/project/mixed-naive-bayes/`
  Alternatively, use your own implementation of NBC.

- Logistic Regression: You are allowed to use the following LR implementation: scikit-learn Logistic Regression.
  (C = $\frac{1}{L}$)

The analysis consists of the following steps:

Pre-Processing: Some of the data provided are categorical, where each class is represented as a number. You will have to pre-process your categorical variables using One-Hot Encoding. This produces a feature for each unique value in a variable. ex. Given X={1,2,3} One-Hot Encoding will produce 3 binary columns indicating the value of the Target variable.
scikit-learn One-Hot Encoder
`https://en.wikipedia.org/wiki/One-hot#Machine_learning_and_statistics`

1. **Randomly** split each dataset into a 75% *training set* and a 25% *test set.*

2. Repeat for K = 50, 60, 70, 80, 90, 100

   a) Keep the first K% samples of the training set

   b) Apply all three classifiers mentioned above to that data subset

   c) Evaluate them on the *test set* from Step 1.

   d) Compute the accuracy (percentage of correctly classified samples).

3. Repeat steps 1 and 2 and compute the average accuracy over 100 repetitions for each K. Repeat the process for each classifier and each dataset. You should end up with 6 values for each of the three classifiers per dataset.

4. For each dataset, create a plot showing the relationship between accuracy and sample size for each classifier.
    a) x axis: sample size of the training set (50%, 60%, . . . , 100%)
    b) y axis: average accuracy of each classifier over 100 reps
    c) Use titles, axis labels and legends.
5. Write a brief report containing the plots for each dataset as well as any observations or comments you have about the results.
    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?

NOTE: *The entire analysis 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 λ 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 λ 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. *Dataset*3.3_*X.txt*: sample values (samples along rows and features along columns)

b. *Dataset*3.3_*Y.txt*: sample labels (samples along rows)