Assignment 3

Classification

## Objective

* Learn basic techniques for classification using scikit-learn

## Data Description

In this assignment you are given an anonymized data.csv dataset in the datasets/assignment3/ shared folder. The provided.ipynb file shows how to load the dataset and create a training + test dataset out of it. The target variable in this dataset admits values {0,1} so the prediction problem Y ~ f(X) is a binary classification problem.

## Assignment Deliverables

Augment the provded .ipynb notebook with your analysis and complete the following in that notebook. Make sure that the test dataset is never analyzed until you are ready to compare the performance of all your models.

**Analyze basic characteristics of your training Data**

What are the basic characteristics of your Target variable? What about the characteristics of the features? Which of the features are numeric and which of them are discrete / possibly categorical? Which features are related? Visualize to demonstrate your findings and explain your results in the provded .ipynb file.

**Preprocess the data**

What type of pre-processing does your training data need? Justify your answers using the results of the previous section. Create a pre-processing pipeline that handles both numerical and categorical features. Make sure to use either pre-existing scikit-learn preprocessing machinery or wrap any additional pre-processing steps into a TransformerMixin class. If your preprocessing step depends on some parameters, make sure to expose them in a way that will allow you to test for optimality of these parameters later.

**Fit a few classification models**

Use the SGDClassifier model with ‘log’ loss. Use three versions of the classifier corresponding to the penalty parameter set to *‘l2’, ‘l1’, or ‘elasticnet’.*

For each of the penalty choices use GridSearchCV to calculate the optimal alpha parameter according to the F1 score. Make sure to read the GridSearchCV and SGDClassifier documentation on how to do that. In the case of ‘elasticnet’ penalty parameter, also find the optimal l1\_ratio parameter.

Using joblib, save the three optimal parameter models as optimal\_l2.pkl, optimal\_l1.pkl and optimal\_elasticnet.pkl.

Is F1 score the appropriate metric? Plot the ROC curve and the Precision-Recall curve of your three optimal model parameter models. How would the ROC / PR curves look differently if you used suboptimal model parameters? Explain your results.

**Extra credit 1**

Implement any other scikit-learn classification models of your choice and find the top model according to the F1 score metric using GridSearchCV. Top 3 F1 scores in the class get an extra credit.

**Finally use the test set**

Next, out of all the optimal models that you computed above, which one performs best in the test set. Save your best model as a best\_model.pkl. What is it’s F1 score on the test set?

**Extra credit 2**

Ieshan will test the F1 score of each of the best\_model.pkl files on an out-of-sample validation set and will reward an extra credit to the top model.