CSCI-P556
Fall 2018
Assignment 3
Due 11:59PM, Nov. 2, 2018

Utsav Patel (utpatel)

November 5, 2018

1 Introduction

In this assignment you are being given the following four data files:

1. a3-train.data
2. a3-train.labels
3. a3-test.data
4. a3-test.labels

There are 2,000 training rows and 600 test rows, each row has 500 features, and there are only two labels, -1 and 1.

The tasks for this assignment consist of creating a Jupyter Notebook in which you will perform exploratory data analysis, build some baseline models, perform feature engineering, build some more models that you will optimize, and finally benchmark those models. Additionally, you will have to write a written report in LaTeX in which you will explain all the work done for each of the following sections. There isn't strict page limit, but keep in mind that this is a graduate course and submitting sections that are two sentences long is not acceptable, nor are 50 pages report. Be concise.

2 Exploratory Data Analysis

The first step in the machine learning pipeline is taking a look at the data to make sure that you understand what you are working with. This step includes tasks such as checking for unique identifier columns that have to be discarded, making sure there are no missing or weird values, are the labels balanced, etc.

1. Firstly for EDA, I did full EDA by pandas_profiling library and built a ProfileReport. It explains shape of DataFrame ,variable types-(Numeric, Categorical,Boolean,Date) , missing values and rejected columns(Pandas Profiling Report rejects 2 columns A and B if they have high correlation among them AND both are having almost same correlation with the target column– which helps us determine that these two columns are almost replica of each other and we don't receive any extra information by using both of this two columns ie Keep one and discard other). Also ProfilingReport tells us for each column: no of Distinct values, Unique value %, Missing Value % , Number of Missing values, % of infinite values, Number of infinite values, Mean of values of that column, Minimum value of that column, Maximum value and % of Zeros in the column. Also it has Toggle details button which when clicked consists of Quantile Statistics, Histogram, Common Values and Extreme Values section which are self explanatory if we see in jupyter notebook submitted .
2. Counted no of rows for target variable -1 and that for +1. We got to know that there are equally

1000 rows each for both of target variables.
3. Analyzed which columns were having somewhat higher Minimum and Maximum values for their columns. eg: Col : 105 Max is : 999 Min is : 0 . Now this two values of Minimum and Maximum are somewhat not similar and that needs to be noted while doing EDA. Some other columns having similar awkward min and max values were: 105,152,436,410,448,446.
4. Scaling with StandardScalar

3 Baseline Models

Having identified a suitable feature set to train our models with, you will train at least three baseline models with the default parameters to get an idea of what is the minimum performance that can be achieved before performing feature engineering and optimizing the model's hyperparameters. I performed many Baseline models:

Model	Accuracy
1. Logistic Regression	**58**%
2. K Nearest Neighbour	**51**%
3. Support Vector Machine - SVM	
A. kernel="linear"	**58**%
B. kernel='poly'	**58**%
C. kernel='rbf'	**58**%
D. kernel='sigmoid'	**59**%
4. DecisionTreeClassifier :	**78**%
5.RandomForestClassifier :	**63**%
6. Naive bayes :	**59**%
7. Perceptron	**55**%
8.Multi-Layer Perceptron	**57**%
9.XGBoost Classifier	**77**%
10. AdaBoost Classifier	**60**%

4 Feature Engineering

Now the fun part begins. Can you make any changes to the data that could improve your models performance? Maybe dropping correlated columns or making a new feature out of two or more. Be creative.
1. First Approach of mine was to try PCA(Principal Component Analysis). I made all models with PCA but couldn't achieve accuracy more than 88% for XGBoost and not more than 82% for Random Forests. Thus I had to undo this idea and decided NOT TO USE PCA.
2. I dropped 10 columns that were having very high correlation with other columns and those two columns were also having similarity in correlation with the Target column. So if A and B are two columns , if condition1(both are HIGH correlated) AND condition 2 (both have nearly SAME correlation with the target column) satisfies, any one we are keeping and we are dropping other. Thus, we found 10 such columns from Pandas Profiling in EDA section.
3. Next, I checked whether there are any other columns that satisfies condition1 and 2 mentioned in previous point, but fortunately we found no such column.

5 Model Building

In this section you will build at least three models in which you will try to achieve the highest possible performance on the test set. **Your models performance on the test set will be taken into consideration when we grade this assignment**. You are free to use whichever models and techniques

(stacking, ensembles, etc.) you want and are not restricted to the ones that we have covered in class.
1. I searched for most 11 positively correlated columns and most 10 negatively correlated columns which might prove to be the best consideration as it will help us describe the target in a much accurate way than others. Models I built with this logic and after doing Parameter Tuning, I found below final accuracies:

Model	Accuracy
1. Logistic Regression	**60**%
2. K Nearest Neighbour	**70**%

3.RandomForestClassifier with using 10 positively and 10 negatively correlated columns : **82**%

4. Random Forest with using Feature_importances_ attribute and retrieving just 9 most important features - **90**%

5. Perceptron : **59**%
(could not achieve more than this even after parameter tuning :( )

6.XGBoost : Same Concept of using 10 most positive and 10 most negative- **83**% accuracy

7. XGBoost: Using whole dataset but dropped that 10 columns which we found highly correlated (the ones we dropped in feature engineering): **84**% accuracy

8.XGBoost: Approach: I took only 10 columns to train and then predicted with Accuracy of **88**% . Those 10 columns were from correlation matrix . It was smart choices that dont repeat any two columns having correlation and choosing columns which were having high correlation with Target .

9. XGBoost:**88**% Approach: Combined above 10 self chosen features with 9 features that Random Forest found useful by feature_importance , but still I could not get more than **88**% with XGBoost.

6 Discussion

Finally, describe how the models compared against each other and the baselines. Was their performance as good as you expected? Were there any challenges? Is there anything that could improve your models performance?

Models were really not as good in Baseline phase. The accuracy of all models except Decision Trees and XGBoost were not good at all. The performance was never what I expected .
Challenges: The challenge that I found mostly during the assignment was coming up with hyperparameter that can increase the accuracy. Now, in order to come up with accuracy I had to make for loops of all the parameters that I need to come up with. Suppose that there are 3 hyperparameters, I had to make three for loops and changing each values in each for loop: ex: first for loop of no_estimators, second of n_jobs, third of random_state. So to calculate best of all, it took more than 20 minutes per run and still sometimes I had no good result. My jupyter notebook got crashed 2 times and I had to reinstall anaconda. Also, intrinsically to come up with such a model that could perform too well in this kind of problems was hard. I tried almost all models I had knowledge of, tried parameter tuning of all models but still accuracy never went up 90% .

I tried almost everything that could improve model's performance. Still, there is a place that I am missing something. Maybe I should have tried Keras-Tensorflow also and LightGBM too. I tried Keras-

tensorflow on first set in Baseline and it never increased 60% in it , so still wondering what could work. Maybe We need to focus more on the EDA and Feature Engineering part as I feel that the noise is purposely added to the dataset and thus some bunch of 5-6 columns are made 500 columns by just adding more noise.

Overall Thanks for this assignment, it was really good experience.