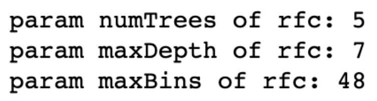
# Assignment 2 Report

## 1. Question 1

1.1

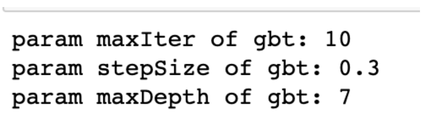
The main purpose of  Q1.1 is to use pipeline and cross-validation (a.k.a CV) to find the best parameters for RandomForest (a.k.a RF) and Gradient boosting (a.k.a GBT ) models. According to the requirement of question, I extract 5% of the whole dataset as the dataset to find the best parameters by using “.sample( )” function. Then I create a pipeline which include feature vectorization and random forest classifier algorithm. After creating a parameter grid and choosing three option for three parameters for RF classifier, the CV process can be started. I fit the training data for CVmodel of RF classifier and find the best parameter of my choice. Then I do the same process to find best parameters for GBT model.

The picture below shows the best parameters of RF classifier.



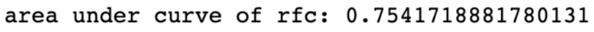
```
param numTrees of rfc: 5
param maxDepth of rfc: 7
param maxBins of rfc: 48
```

In the same way ,the picture below shows the best parameters of GBT classifier.

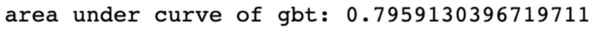


```
param maxIter of gbt: 10
param stepSize of gbt: 0.3
param maxDepth of gbt: 7
```

As performance measure, I report the area under curve.



```
area under curve of rfc: 0.7541718881780131
```



```
area under curve of gbt: 0.7959130396719711
```

1.2

In the previous stage, I got the best parameters for each model, in this part I use these parameters to train the models on whole dataset.

| Performance \ model | Random Forest | Gradient Boosting |
|---|---|---|
| **Area under curve** | 0.758028787837179 | 0.797505043605141 |
| **Accuracy** | 0.688996867142092 | 0.719796837074415 |

Training time on 10 cores: totally 655 second
Random Forest : 305 second
Gradient Boosting : 350 second

Training time on 20 cores: totally 556 second
Random Forest : 255 second
Gradient Boosting : 301 second

1.3

I get the three most relevant feature for both models by using featureImportance function.

For RF classifier, they are 26th 28th 27th columns of features.
For GBT, they are 26th 27th 28th columns of features.

1.4

Observation 1: For performance measure, area under curve of both algorithms are larger than accuracy.
Observation 2: For performance of GBT, both area under curve and accuracy are larger than random forest.
Observation 3: When training on HPC, using 20 cores is faster than using 10 cores.

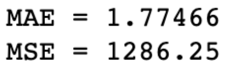
## 2. Question 2

2.1

For the missing data, I remove the rows with missing data. Because the dataset is highly imbalanced, the missing data doesn't influence the training result.
For categorical values, I convert them by applying StringIndexer( ) to assign indices to each category in the categorical columns and applying OneHotEncoderEstimator() to convert categorical columns to onehot encoded vectors.
For imbalanced data, I assign weights for each class to penalize the majority class by assigning less weight and boost the minority class by assigning bigger weight. I add a new column of feature called " weight " and assign the inverse ratio of each class. Because 99% of the claim amount is 0 so I assign 0.01 for 0 and 0.99 for non-zero.

2.2

The picture below is Mae and mse of linear regression:

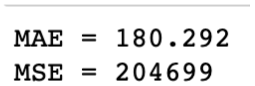


```
MAE = 1.77466
MSE = 1286.25
```

Training time on 10 cores: 131 second
Training time on 20 cores: 102 second

2.3

The picture below is Mae and mse of  Gamma regression:



```
MAE = 180.292
MSE = 204699
```

Training time on 10 cores: 896 second
Training time on 20 cores: 806 second

2.4

Observation 1: In pre data processing stage, add a column called “weight” can solve the problem of label data is highly imbalanced in some extent.
Observation 2: As performance, mean absolute error is much smaller than mean square error.
Observation 3: For performance of training models in tandem, Mae and Mse are larger than the performance of a single linear regression.