Report

January 14, 2020

Introduction of data

The prepossessing steps based on the origin data is to exclude those trips with speed as 0 and the distance during each trip less than 10 miles, which makes the number of rows decrease from 1048575 to 1045244. Our data has only 7753 trips with critical events that is made up only 0.0074% of the total data. As a result, when we do the analysis, we should use some re-sample methods.

In terms of the features, here I applied the *interval time, speed_mean, speed_sd, distance, age, prep_intense, prep_prob, wind, visibility and cum_drive*. The output is a binary variable named *CE* indicating whether there happens the critical events for each trip.

Methods

Re-sampling methods

There are two basic methods helping us to solve the unbalance data. The one is under-sampling and the other one is over-sampling. In python, I used NearMiss algorithm to math the data without critical events to the other one. As the oversampling, the algorithm named SMOTE(Synthetic Minority Oversampling Technique) is applied. The problem is that sometimes the SMOTE could generate noisy samples by interpolating the data points between marginal outliers and inliers. Consequently, the other two methods are better in dealing with noisy factors during the oversampling which are SMOTETomek and SMTEENN.

Procedure of Neural Network(NN) analysis based on over-sampling

For the NN, I build three layers and the output function is a sigmoid function. The whole procedure is very straightforward. Firstly, the whole data set has been split into two groups based on *CE*. For each group, it randomly selected 20% of data as the test set and the left part is the training data. Combining the test and train data from each group, the final test data and train data have been generated. After that, the feature scaling has been used to make sure the range among all the features are not huge different. The following step is to use oversampling method to fit the training data. Here I only consider the SMOTETomek. Finally, the training data and test have been prepared. It is noted that the oversampling method only applied on the training data.

Before the NN, I used the K-fold cross validation on the training data to evaluate the NN model's performance and the k is 5. The validation score is 0.395735, 0.400196, 0.708272, 0.570189 and 0.533187. If we train the whole training data and do the prediction on the test training data , the overall accuracy is around 67%. However, since the test data is an unbalance data, ROC could be a better way to evaluate the performance. The image 1 Comparing with the validation score, the NN model under the oversampling makes sense and doesn't have the over fitting problem.

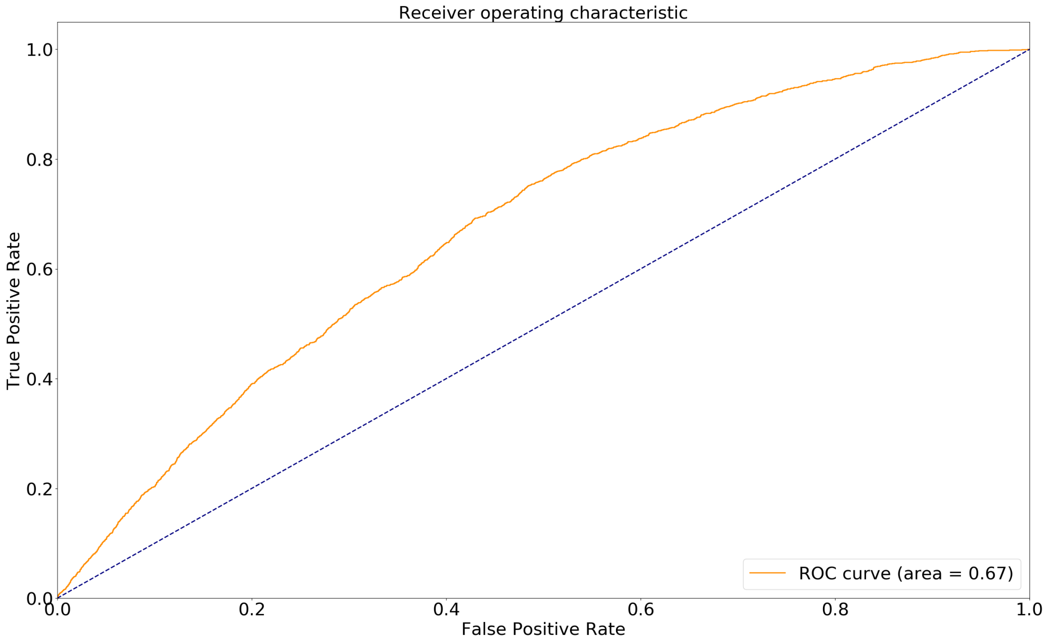

Figure 1: ROC: oversample

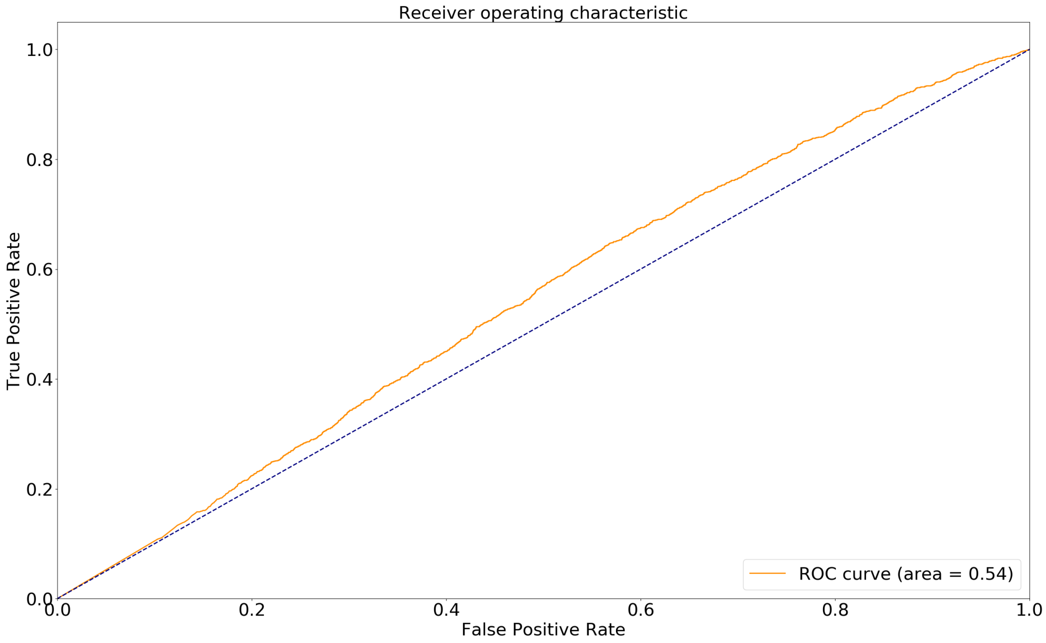

Figure 2: ROC: only applying under-sampling on the training data

Procedure of Neural Network(NN) analysis based on under-sampling

I followed the same procedure as the process in section , but it encounters one problem. It is mentioned that after integrating the training and testing data from the two groups, the oversampling method only applies on the training data. It causes the problem that the size of test data is way larger than the training set if we only fit the training set by using the under-sample technique. After re-sample the training data, the number of rows decreases from 836195 to 12404, while the test data has 209049 size. That means the training data is much less than the test data, it is very likely to have under-fitting problem. Although the mean cross validation score are around 93%, but ROC implied the model is not good enough for the testing data. The image 2 shows the ROC and it indicates the performance of our data based on the testing data is not very good. The figure 3 shows if we under-sample the training and testing data, we will get the overall accuracy as 97% and the ROC shows this model fits the testing data well. The cross validation got the score as 0.98549, 0.948408, 0.926642, 0.861749 and 0.86371.

However, I am not sure whether this makes sense, so I tried another method as well. This time I do under-sampling on the whole data, then split the balance data into training and testing sets. The cross validation score is 0.943168, 0.94075, 0.928658, 0.925836 and 0.927419. The image 4 is the ROC.

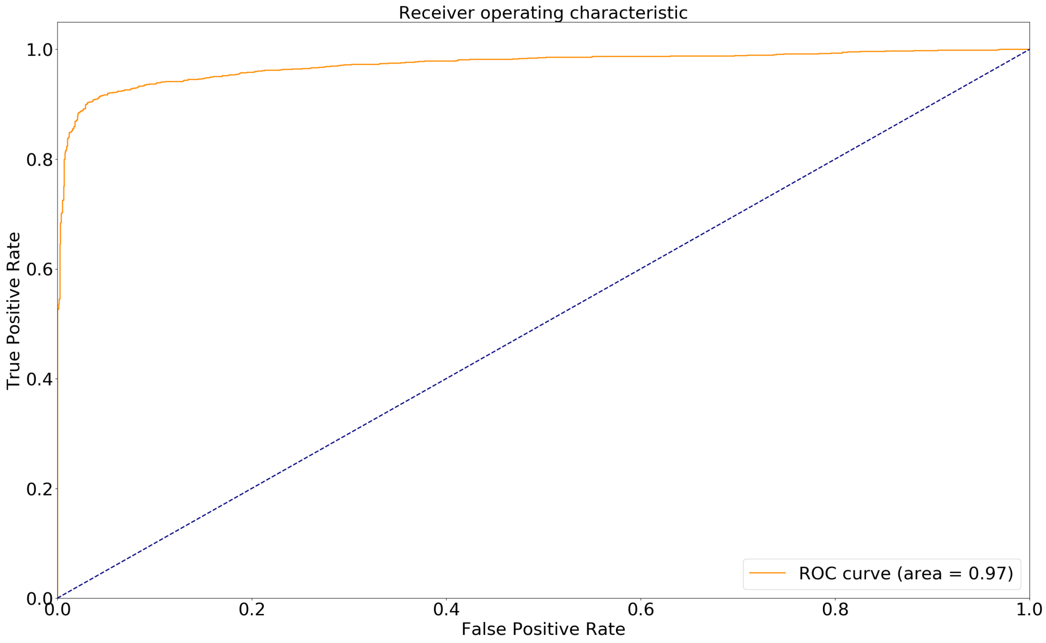

Figure 3: ROC: applying under-sampling on the training and testing data

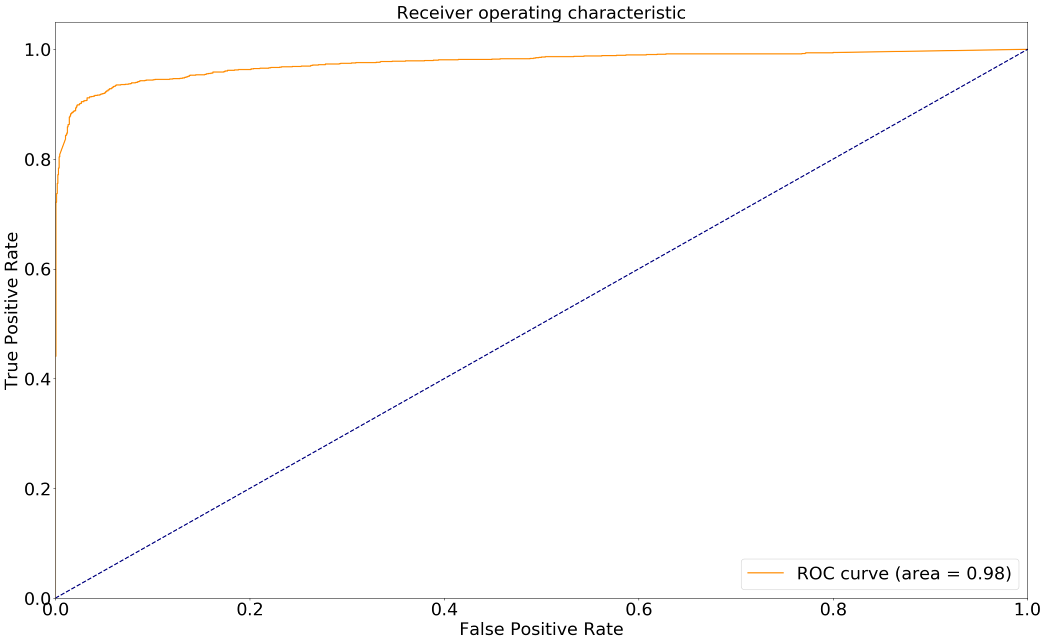

Figure 4: ROC: under sampling the data firstly

Questions

1. For the under-sampling method, which one makes more sense from the statistics? I don't think it is a good model if we only under sample the training data? Although the result is okay after the testing data also applied the under-sampling technique, I am not sure this makes sense. The second method that applies the under sampling firstly obtain the similar result like the first one in terms of the ROC and cross validation score. For me, the second one which do the under-sampling firstly make more sense for me.

2. For the over-sampling method, the model didn't fit well for our training data, but the cross-validation didn't indicate our model has over fitting problem, but how to improve the performance of the model? For each run, it takes at least 1.5 hours.