ECE 4950

Kaggle Competition Report

Yuqi Hong

May 8, 2019

Data Cleaning

- Training Dataset

There are 19 columns in original training dataset, first we should decide how much features we need to use for training. If we take a look at data for each features by using unique() method in pandas, we can find that for feature “ID” and “Summons Number”, all data of these features are unique, so training cannot learn any distribution from them, so I did not include them in training dataset.

Then I have to deal with null value in dataset. First, I found that for feature “Judgment Entry Date”, there were only 6974 non-null data, which was less than 10% of total number of data in training dataset, and training cannot learn much from this feature, so I did not include that. Then for null values for feature “county” and “issuing agency”, I filled them with the mode, because the number of mode appeared in the dataset was very high. However, for feature “violation” and “violation time”, the number of mode appeared in the dataset was not as high as “county” and “issuing agency”, so I have to delete these null values in these two features, but it did not affect much because there were only 10 null values in “violation” and 10 null values in “violation time”. Finally, there were 79989 rows * 19 columns non-null training data.

Since some features’ data types are not integer, I converted them to integer. I created a dictionary for each feature, the keys are unique strings in original dataset, and the corresponding values are integers start from 0. However, for feature “Issue Date” and “Violation Time”, because the format of data is time, so I separated them into parts: for “Issue Date”, I separated the date into year, month and day; for “Violation Time”, I separated the time into minute, hour, and AM/PM.

- Testing Dataset

For testing dataset, since we should not change the size of the dataset, so if there were null values in each feature, I just fill them with 0.

Then I used same idea of converting strings to integer for all features. Finally, there were 200000 rows * 19 columns non-null testing data.

Machine Learning Algorithms

- Decision Tree Classifer

I tried decision tree classifier for training, and after multiple testing by different parameters, I found the testing accuracy achieved maximum when maximum depth of tree was around 15, and the accuracy is around 83%.

- Bagging Classifier

I tried to add bagging classifier on decision tree classifier, so it became random forest classifier. I found the testing accuracy was improved and it can achieve 85% or higher if I set the number of estimators to be around 150.

- AdaBoost Classifier

I tried AdaBoost classifier on random forest, but the accuracy was not improved, even if I tried to increase the number of estimators, the performance was not as good as random forest.

- XGBoost Classifier

XGBoost is one of algorithms in boosting. It always performs better than other boosting classifiers. I tried this classifier and found that it did improve the accuracy compare with random forest and AdaBoost. The best accuracy that I can achieve was 85.548%, with maximum depth equals to 15, number of estimators equals to 41, and learning rate equals to 0.1.

- Other Classifiers

SVM classifier took very long time for training (around 30 minutes and had not converged yet), so the training is expensive and not a good classifier for this project.

Gaussian Naïve Bayes Classifier was fast, but accuracy was only around 60%.

K Neighbors Classifier can achieve accuracy around 80%, but is not as good as decision tree classifier.

Neural Network (MLP Classifier) can achieve accuracy around 83%, which is a little bit lower than decision tree classifier.

Result

The best accuracy I can achieve was 85.548%, and the classifier I used to achieve this accuracy was XGBoost classifier, with maximum depth equals to 15, number of estimators equals to 41, and learning rate equals to 0.1. All other parameters were default values.

Conclusion

In this competition, I have learned many data cleaning method, such as how to select useful features for training, how to deal with null values, and how to convert non-number data to integers or floats. I have also reviewed many classifiers I learned this semester, and also learned some new classifiers. Tuning parameters were the most time-consuming part, I tested a lot of parameters combination and finally found the parameters with best performance, but there might be parameters combination and also better classifiers with better performance, so further researches are needed to improve the performance.