ECE 4950 Project Report

Jie Huang(jh2674)

The goal of the project is to make predictions of traffic violation judgments based on the dataset we have. We have training dataset containing cases of traffic violations with 17 features in total. I decided to use randomforestclassifier first to solve the multi-classification problem because this model is easy to use and has good performance overall. What's more, we can check the feature importance after training. So, the very first step I do is to distinguish different types of data and it can be grouped into two categories in general. One is text and the other is number. Different type of data needs different data cleaning methods and techniques.

Before extracting some useful features from training the dataset, one thing we need to do is to check if the dataset has null. The key is to fill NaN values with some specific content or using certain methods to fill holes. According to statistical results, features like 'Violation', 'Issuing Agency', 'Precinct', 'License Type', and 'Violation Time' have very few null values, so I propagate non-null value forward and replace all NaN elements. I also tried 'bfill' method, but it has slightly influence on improving the prediction accuracy. Some attributes like 'County' and 'State', I filled holes with content 'UNKNOWN'. And for 'Judgment Entry Date', since most of the values are NaN, I simply replaced them with the specific date '1/1/1000'. Overall, these text features now have a contribution to model training.

The first data cleaning method I tried is to number the text data. In other words, we need to find all the unique texts within a feature of the dataset. For example, the attribute "License Type" has 60 different types and we can replace 60 different license type with numbers from 1 to 60 respectively. To transform the text into an index, I first built some dictionaries and map them with corresponding features. In this way, we can transform text data into integer index which can put into a machine learning model. Using this method, we can add more text features into the model but not all of them can improve the performance of the model.

The second technique I used is extracting components from the original data. This method is suitable to date and time since they have several components. For example, 'Issue Date' consists of three different information: day, month and year. Using the extracting method, we can divide the original information and acquire day, month and year separately. Similarly, time

information can be separated to hours and minutes. After several experiments, the year information of ‘Issue Date’ and ‘Judgment Entry Date’ can improve the prediction accuracy. Even though month information seems to have slightly contribution to accuracy, I try another way and transform it from one dimension to two dimensions. In other words, I calculated its sine and cosine values since the month variable is periodic and continuous. Unexpectedly, they have a contribution to the prediction accuracy in comparison to one-dimension month information. In addition, I also try to transform all the date information into the day of the week, but it makes the model somehow overfit and it failed to improve the prediction accuracy. For ‘Violation Time’, I tried to extract hour and minute, transform it to the total minute of a day, and calculate the time period of a day. But none of them can make a great improvement.

Next algorithm I have tried is transforming text data into its frequency or its appearance times. I applied this method to feature ‘Plate’. First, I calculate the frequency of each plate and built a dictionary for that. Then, I replaced all the text with its corresponding frequency. From the result, the model seems to have better performance with frequency information than with pure index.

As for some specific type of features like money and summon number, I basically put them directly into the model since numbers are quite effective and useful information to train a model. However, I had a further data process idea. What came to my mind is that I can make subtraction between features and create a new one. Take money data as an example, a key feature I discover is using ‘Interest Amount’ to subtract ‘Reduction Amount’ and a new feature ‘Interest VS Reduction’ came out. Surprisingly, the importance of the feature ranks the first among all the features which made great progress in improving the prediction accuracy. The public score raised from 0.85688 to 0.85791 which is a pretty great improvement comparing to the improvement of other methods. Therefore, I did the same way to ‘County’ with ‘State’ and create a new feature which contributes to better performance on prediction. Similarly, subtracting the year of ‘Issue Date’ and the year of ‘Judgment Entry Date’ also makes a slight improvement on accuracy.