Using machine learning techniques to analyse the dataset of traffic accidents in UK

Final Project Report - Group 1

Mengjie Zhang, Yilin Wang, Xining Wang

6202 Machine Learning I

George Washington University

Introduction

With fast developing urbanization during the past decade, automobile traffic volume is generally increasing in major cities. In the field of transportation planning, annual average daily traffic (AADT) is a measure widely used to present how busy the road is. AADT means the total volume of vehicle traffic of a highway or road for a year, or by time of day and day of week. In the United Kingdom AADT is a measure of traffic used by local highway authorities to forecast maintenance needs and expenditure.

One of the goals of government transportation departments is to respond to traffic accidents and reduce the occurrence of accidents by maintenance. So it is important to gain as much information from historical records of traffic accidents. The change of traffic volume, along with road conditions are also expected to affect the occurrence of accidents. This project aims to apply machine learning techniques to model the accidents occurrence and severity according to road conditions and traffic volumes (mainly AADT)

Description of Dataset

The data were downloaded from Kaggle, named “1.6 million UK traffic accidents”. It has more than 400,000 observations and over 30 features (location, accident severity, weather conditions and date, etc). Thus, we suppose that this dataset is large and good enough to train a deep network. The other dataset were also from Kaggle, it recorded the annual average daily traffic conditions in minor and major roads, including over 25 variables and 10,000 observations. The links for these two datasets are as below:

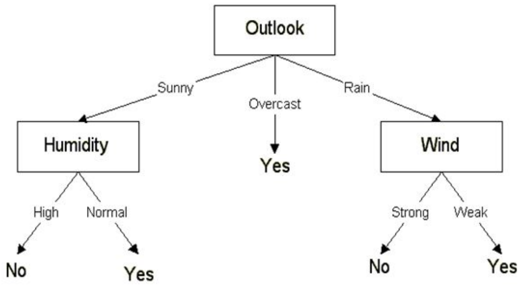
https://www.kaggle.com/yesterdog/eda-of-1-6-mil-traffic-accidents-in-london/data

https://www.kaggle.com/coolcoder001/predicting-car-count/data

Machine Learning Network

Random Forest

A Random Forest Classifier algorithm is used in this project to predict the severity of future traffic accidents given other predictive features including location, road surface condition, road level, etc. The Random Forest Algorithm is an ensemble method with Decision Trees as its base model. Below is a brief overview of the Decision Tree Classifier.



The graph above shows a simple set up for a Decision Tree Classifier. This algorithm would simply keep asking questions that would make the best split of the training dataset, until the purity of labels in the child nodes reach a preset threshold, or other early stopping criteria is triggered.

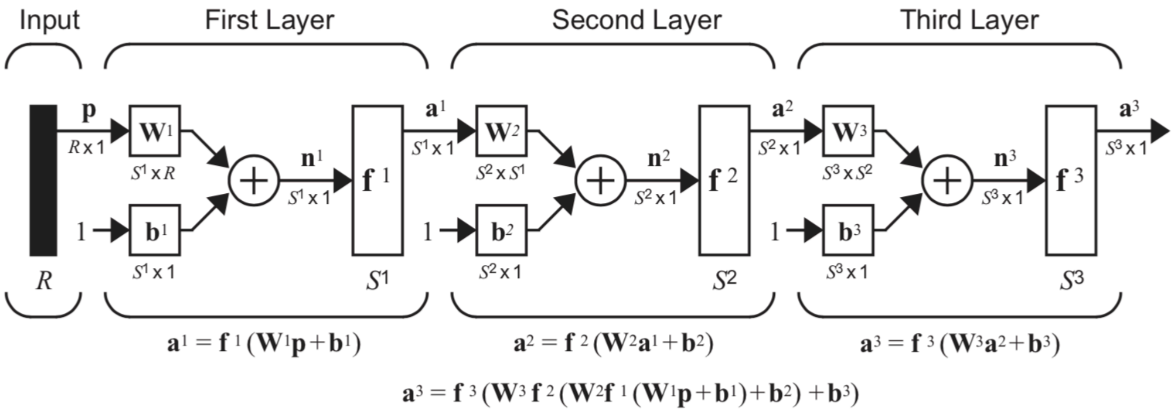
As simple as this algorithm is, it is considered powerful since as the tree grows deeper, less datapoint would remain in a single leaf node, and would eventually achieve 100% purity in the leaf nodes (Number of leaf nodes equals number of data points in extreme situations). Because of this reason, it is almost certain that a single Decision Tree Classifier would overfit to the training data without any pruning.

In order to fight overfitting in Decision Tree models, we would use a statistical method called Bootstrap Aggregating (Bagging). In this sampling method we would draw S samples of M data points from N records with replacement, and build Decision Tree Classifiers on every samples, in this way we are able to bring in new randomness into the model, which is believed to make it generalize better.

In the meantime, the Random Forest algorithm would introduce random feature selections on every Bootstrap samples we introduced above, which would reduce overfitting to a higher extend and result in a more robust model. A binary Random Forest Classifier will be used in our project predicting severity of traffic accidents.

Neural Network

Neural Network used mathematical equations to build up useful relations of inputs and outputs via several processes. We chose Multilayer perceptron to conduct our data analysis and prediction. The structure of multilayer perceptron includes inputs, then one or more hidden layers and the output layer, each of them has multiple neurons.



Naive Bayes

Based on Bayes theorem, Naive Bayes classifier used conditional independence to calculate the posteriors, one values on the condition of other attributes happens. It has drawback that the predictive accuracy and the assumption of class conditional independence are highly interrelated. However, in the real world, variables could be depended on each other.

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

Support Vector Machine (SVM)

One of the most popular machine learning algorithms is Support Vector Machine (SVM). Unconstrained optimization problems were solved by SVM classifier with it loss function. Support vectors are inputs which were the nearest to the optimal decision hyperplane. It classifies data points into the same categories and find out the robust decision boundary, even they are not linearly separable.

$$\min_{x} \frac{1}{2}w^T w + C \sum_{i=1}^{n} \xi(w; x_i, y_i)$$

Experimental Setup

Before building the machine learning network, we first preprocessed the data set. We dropped unrelevant columns and extracted relevant features from the originial data, shown in Table 1. Accident_Severity is the target and has three levels 1, 2, 3. Severity level 1 and 2 are combined as non-severe accidents and labelled 0, and severity level 3 as severe accidents were labelled 1. We also encoded categorical features using One Hot Encoding.

Then we implemented functions get_performance_metrics, plot_roc_curve, train_model and train_model_svc to make training algorithms easier later as well as monitor performance of trained algorithms.

Finally we built machine learning classifiers using random forest, neural network, Naive Bayes and support vector machine. We trained the models and displayed performance metrics and ROC plot by calling the functions.

Table 1. Extracted feature names and data type.

Variable name	Data type
Accident_Severity	Integer
Day_of_Week	Integer
Road_Type	Object
Speed_limit	Integer
Pedestrian_Crossing-Human_Control	Object
Pedestrian_Crossing-Physical_Facilities	Object
Light_Conditions	Object
Weather_Conditions	Object
Road_Surface_Conditions	Object
Special_Conditions_at_Site	Object
Carriageway_Hazards	Object
Urban_or_Rural_Area	Integer
Did_Police_Officer_Attend_Scene_of_Accident	Object
Hour	Integer
Peak	Integer

Results

I. Random Forest

We implemented a Random Forest Classifier with n_estimators=100. Random Forest tend to overfit as we didn't limit the maximum depth. But we limited the minimum leaf node and a larger number for number of trees to fight overfitting. This is the model we found using Grid-Search that achieved the best testing AUC. The AUC is 0.60, while the accuracy is 84%. The accuracy inflated because of the imbalance in our dataset. Judging from the ROC curve, we can see that the model performs better on the positive side compared to the negative side, it's because we have more positive data.

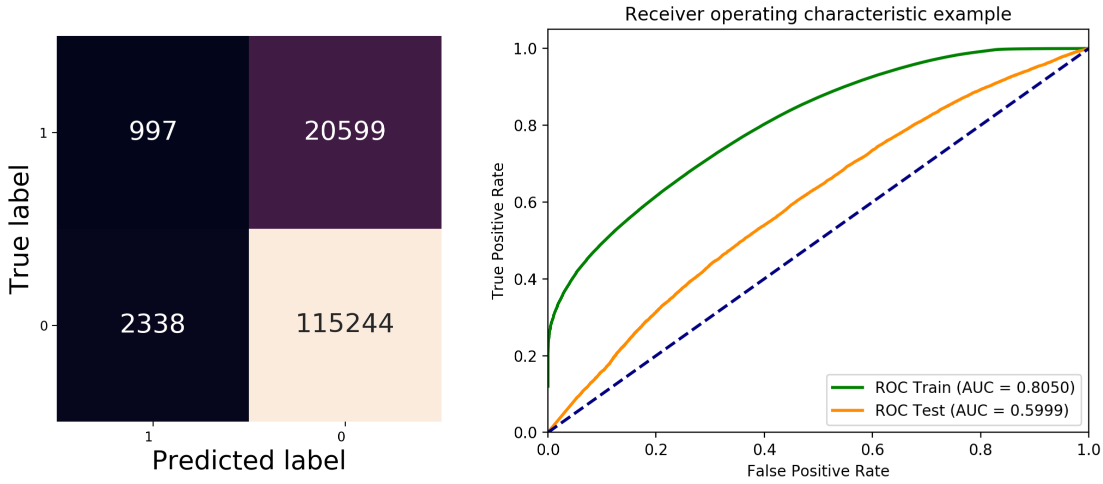


Fig 1. Confusion matrix and ROC plot of Random forest.

II. Neural Network

The multilayer perceptron performs the best when considering overfitting. There's almost no overfitting in this model, but the AUC performance is significantly lower than the result from the Random Forest Classifier even though the accuracy are identical because of the imbalance in our dataset.

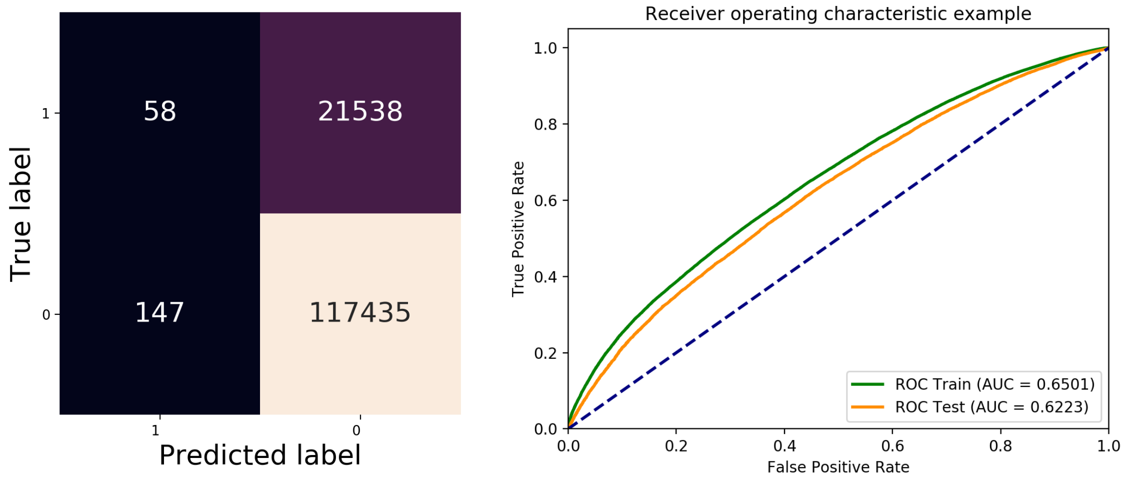


Fig 2. Confusion matrix and ROC plot of neural network.

III. Naive Bayes

The performance metrics of Naive Bayes have no difference between training and test sets. The AUC and model accuracy for test set are 0.61 and 0.51 respectively. The precision is

0.78, while recall is only 0.51. The classification report shows that class 1 shows higher precision (0.89) than class 0 (0.19) but lower recall.

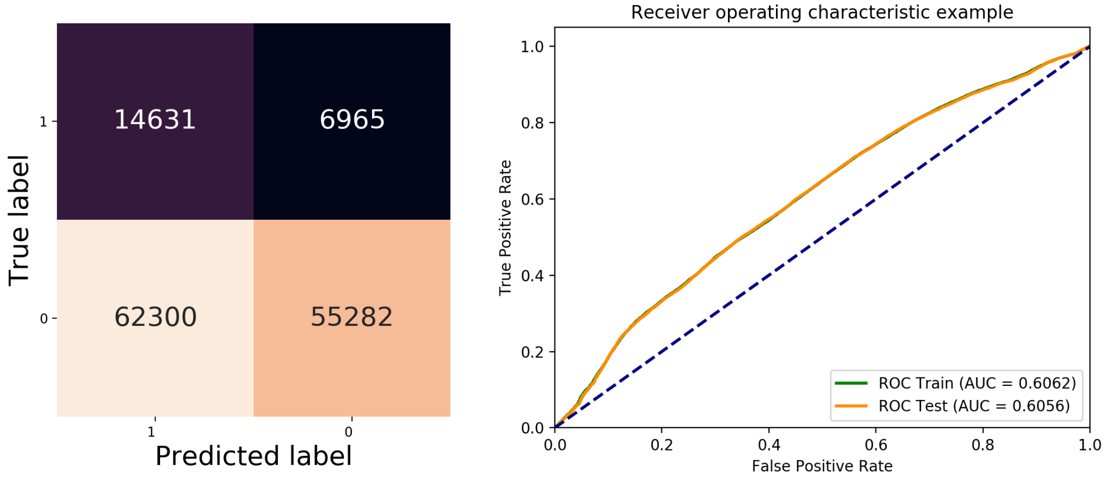


Fig 3. Confusion matrix and ROC plot of Naive Bayes.

IV. Support Vector Machine

The performance metrics of linear SVC have no difference between training and test sets. The AUC and model accuracy for test set are 0.61 and 0.84, respectively. The linear SVC has better performance than Naive Bayes, considering its high precision and recall (0.75 and 0.84 respectively). The classification report shows that precision and recall both are very low for class 0 (0.25 and 0.01 respectively), although they are very high for class 1 (0.85 and 1).

Table 2. The classification report for linear SVC

Class	Precision	Recall	F1-score	support
0	0.25	0.01	0.01	21596
1	0.85	1.00	0.91	117582
Avg/total	0.75	0.84	0.77	13978

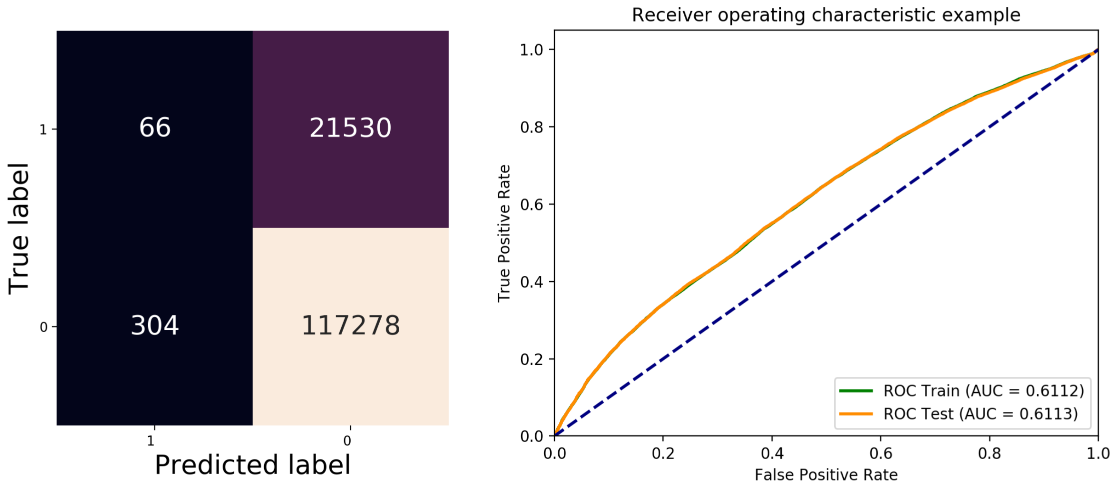


Fig 4. Confusion matrix and ROC plot of support vector machine.

Summary & Conclusion

We have implemented four machine learning methods (random forest, neural networks, Naive Bayes and support vector machine) on the accident dataset with 14 features and >400,000 observations. By comparing the performance metrics, the AUC and accuracy of random forest, neural networks and support vector machine are not significantly different, while the accuracy of Naive Bayes is significantly lower than others. The recall of Naive Bayes is also significantly lower than other, while precision of all methods are close. Moreover, by comparing performance metrics of two target classes (0 and 1), we have found that precision and recall of class 0 are always much lower than the metrics of class 1 (Table 3). This indicates that the models have low sensitivity for class 0. In this data set,the observations of class 1 outnumbered the observation of class 0. The imbalance of the data could negatively affect the model performance.

In conclusion, for the accident dataset, random forest, neural network (MLP), and support vector machine (linear SVC) shows better performance than Naive Bayes, although future work is needed to improve the AUC and sensitivity of class 0 (nonsevere accidents).

Table 3. The summary of performance metrics for all machine learning methods

Method	AUC	Accuracy	Precision	Recall	F1-score
Random forest	0.60	0.84	0.76	0.84	0.78
Neural network	0.62	0.85	0.76	0.84	0.77
Naïve Bayes	0.60	0.51	0.78	0.51	0.57
SVM	0.62	0.84	0.75	0.84	0.77

References

Hoang Nguyen , Chen Cai, Fang Chen(2017). Automatic classification of traffic incidents' severity using machine learning approaches. *IET Intell. Transp. Syst.*, Vol. 11 Iss. 10, pp. 615-623.

UK Department of transport https://www.dft.gov.uk/traffic-counts/index.php
Kaggle https://www.kaggle.com/daveianhickey/2000-16-traffic-flow-england-scotland-wales
Neural Network Design (2nd Ed), by Martin T Hagan