PREDICTION OF SEVERITY OF TRAFFIC ACCIDENTS BY USING MACHINE LEARNING CLASSIFICATION ALGORITHMS

Furkan ÇAKMAK,Tarık Can ŞAHİN, İlker BEDİR
Bilgisayar Mühendisliği Bölümü
Yıldız Teknik Üniversitesi, 34220 Istanbul, Türkiye
{fcakmak, l1117090, l1116036}@yildiz.edu.tr

Abstract

Nowadays, with the increase in the number of vehicles, traffic accidents can occur almost every day. These accidents can primarily affect the vehicle owners or victims of the accident both financially and mentally. In addition, traffic accidents can cause disruptions in terms of time in the daily activities of the drivers who are not directly affected by the accident or in their reach to another place. With this project, the effect of a traffic accident on traffic in a region will be analyzed, and in the event of an accident in that region, its effect on traffic flow can be predicted in degrees.

Keywords: Traffic Accidents, severity, prediction, random forest, machine learning, naive bayes, decision tree, classification

I. Introduction

Innovations and developments all over the world directly affect freight and passenger mobility. Factors such as the improvement of the economic situation of people, the increasing trade volume, the increase and acceleration of the circulation of goods, the improvement of road and vehicle quality, the change of direction of transportation modes and the people offering new alternatives also affect the transportation systems and security. The increase in the number of vehicles registered in traffic, the increase in the population, the increase in the number of people with driving licenses, the speed and competition in the transportation sector, the trend of people to a more mobile business and lifestyle compared to previous years, the traffic density increases and this situation is also reflected in traffic safety . Unfortunately, according to the figures of the World Health Organization, more than one million people die as a result of a traffic accident every year. When the causes of death of people in the world are examined, deaths due to traffic accidents continue to rise to the top every year. By 2030, deaths due to traffic accidents are predicted to rank fifth among the causes of death. In addition, the rate of youth among those who died as a result of a traffic accident is quite high. Considering the injuries and injuries and the economic losses incurred in addition to the deaths, it becomes clear how important and indispensable it is to prevent traffic accidents.

Since it is a functional language for the project, "Python" programming language was used. For graphs and statistical data, Python's "matploitlib" and "numpy" libraries were used and many statistical prediction data were examined. According to these examinations the most appropriate machine learning algorithm for the most accurate prediction was explained by detailed comparisons. In the realization phase of the project, "datetime" library has been used to calculate the working time of the algorithm also to determine the algorithm that can produce the fastest solution and inform the user.

II. Material and Methods

A. Data set Information

We used only one data set for our work and it can be reached from a web site which has more than 1 million data sets in itself [1].

1) The Traffic Accident Data set: In this study, we used a data set which has more than 3.5 million traffic accident records. Data set columns contains 49 features which have been classified as weather condition, (temperature, humidity, etc.) road information (roundabout, traffic signs, etc.), accident duration, severity, for every traffic accident record. During all studies, severity was a target column for us and we developed a model which can predict severity by examining its relationships with other features.

B. Decision Tree Algorithm Module

[2] A decision tree is a flowchart-like tree structure where an internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the attribute value. It partitions the tree in recursively manner call recursive partitioning. This flowchart-like structure helps you in decision making. It's visualization like a flowchart diagram which easily mimics the human level thinking. That is why decision trees are easy to understand and interpret.

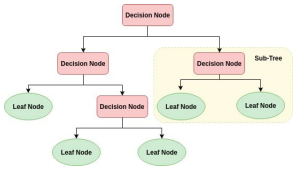


Figure 1 Decision Tree Nodes and Attributes

The basic idea behind any decision tree algorithm is as follows:

- Select the best attribute using Attribute Selection Measures(ASM) to split the records. Make that attribute a decision node and breaks the data set into smaller subsets.
- Starts tree building by repeating this process recursively for each child until one of the condition will match:
- All the tuples belong to the same attribute value.
- There are no more remaining attributes.
- There are no more instances.

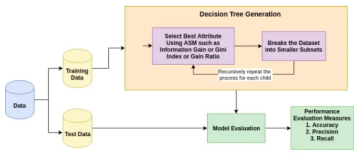


Figure 2 Decision Tree Algorithm Steps

Decision Tree Algorithm is fast also trains classification models with a high accuracy score. So this algorithm is our most important target while predicting our model.

C. Random Forest Algorithm Module

[3] It technically is an ensemble method (based on the divide-and-conquer approach) of decision trees generated on a randomly split data set. This collection of decision tree classifiers is also known as the forest. The individual decision trees are generated using an attribute selection indicator such as information gain, gain ratio, and Gini index for each attribute. Each tree depends on an independent random sample. In a classification problem, each tree votes and the most popular class is chosen as the final result. In the case of regression, the average of all the tree outputs is considered as the final result. It is simpler and more powerful compared to the other non-linear classification algorithms. Here we can see the steps of Random Forest Algorithm while its working.

- Select random samples from a given data set.
- Construct a decision tree for each sample and get a prediction result from each decision tree.
- Perform a vote for each predicted result.
- Select the prediction result with the most votes as the final prediction.

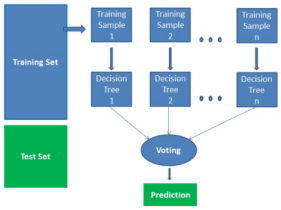


Figure 3 Random Forest Algorithm Steps

D. Naive Bayes Algorithm Module

[4] The basic logic behind Naive Bayes Algorithm is to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction. There are 3 types of it.

1) Gaussian Naive Bayes: It is used in classification and it assumes that features follow a normal distribution.

2) Multinomial Naive Bayes: It is used for discrete counts. Mostly used for text classification problems. For example Spam E-mail Prediction.

3) Bernoulli Naive Bayes: The binomial model is useful if your feature vectors are binary (i.e. zeros and ones).

In here Naive Bayes Algorithm's advantages can be seen directly:

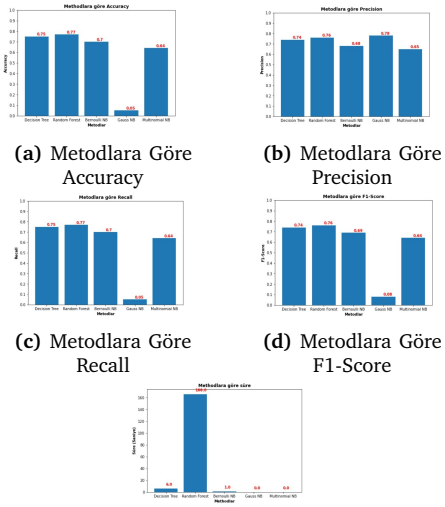
- It is easy and fast to predict class of test data set. It also perform well in multi class prediction.
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

In this article we focused mostly Decision Tree Algorithm and implemented various techniques to improve that algorithm for our prediction module. Also discussed other 4 algorithms against Decision Tree Algorithm's performance.

E. Proposed Approach

There are lots of records in this data set. And time is important factor for this model. In Figure 4 it can be clearly seen that most effective algorithm by time factor and accuracy rate is Decision Tree Algorithm. But data set is not arranged yet, just dropped rows which have null values in themselves.

It's decided to increase these accuracy scores by optimizing data set and algorithm modules. First, null values are filled with mean calculation of their columns. After this method accuracy rate is increased from 0.75 to 0.77 for Decision Tree Algorithm and from 0.77 to 0.79 for Random Forest Algorithm. Finally MICE model is used in modules. With this module, it's reached to best and highest accuracy score values for each algorithm.



(a) Metodlara Göre Accuracy

(b) Metodlara Göre Precision

(c) Metodlara Göre Recall

(d) Metodlara Göre F1-Score

(e) Metodlara Göre Süre

Figure 4 Test boyutu = 0.2 algoritmaların "Accuracy", "f-1 score" ve "Precision" değerleri ve hız verisi

III. Experimental Results

The proposed approach was evaluated in the detection performance by MICE phase.

A. The Detection Performance

[5] There are a variety of multiple imputation algorithms and implementations available. The most popular algorithm is called MICE. Here are steps for filling null values in data set:

- A simple imputation, such as imputing the mean, is performed for every missing value in the data set. These mean imputations can be thought of as “place holders.”
- The “place holder” mean imputations for one variable (“var”) are set back to missing.
- The observed values from the variable “var” in Step 2 are regressed on the other variables in the imputation model, which may or may not consist of all of the variables in the data set. In other words, “var” is the dependent variable in a regression model and all the other variables are independent variables in the regression model.
- The missing values for “var” are then replaced with predictions (imputations) from the regression model. When “var” is subsequently used as an independent variable in the regression models for other variables, both the observed and these imputed values will be used.
- Steps 2–4 are then repeated for each variable that has missing data. The cycling through each of the variables constitutes one iteration or “cycle.” At the end of one cycle all of the missing values have been replaced with predictions from regressions that reflect the relationships observed in the data.
- Steps 2 through 4 are repeated for a number of cycles, with the imputations being updated at each cycle. At the end of these cycles the final imputations are retained, resulting in one imputed data set. Generally, ten cycles are performed; however, research is needed to identify the optimal number of cycles when imputing data under different conditions. The idea is that by the end of the cycles the distribution of the parameters governing the imputations (e.g., the coefficients in the regression models) should have converged in the sense of becoming stable.

Also it’s generated graph by training the data set for Decision Tree Algorithm to find optimum max depth parameter.

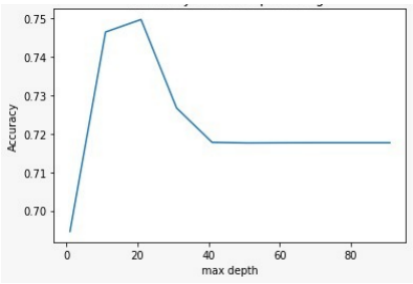


Figure 5 Max Depth-Accuracy Relationship for Decision Tree Algorithm

B. The Classification Performance

Finally nearly %20 of 3.5 million rows prepared for the test, the accuracy values of the 3 algorithms are added in Table-1. Also information of confusion matrix for top 3 algorithm can be seen in Figure 6

	ALGORITHMS USED FOR TEST		
	Decision Tree Algorithm	Random Forest Algorithm	(Bernoulli) Naive Bayes Algorithm
Overall	**0.83**	**0.84**	**0.64**

Table 1 Top Algorithms Accuracy Table

(a) Decision Tree Algorithm

(b) Random Forest Algorithm

(c) Bernoulli Naive - Bayes Algorithm

(d) Gaussian Naive - Bayes Algorithm

Figure 6 Test size = 0.2 Confusion Matrix of Algorithms which are integrated by MICE model

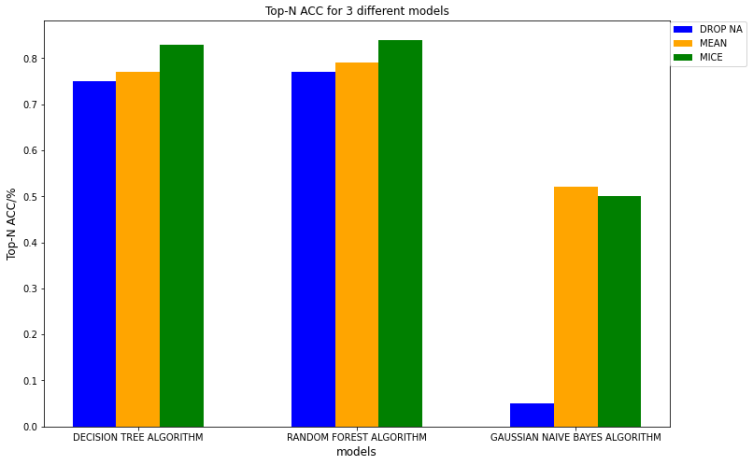


Figure 7 Bar Graph for top 3 algorithm

According to Figure 6 and Table 1, it can be clearly seen that, accuracy scores are increased to %83 for Decision Tree Algorithm, to %84 for Random Forest Algorithm, to %64 for Bernoulli Naive Bayes and to %50 for Gaussian Naive Bayes by optimizing the data set with MICE model.

IV. Conclusion

Following the performance improvements mentioned above, there have been noticeable improvements in accuracy and time. Especially the Decision Tree Algorithm has been highly efficient. While there was a predictive ability of %75 in the first steps, it became a model that rose up to %83 with improvements. Along with these, the accuracy value of the Random Forest Algorithm, which is a little weak in terms of time, approached from %77 to %84. However, since the speed of education is as important as efficiency, it has been observed that Random Forest Algorithm is rather slow in training and testing compared to Decision Tree Algorithm. For the Bernoulli Naive Bayes Algorithm, while it had an accuracy value of %70 in the first cases, the ability to predict with %64 accuracy decreased with the optimizations in eliminating the lost data in the data set. Another algorithm that has changed significantly is the Gaussian Naive Bayes Algorithm. In the data set in which rows with "NaN" values were dropped, it was at a low rate with %0.05 accuracy, after filling these data with MICE method, the accuracy rate increased to %0.5. Although there is a low estimation rate for this model, it has achieved the most increase with a noticeable increase.

For this model developed, it was decided that the most suitable and fastest working algorithm for prediction in the system was the Decision Tree Algorithm, and it was decided to predict the probable accident severity by using the Decision Tree Algorithm of the accident data entered from the user for estimation.

References

[1] S. Moosavi, "Us accidents (3.5 million records) a countrywide traffic accident dataset (2016 - 2020)," vol. 26, 2020.

[2] A. Navlani, "Decision tree classification in python," *26*, 2018.

[3] ——, "Understanding random forests classifiers in python," vol. 26, 2018.

[4] S. Ray, "6 easy steps to learn naive bayes algorithm with codes in python and r," vol. 26, 2017.

[5] A. Bilogur, "Mice," *Simple Techniques for missing data imputation*, vol. 26, 2018.