Stats 101C Final Project

Predictive Analysis of Car Accident Severity

Group O

David Chabra, Kai-Li (Kelly) Chang, Harry Kim, Chan-Wai Wong

Abstract

The goal of this project was to predict car accident severity using statistical classification methods. This report provides a summary of how we developed and applied our final model as well as the limitations of the project and the conclusions we came to. The data for this project was a limited selection from a Countrywide Traffic Accident Dataset (2016 - 2021) provided by professor Akram Almohalwas.

Our final model uses many of the provided futures as well as many futures derived from the provided data, we also combined this with other national data sets. This resulted in our final random forest model with a prediction accuracy of 94.25% on testing data.

1. Introduction

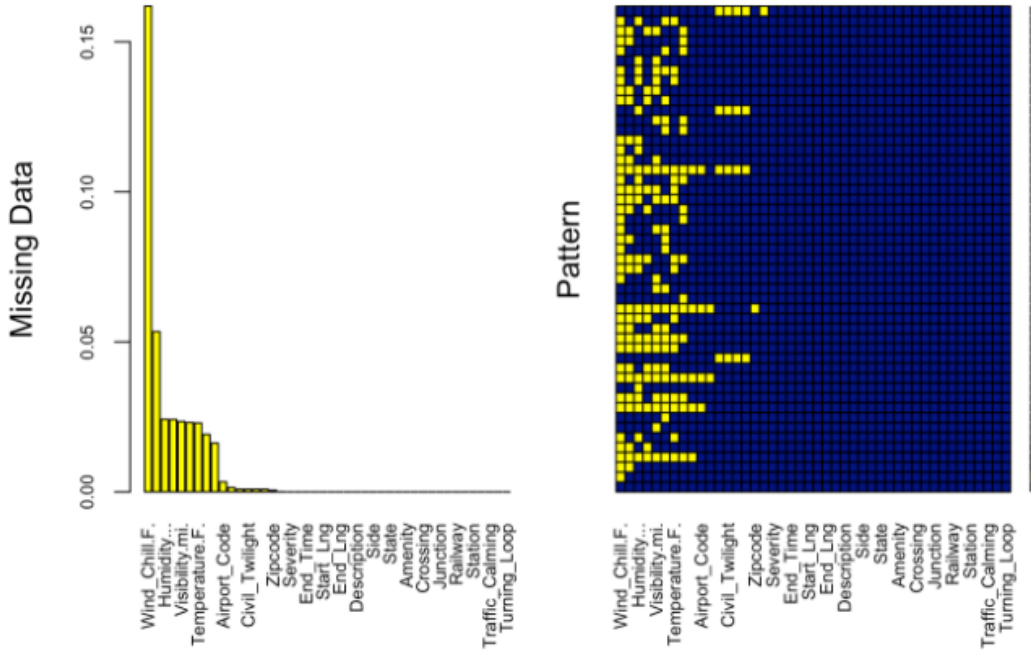
More than 90 people die in car accidents every day and around 2 million drivers in car accidents experience permanent injuries every year. (Predicting Car Accidents' Severity | Kaggle) The U.S. is highly dependent on cars for travel so understanding and preventing car accidents is essential. For this project, we were given two data sets on car accidents, our training data with 35,000 observations and testing data with 15,000 observations. In both datasets, the rows represent observations, and the columns represent predictors. Our goal was to predict accident severity which is either mild or severe. Since our testing data was used to evaluate our model it did not contain the severity variable. The features of the accident given for our project came in a lot of different forms. We have the description of the accident as well as information on the timeframe, location, and other information about the accident. This gave us a lot of room to change our data to make it more suitable for our analysis. We also had the option to join external data sets since we had the location data, for example, we could attach other information related to the county or zip code of the car accident.

2. Data Analysis

Our data is a limited selection from the countrywide traffic accident dataset from 2016 to 2021. Two datasets were provided for the project, the training data, “Acctrain.csv”, which had 35,000 observations and 44 columns, and the testing data, “AcctestNoYNew.csv”, which had 15,000 observations and 43 columns. In both datasets, the unit of observation is a single car accident, in which each row contains information about the incident, such as time, location, description, and other factors. Unfortunately, the datasets contained several missing values. For instance, the training data had a total of 13,211 missing values and the testing data had 5,842

missing values. As the presence of missing values would cause problems in model fitting, the NA values had to be removed.

Because using the na.omit() function would delete up to 13,211 and 5,842 observations in training and testing data respectively, we looked for other ways of removing the missing values. We decided to look at the distribution of the missing values in each predictor.



We found that the predictors “Wind_Chill.F.” and “Wind_Speed.mph.” had a significantly high number of missing values, with “Wind_Chill.F.” having 5,666 missing values in training data, 2,485 missing values in testing, and “Wind_Speed.mph.” having 1,870 missing values in training, 843 missing values in testing. Since the missing values from the two predictors were more than 5% of the number of observations of each dataset, we decided that imputing the missing values of the two predictors could cause misleading results, so they were removed. For the rest of the missing values in the dataset, we used the mice() function to impute the values.

Additionally, some of the predictors that were insignificant, uncorrelated, redundant, used to extract data, or unusable, such as the "Country" predictor that had a single factor of "US", were removed.

3. Text mining

To optimize the description predictor, we used a text-mining technique to find significance in the description predictor and turn it into more accessible data. Since the values of the description predictor were a block of text, we had to transform the data for it to be used in modeling. To achieve that we wrote two separate files containing only the description column based on the severity (mild and severe), and then we used Voyant Tool (https://voyant-tools.org/) to do text mining.

We found that there were two keywords in the descriptions of severe accidents - accident, and incident. According to Oxford Languages, "accident" means "an unfortunate incident that happens unexpectedly and unintentionally, typically resulting in damage or injury,", and incident means "an event or occurrence," (Oxford Languages). So we thought these two words could help us identify car accident severity effectively. However, we found that the most frequent word in the mild cases was also "accident", therefore just using the presence of the word "accident" and "incident" as a predictor was not enough.

After closely examining the data, we found that, in severe cases, those which contained "incidents" were usually followed by the phrase "road closed."

In addition, in the mild cases, those which contained “accidents” usually had “stationary traffic” and “slow traffic” in the description. The other phrase that only appeared in the mild cases was “with caution”

4. Data transformation

To transform our data to make it suitable for model fitting, we created several extra predictors based on four given predictors- Start_Time and End_Time, Description, Weather_Condition, and Sylcount description, and additionally, we imported and merged the census data which had information such as population density. The data in both training and testing contained predictors that had values that could not be used as quantitative or qualitative

data or had too many factors to be viable for model fitting. So we used techniques such as text mining, and string detect (str_detect) to transform our data

A. Time Data:

The two given time predictors “Start_Time” and “End_End” were not useable in their raw character form and thus had to be transformed using as_datetime(), then from the transformed time data, we looked for any significance that could help our model.

The pandemic that began in 2020 that is still continuing caused the amount of traffic to go down significantly between 2020 and 2021, and hence there were fewer car accidents reported in 2020 and 2021 than in the previous years. So based on the years, we created a “covid_year” predictor for when the year of the accident was 2020 or 2021.

Additionally, during the holiday seasons in November and December, we speculated that people would have traveled more for Thanksgiving, Christmas, and New Year’s Eve, which relates to a high car accident rate. Therefore, we included another boolean predictor “holiday” that identifies whether the timestamp of the accident was during the holiday season.

Lastly, we noticed that the duration of the accident (time-end - time-start) could be related to the severity of the accident, so we added another quantitative “duration” predictor.

B. Description:

We created several predictors based on the result from text mining the description data. Based on our text mining results we added boolean predictors “uses_incident” and “uses_accident” on whether the word “incident” or “accident” was used in the description data. To improve the accuracy of our model we also added predictors such as “uses_with_caution,” “uses_road closed”, “uses_stationary_traffic”, “uses_slow_traffic”, and more.

C. Weather Condition:

The “Weather_Condition” predictor is qualitative character data that had the weather condition of when the accident happened. But since the predictor had 70 different weather conditions, and therefore 70 levels of factors, there was a need for formatting in order for the data to be used for modeling. In order to reduce the factors, we created boolean predictors for most of the words that were in the “Weather_Conditoin” data. For instance, a weather condition of “heavy rain” would have the value TRUE for “weather_heavy” and “weather_rain” predictors. But one thing we noticed was that some weather conditions were not present in either mild or severe accidents, so we simply removed them.

D. Sylcount description

We used the readability function from the “sylcount” library to generate many new predictors from our description predictor. The new predictors were the number of characters in the description, the number of word characters, the number of words, the number of nonwords, the number of sentences, the number of syllables, and the number of polysyllables (in this case words with 3+ syllables). It then uses these features to calculate various reading ease indices. These are the Flesch reading ease score, Flesch-Kincaid grade level score, Automatic Readability Index score, Simple Measure of Gobbledygook (SMOG) score, and the Coleman-Liau Index score. Several of these generate infinite values for a blank description so they were not used in our final model.

E. Census data

We imported outside data, *US Zip Codes Points- United States of America* (https://data.opendatasoft.com/explore/dataset/georef-united-states-of-america-zc-point%40public/table/), and merged the data based on the zipcodes to obtain population and density data. The zip code data in its raw form could not be used as a quantitative predictor and had too many factors (15,863 levels). To make use of the zip code data, we merged the *US Zip Codes Points* data which had information such as city name, ZCTA, population, density, timezone, geopoint, and more. But from the many variables, we chose population and density to be used as a predictor in our model.

5. Methods and Models

A. Principal Component Analysis (PCA):

PCA is a dimension reduction method that is often used to reduce the large size of the data set. It helps to find the correlation within each predictor and removes the uncorrelated features. We expected that this step would help us to speed up the algorithm performance and reduce overfitting by reducing 73 predictors when we fit models.

	PC1	**PC2**	**PC3**	**PC4**	**PC5**	**PC6**	**PC7**
Standard deviation	2.305	1.6507	1.5146	1.3346	1.1701	1.1267	1.0016
Proportion of Variance	0.253	0.1298	0.1092	0.0848	0.0605	0.0605	0.0478
Cumulative Proportion	0.253	0.3828	0.4920	0.5768	0.7025	0.7025	0.7503
	PC8	**PC9**	**PC10**	**PC11**	**PC12**	**PC13**	**PC14**
Standard deviation	0.9912	0.9164	0.8863	0.8637	0.8317	0.7775	0.6570

Proportion of Variance	0.0468	0.0400	0.0374	0.0355	0.0329	0.0288	0.0206
Cumulative Proportion	0.7971	0.8370	0.8745	0.9100	0.9429	0.9717	0.9923
	PC15	**PC16**	**PC17**	**PC18**	**PC19**	**PC20**	**PC21**
Standard deviation	0.3026	0.2204	0.1181	0.0870	0.0325	0.0022	0.0009
Proportion of Variance	0.0044	0.0023	0.0007	0.0004	0.0001	0.0000	0.0000
Cumulative Proportion	0.9966	0.9989	0.9996	1.0000	1.0000	1.0000	1.0000

Figure 5.1: A summary table for PCA model

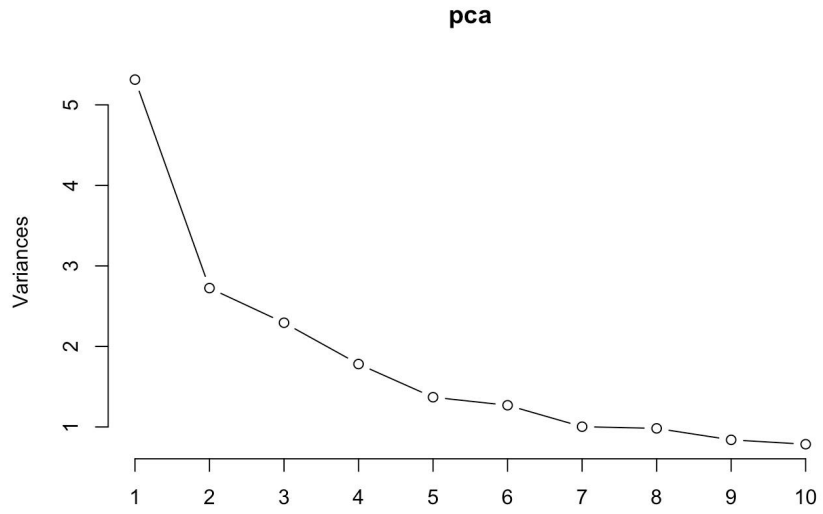


Figure 5.2: A scree plot of PCA model

We scaled all numeric predictions and only applied them for the PCA model in order to reduce the number of predictors. Our purpose was to select the best number of components that could explain about 80% of the data and then combine them with other character predictors for fitting classification models. Based on the PCA summary table, we observed that 9 principle components can explain about 83.7% of the data. However, we did not know which principal

component was correlated to each numerical predictor. The training accuracy we got by just using the numerical predictors was 89.92%.

B. Logistic Regression

Logistic regression is a straightforward but incredibly effective approach for solving binary classification problems. It is easy to implement, interpret, and efficient to train the model. We used the predictors we got from the feature selection to predict the accuracy. The training accuracy rate we got was 93.12%.

C. Random Forest

Random Forest is a supervised machine-learning technique that can be used for both classification and regression problems. It is easier to perform since scaling and standardizing is not necessary. It helps to reduce overfitting within each decision tree. So, we used it to perform feature selection instead of using PCA. We applied randomForest() three times and obtained our best subset of predictors using Variable Importance Plots.

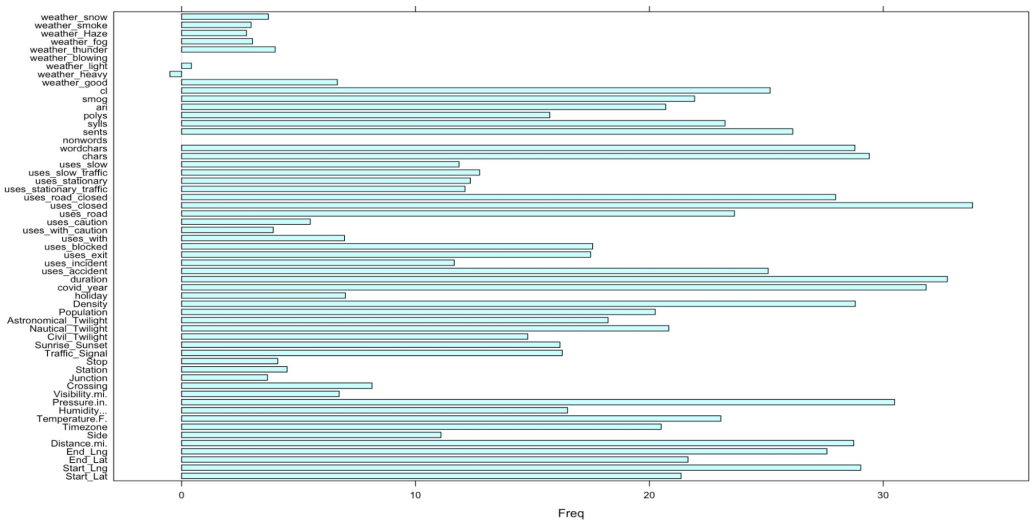


Figure 5.3: Final predictors for MeanDecreaseAccuracy

In the process, we removed the variables with negative or low MeanDecreaseAccuracy. By the end of the process, we reduced the number of predictors from 73 to 58. Finally, we sorted the mean decrease accuracy of the predictors in descending order to find the 10 most informative predictors which were, used_closed, duration, covid_year, Pressure.in., chars, Start_Lng, Density, wordchars, Distance.mi., and uses_road_closed.

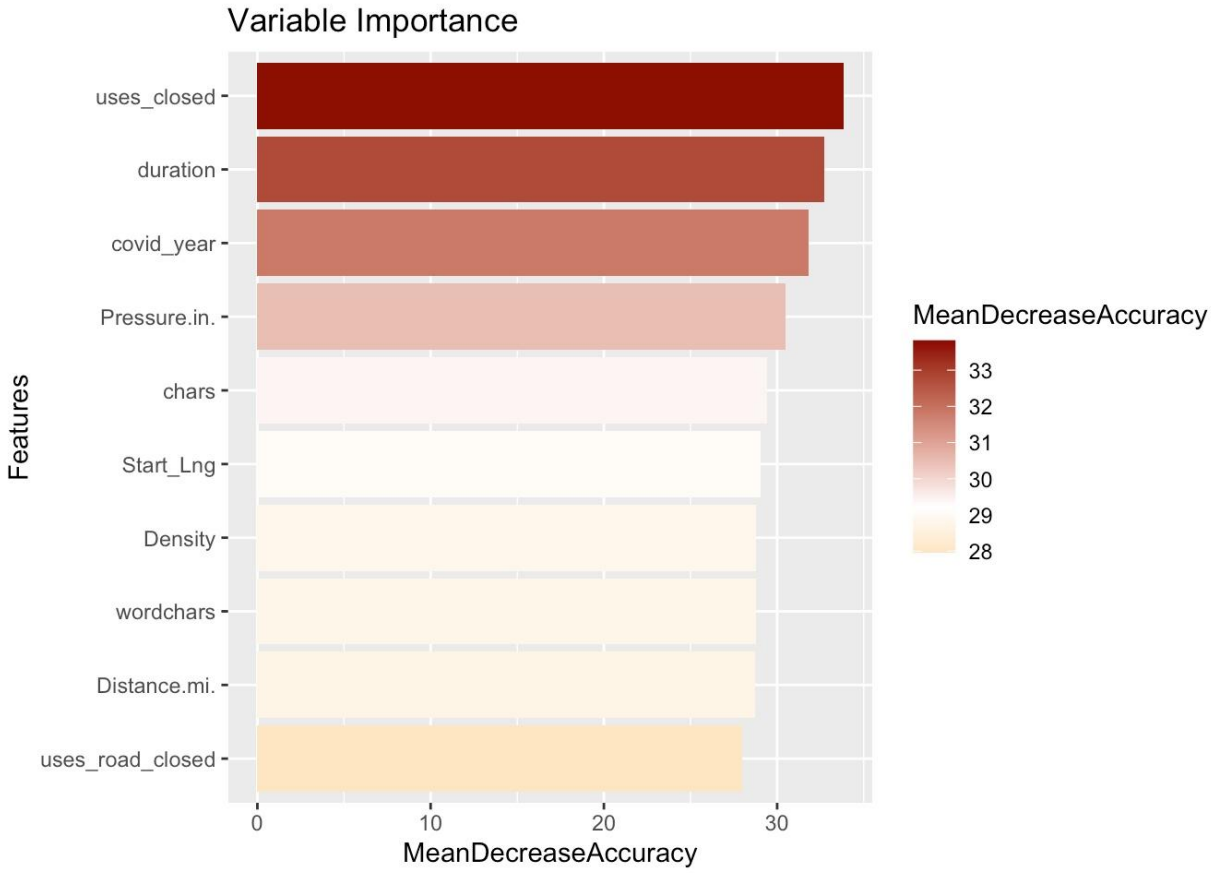


Figure 5.4: A gg-plot for MeanDecreaseAccuracy section from the summary of varImpPlot()

Since the randomForest model had a training accuracy of 100%, which was the highest training accuracy, we used this last subset of predictors to make the prediction. The combined public and private accuracy rate we got on Kaggle was 94.25% This is also the best prediction we made.

6. Discussion and Limitation

The dataset had a combination of both qualitative and quantitative predictors and therefore certain modeling methods were not suitable such as KNN and ANN. Additionally, the ratio between severe and mild accidents was highly skewed therefore clustering methods were not applicable.

In terms of the randomForest model we used, we noticed a strong case of overfitting as our training accuracy was 100%. If we were given more we would explore smaller subsets of predictors in order to decrease the variability in our model.

While testing data shows this model's validity on our selection of the U.S. traffic accident data this does not necessarily mean it will perform well when generalized to other data sets for car accidents.

7. Conclusion and Recommendation

This project provided many challenges, however the random forest model we developed seemed to accurately predict car accident severity for our testing data. Many other teams reached similar results using different methods. To improve our models we suggest mixing approaches and trying out all predictors created by all teams. There's also much more publicly available data for zip code and county that could be joined to this data set given more time. That being said many of the top groups doing this project had similar testing scores so this combination may still not lead to better results.

Throughout this project, it became clear that description was a vitally important predictor. This shows how even with all of the other information using accounts from the

situation is still extremely useful in understanding car accidents. It also demonstrates how car accident severity is not encapsulated well by one predictor. This project also shows the importance of simplifying complex predictors since simplifying time, description, and whether allowed us to create a much better model. Overall this project provided an excellent opportunity to practice our skills and demonstrate the power of classification algorithms to a vitally important issue.