SAFE DRIVE: GETTING PEOPLE HOME SAFE WITH MACHINE LEARNING

A PREPRINT

Harish Chandrasekaran
University of Virginia
Charlottesville, Virginia
hc9ne@virginia.edu

Rohit Chhatre
University of Virginia
Charlottesville, Virginia
rc7zb@virginia.edu

Jared Hood
University of Virginia
Charlottesville, Virginia
jwh6ry@virginia.edu

December 9, 2019

ABSTRACT

In this project our team aims to predict the likelihood of an accident occurring based on the different factors a driver faces. The data comes from car crash data collected across the state of Virginia by the Department of Motorized Vehicles. Some of the important features identified in this project are the speed limit, day of the week, weather condition, visibility level, intersection type, and population density. Since all of the data in the dataset represents a crash that occurred, the models used needed to be able to perform outlier detection. The models used were one class support vector machines, k-nearest neighbors, and isolation forest. With the one class SVM the different kernels tested were rbf, polynomial, and linear. Overall the isolation forest model achieved the best accuracy on the testing set with 89.7% and an F1-score of 0.946.

1 Introduction

Due to the importance and reliance on cars as a primary mode of transportation, it is important that we properly attempt to mitigate the major risks we face as we operate our powerful, yet mundane vehicles. In Virginia, car crashes have contributed to 819 deaths in the last year and a staggering 182 injuries per day. Despite being an important skill, safe driving has been highly overlooked, as we can observe from these statistics. Before even taking a look at the data, we can hypothesize that there are several factors that may indicate whether a crash is likely. For example, driving conditions such as weather and visibility can have a large impact on how drivers act on the road. As such, this project will look to use crash data to predict incidents before they happen and perhaps identify problem areas where accidents are more likely.

1.1 Dataset

The dataset for this project was sourced from the Virginia Department of Motorized Vehicles. It contains over 800,000 samples of various crashes with over 200 features, some of which are numerical and others are categorical. Some of the features include: collision type, date, area (rural or urban), latitude, longitude, time, weather conditions, and day of the week. Although the dataset contains a large number of features, we chose not to make use of all 200 features as some would not help us gain additional insights and train our model. However, after some initial data analysis we identified some features that may be of importance to us, such as the location, weather conditions, time of day, and perhaps even indicators about the driver such as gender and age.

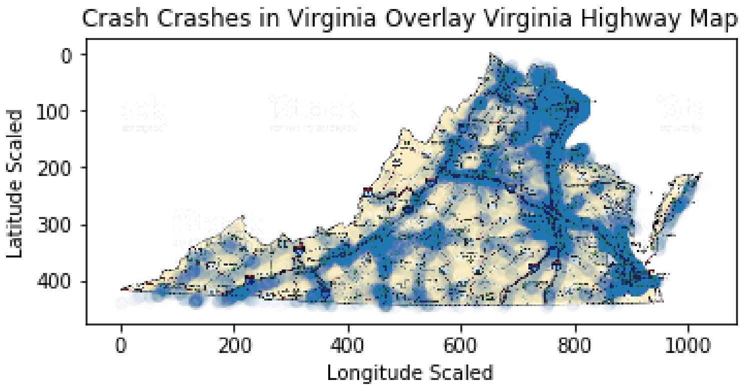

Figure 1: Virginia Crashes.

1.2 Related Work

In a project titled "Using Machine Learning to Predict Car Accident Risk", car accident data from the state of Utah is used to train a model to predict the likelihood of an accident. The crash data included two distinct types of features: static and dynamic. The static features were parts of the data that do not change with time such as road geometry, speed limit, and population density. The dynamic features include things that change with time such as weather, time of day, and date. In this project car accident risk prediction was posed as a classification problem with two labels, accident and no accident. The model used in the project was gradient boosting, which builds upon the machine learning concept of a decision tree. After training until convergence with 10 rounds of early stopping the model achieved a final ROC AUC of 0.0828. The model achieved a recall value of 0.89 and a precision value of 0.31 meaning that the model was able to predict 89% of car accidents correctly 30% of the time. The most important features in the model were time of day, temperature, and population density. In conclusion the model was not perfect, but served as an indicator into the ability to predict accidents.

Overall, our research in this area aims to answer the following questions:

1. Can we predict crashes based on the features and samples from the dataset?
2. How can we use these insights to more efficiently deploy resources and save more lives?

2 Method

In order to gain insights from the data, one step in this project was data visualization. From the original data set the longitude and latitude of each car crash was extracted, along with the speed at which the car occurred. Each crash was then plotted with a low alpha value onto an overlaid Virginia road map by longitude and latitude. Another map was also created plotting again by longitude and latitude, but this time including a heat map based on the speed the car crash occurred. This map indicated that a large number of crashes occur at lower speeds and in high density areas. This leads to the insight that the population density and crash speed were important factors in predicting a car crash.

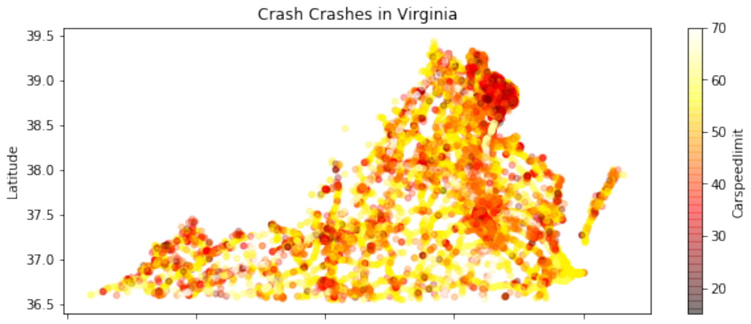

Figure 2: Virginia Crashes Heat Map.

The Virginia Crash Data set from the DMV contains over 200 features so the next step in this project was selecting the ones which were most likely to be able to predict a crash. The features selected included both numerical and categorical data. The categories included car speed limit, alcohol or no alcohol, light level, day of the week, weather conditions, population density, night or day, age, intersection type, and time of day. After selecting the categories the new feature set was checked for Null values. There were only 5 samples that had Null values so they were simply dropped from the dataset over using an imputer to fill in the values. Before using the new dataset with the selected features, the data was put through a pipeline in order to standardize it. The numerical features were standard scaled and the categorical features were one hot encoded. The features were then recombined into one data structure with the use of a column transformer. Finally, the dataset was split into a training and a testing set, which were then both split into smaller subsets 10% of the size in order to be used for tuning hyperparameters.

All of the samples from the dataset are instances when a car crash occurred, which means that all of the samples are members of the same class. This was important in choosing the models to test the dataset on as the model needed to be able to perform outlier/ novelty detection. This works by creating a boundary around the training set and identifying anything outside of this boundary as not a car crash. The first model used was the One Class SVM from the sklearn library. The One Class SVM was tested with the rbf, polynomial, and linear kernels. As each kernel took multiple hyperparameters as input, a gridsearch was performed in order to tune the hyperparameters to find the best pair for each model. The gridsearch was performed on a logarithmic scale for the gamma parameter and a linear scale for nu. Additionally, for the polynomial kernel the coef0 parameter was added with possible values of 0 or 1. The gridsearch was done on the small training and testing sets as doing it on the larger set would've taken a long time. After the optimal hyperparameters were identified the different kernels were then fit on the entire training dataset. After training each model was then evaluated for its accuracy. Finally the confusion matrix for each model was found and from which the F1-score was calculated.

The No Free Lunch Theorem dictates that multiple models should be tested and the best picked from them. Another model used in this project was the local outlier factor (LOF) from the sklearn k-nearest neighbors. Unlike how the one class SVM creates a decision boundary, the LOF computes a score reflecting the degree of abnormality of the observations. This measures the local density deviation of a given data point with respect to its neighbors and then identifies outliers as samples with substantially lower density than its neighbor. As with the one class SVM, with this model the hyperparameters were tuned on the smaller dataset and then the model was run on the larger dataset. Also, the novelty parameter was set to true, which allows the model to run on a dataset composed of only one class. After training the accuracy and f1 score were computed for the model.

The final model tested in this project was the isolation forest sklearn.ensemble. This model created a decision tree for the different features and used this to classify the crashes. As with the previous models the hyperparameters were tuned for this model as well. Then the accuracy and f1 scores were calculated.

3 Experiments

With the One Class SVM, three different kernels were tested including rbf, polynomial, and linear. The rbf kernel was run with the hyperparameters: gamma = 0.0056, nu = 0.3. The model achieved a testing accuracy of 69.74% and an F1-score of 0.822. The polynomial kernel was run with the hyperparameters: gamma = 0.0056, nu = 0.3, and coef0 = 1. The model achieved a testing accuracy of 69.56% and an F1-score of 0.820. The linear kernel was run with the hyperparameters: nu = 0.3. The model achieved a testing accuracy of 69.74% and an F1-score of 0.822. The neighbors local outlier factor model was run with the parameters: novelty = True, n_neighbors = 20, contamination = auto. The model achieved a testing accuracy of 78.12% and an F1-score of 0.877. The isolation forest model was run with parameters: n_estimators = 10, warm_start = True, n_jo = 100, behaviour = new. This model achieved an accuracy of 89.69% and an F1-score of 0.946. All of the results are summarized in the table below.

Experiment Results		
Model	Accuracy	F1-score
RBF Kernel	69.74%	0.822
Polynomial Kernel	69.56%	0.820
Linear Kernel	69.74%	0.822
Local Outlier Factor	78.12%	0.877
Isolation Forest	89.69%	0.946

4 Results

The main goal in this project was to create a model that can predict whether a crash will occur based on different features, like weather and light conditions. The Isolation Forest model performed the best out of all the different models that were tested. It resulted in a 89.67% testing accuracy. Based on the confusion matrix, the model classified 4496 cases as crashes and 517 cases as non crashes with an F1-score of 0.9456. The best indicators of a crash out of all the features were:

1. Alcohol use
2. Speed
3. Weather
4. Light Condition
5. Population Density

Although these are the features that one would expect to cause an accident, we had an interesting find in our results. Contrary to what people might expect, a lot of the crashes were in low speed limit zones compared to areas of high speed limits. This is because most in areas with high population densities like cities, there are more cars and thus more accidents.

5 Conclusion

At the conclusion of this project we were able to effectively predict car crashes in Virginia with around 90% accuracy. This result can help protect the well being of Virginia and it's residents by being able to identify whether a car crash may occur given certain circumstances. Further, the model identified several features which people can be aware of that are indicative of a higher crash percentage. When there are scenarios in which some of these features are present, law enforcement can step in to help provide preventative measures, so accidents are less likely to occur. For example, officials can put out warnings when there are hazardous road conditions. Also, police can be stationed in areas where the crash probability is higher in order to be closer to potential crashes and prevent more injuries and deaths on the road.

While this project achieved a fairly high success rate, there are many steps we can take to improve our model. Due to the limitations of our dataset we used a One Class SVM, however we would have liked to extend our research by including negative sampling. By doing so, we would aim to classify low-risk areas in addition to the high-risk areas we are now identifying. In addition, we would have liked to look into the possibility of creating an application wherein the user can input the current conditions and receive a safety rating. This would prepare drivers mentally when hazardous conditions are present, which can in turn also reduce the number of distracted driving incidents.

6 Contribution

Harish Chandrasekaran: I was involved in the initial data visualization and preprocessing steps as we decided how we would approach and tackle this project. Later, I researched and implemented K-Nearest Neighbors and Isolation forest to detect outliers in the dataset. In a One Class SVM, this is important as our outliers are classified differently than the rest of our data. In the report, I worked on the introduction, results, and conclusion sections.

Rohit Chhatre: I contributed to the initial idea to do a project with the crash data in Virginia and to the project proposal. With the code for the project, I helped with data cleaning and figuring out which features were relevant to include. I created the initial pipeline for all the data and removed any excess features that were not needed. I also helped tune the hyperparameters for k-nearest neighbors and isolation forest. In the report, I wrote the Results and Conclusion section.

Jared Hood: I completed the data visualization of the dataset. This included mapping the longitude and latitude of crashes in Virginia and overlaying them over a map of Virginia highways and cities. I also created a heat map of the crashes based on the speed at which the crash occured. Another part of the project I worked on was creating the One Class SVM models and tuning the hyperparameters with gridsearch to find the best combination for each kernel. Then I ran the model on the entire training set with the optimal parameters and found the accuracy and calculated the F1 score. I also wrote the method and experiment sections of the report.

References

[1] Daniel Wilson. Using Machine Learning to Predict Car Accident Risk : *https://medium.com/geoai/using-machine-learning-to-predict-car-accident-risk-4d92c91a7d57*

[2] Sklearn. Novelty and Outlier Detection : *https://scikit-learn.org/stable/modules/outlier*$_{d}etection.htmloutlier - detection$

[3] Stack Overflow. How to Use GridSearchCV on OneClassSVM : *https://stackoverflow.com/questions/44698928/is-there-a-way-to-perform-grid-search-hyper-parameter-optimization-on-one-class*

[4] Deepankar Arora Anomaly and Outlier Detection using Local Outlier Factors : *https://www.datasciencecentral.com/profiles/blogs/anamoly-outlier-detection-using-local-outlier-factors*

[5] Department of Motorized Vehicles Full Crash : *https://www.virginiaroads.org/datasets/full-crash*