NOVEL PREDICTION ALGORITHM SUPPORTING THE EXPANSION OF AUTOMATED VEHICLE TESTING IN THE COMMONWEALTH OF VIRGINIA

A PREPRINT

Liem P. Budzien
Department of Computer Science
University of Virginia
Class of 2021
lb7me@virginia.edu

Kevin V. Dang
Department of Computer Science
University of Virginia
Class of 2022
kd9me@virginia.edu

Jennifer P. Huynh
Department of Computer Science
University of Virginia
Class of 2022
jph5au@virginia.edu

October 30, 2020

ABSTRACT

With the growth of the usage of automated self-driving vehicles throughout the country, the practice of running tests on existing roads has come under the attention of many lawmakers and developers alike. Given the recent introduction of the technology, there is concern regarding the safety of such tests being conducted on public roads. This project is focused on solving the issue of delegating roads for the usage of testing self-driving vehicles given that there are only 70 miles of roads in Virginia currently being used right now. By utilizing specific machine learning technique, our experiments intended to predict the severity of crashes given a large data set provided by Virginia, enhancing the ability of the state to recommend areas of expansion for automated vehicle testing. We found that the models that produced the clearest results for road suggestions were clustering by k-means algorithm and using artificial neural networks.

1 Introduction

The experiment is based on predicting the probability of causing a car crash based on different correlating attributes that are given within the dataset, such as weather conditions, time of day, and type of vehicle. The problem and solution are theoretical in the sense that the probability of crashing and severity of the incident is being predicted by the model that we will be applying within the project. We hope to apply the predictions of the model to the larger umbrella of automated driving. Should the model perform well on the Virginia dataset, we would be interested in expanding the geographic areas covered by the algorithm to generalize it to the United States. Virginia is currently a free-for-all for the usage of self-driving cars in the sense that they are not disallowed from being used nor are they being allowed. This is mainly due to the ongoing tests related to the capabilities of self-driving cars in different conditions with a comparison to human-driven cars. Our solution to the prediction of severity, cause, and the probability of an accident would then apply to this current grey-area in relation to self-driving cars.

A more specific solution to self-driving cars that we are hoping to answer would be related to the areas of which self-driving cars have been specifically allowed to be used. There are currently only 70 miles of Virginia roads that are open to self-driving cars which means that only 70 miles have been deemed safe. Our solution would then be able to answer at any given moment the risk of using self-driving cars based on driving conditions that are fed into the model.

Thus, we propose our research question: *"Can past traffic accident data be used to predict the probability or severity of future accidents accurately enough to inform a recommendation about which roads to allow self-driving testing to occur in Virginia?"*

2 Method

2.1 Preprocessing the Data

In terms of data processing, we decided to implement the typical machine learning methods associated with this step. The first step in processing was to clean the data. Several steps were taken to clean the data, starting by dropping data categories/columns that were irrelevant or unhelpful for machine learning. Categories that were not relevant to the modeling, such as 'Case Document Number', and repeated data such as 'X' and 'Longitude' were removed. In addition, many columns containing categorical data lacked values. For example, multiple True/False columns only contained 'True' values, while the cells representing 'False' were left blank. Thus, to make these columns better suited for machine learning, we filled the blank cells with the applicable missing value.

An important step was to drop data columns that were directly correlated to the categorical value our project was predicting. We restricted the training data to be factors available prior to a crash, as all data recorded following a crash were directly correlated. For example, 'Crash Severity' and 'Number of Passenger Deaths' were directly correlated, as one of the crash severity categories was 'death'. Thus, by restricting the training data to only data available before a crash, the model will be more applicable in the real world.

Next, we chose to use a pipeline to pre-process the data given that there were many data columns initially containing categorical data within the CSV file. The decision to use a pipeline was due to the ability of a pipeline to standardize the order and methods applied while pre-processing the data. The pipeline used an OrdinalEncoder to convert the categorical data into numbers able to be trained using machine learning algorithms. OrdinalEncoder was chosen over alternatives due to the nature of the categorical data contained within the CSV. The data was ordinal in nature, such as levels of severity or simple True/False data. Thus, the OrdinalEncoder was the optimal solution to process the data, with the added benefit of keeping the dimensions of the data frame identical. Should OneHotEncoder have been used, with the size of the data set, additional columns for high cardinality columns would have made the machine learning models difficult to train. For numerical data, StandardScaler was utilized to scale the numerical data to equalize the weights of each column on the regression, to prevent certain data categories from influencing the models more than others. Finally, we used a SimpleImputer on the numerical data to fill in missing values, to increase the number of rows able to be trained in the CSV.

2.2 Splitting the Data

Following the pipeline, $train\_test\_split()$ was used in order to separate the data into training and testing sets. After doing so we then further split the data into X and y which then allowed us to move onto implementing different models on the dataset. For specific models, we performed additional preprocessing, such as creating a validation set for an ANN model and creating a smaller training set for the SVM.

2.3 Model and Strategy Selection

In order to find an optimal solution, we employed a variety of different machine learning strategies to analyze the data. To help generate better outcomes, we relied on feature engineering. For example, we took the categorical data column 'Driver_Action_Type_Cd', which is a categorical data column, and created a new category. We converted each string code, which contained a number between 1-20, and split the string and extracted the numerical value. By converting the strings into ordinal numerical values, the machine learning algorithms were better able to parse and train on the data.

For Data Visualization and recommending specific self-driving car corridors, we used KMeans Clustering to generate centroids and clusters of accidents to form regions of roads within Virginia. We also used regression, which we evaluated performance using RMSE scores. Finally, the last strategy utilized were classification models, evaluated using accuracy scores.

3 Experiments

3.1 Regression

For our baseline, we used a Linear Regression on the crash dataset in order to predict the severity of the crash. We used the category of crash severity (with values ranging from 0-4) as our Y, and the other categories as our X values. In addition, find the best regression model to use, we also used Decision Tree and Random Forest regressors. Due to the size of the training data set, the minimum number of samples per leaf was set to 20 to prevent overfitting and to decrease the training time.

3.2 Clustering

Wanting to use a variety of different machine learning techniques as possible for our dataset, an analysis method was an attempt to classify by a clustering method, using a k-means algorithm. We clustered the dataset instances using the training data, taking into account features such as location, crash severity, and weather. We did so using SKLearn's Minibatch KMeans Clustering. We incremented our clusters by a count of 1 in each run, and had a total of four runs. Following the clustering, we plotted each cluster as a scatter plot, with each cluster depicted as a different color to aid the visual analysis.

3.3 ANN

Another machine learning technique that we used was a artificial neural network (ANN) for classification. We utilized the keras neural network library. We decided to implement an ANN in order to see how the network would work towards predicting Crash Severity values given an input layer being the dataset that we cleaned. Neural networks are useful for predicting a value such as Crash Severity because it allows the algorithm to find patterns within the dataset without user input being needed which is especially helpful for dataset such as car crashes where many factors can be correlated with the crash severity. The output layer ended up having a dimension of five which corresponded with the fiver different potential classifications of crash severity as initially seen in the dataset. After comparing various hyperparameters, we found that the optimal loss function to be 'cross_cateogorical_crossentropy' and the best optimizer to be a stochastic graident descent. In addition, the ANN ran for 10 epochs, after which point we found the validation set accuracy score did not improve.

3.4 SVM model

We originally started our model training process by looking to implement SVMs, as a linear SVM would create a simple starting point for the direction of our dataset and was thought to be a personally easier to implement in regards to other algorithms. A linear kernel was chosen due to the size of the training data (about 800,000 rows), as a poly kernel would have taken significantly longer to train. In order to implement the SVMs on our data, we needed to find the best possible parameters associated with each of the different kernels. To do so, we needed to drop as many columns of data that were found to be irrelevant to what we were trying to predict. To measure the results of our SVM implementations, we looked at each output of the confusion matrix, recall, and precision score. As the linear kernel SVM struggled to train on the full data set due to the size, the SVM was trained using 10% of the original training set. To further help performance, a OnevsRestClassifier was used in addition to a BaggingClassifier.

4 Results

4.1 Measuring Accuracy and RMSE

The results for the Regression experiments were the RMSE scores that were computed when comparing the test value and predicted value. Using the data to train the Linear Regression algorithm to predict the crash severity, we were able to train the algorithm to attain a 1.329 RMSE with the scale being $1.0 = 1$ category. The next regressor we used was the Random Forest Regression package from sklearn. Using the built in functions, we were able to fit and evaluate our data producing an RMSE of 1.299. Finally, for our last regression algorithm, we utilized Decision Trees. Following the same steps as for the Random Forest Regressor, we were able to produce an RMSE of 1.398.

Using our basic knowledge of SVM implementation, we ran into several problems where we were continuously computing an accuracy score of 1.0. We ultimately addressed this problem by dropping more data columns that were directly corresponding to the data being predicted (essentially duplicate columns) and attained a more reasonable accuracy score of 0.670.

4.2 Final Models

Using the clustering algorithm MiniBatchKMeans from sklearn, we fit and grouped the clusters in accordance to relative location. We can see the clear division of groups best when we first choose to set $clusters = 3$, see 4 or $clusters = 4$, see Figure 3. When the desired clusters are set to 2, there is no clear separation of groups, as the results tend to overlap over each other, see Figure 2. When desired clusters are set to 5, our maximum number of clusters in our runs, we can clearly see two groups rather than the five desired clusters, see Figure 5. There seems to be more overlaps between all the groups, making it to difficult to visually observe and analyze any distinct groups. According the the Elbow Curve graph shown in Figure 6, we can visually determine that the most optimal number of clusters to use would be 3-4 clusters.

We decided to use a classifier rather than a regressor to evaluate our dataset due to the categorical properties of the values that we were trying to predict being crash severity. Even though crash severity did have numerical values attributed to each of the possible labels, the values were still associated with categorical data; we felt that the classifier would be better equipped to handle the data set and predict the crash severity. In addition, as we were predicting a category, we felt that accuracy was a better metric of performance than RMSE, which represented the numerical error between the predicted category value and actual category value. As a result, the other model that we selected in our final solution was the artificial neural network, ANN. We utilized the ANN to predict the potential classifications of crash severity given our dataset. After running the data through all of the layers, we were able to achieve an accuracy of 0.670.

5 Conclusion

Analyzing our results, we believe that we are able to answer our original research question with reasonable level of certainty. The original scope of the project was to identify certain roads or sections of Virginia that we found fit to use self-driving cars within. Given just our clustering algorithm alone, we were able to plot clusters of Virginia based on the attributes of the data set. For example, in Figure 2, we are able to see that the red cluster typically outlines the larger roads of Virginia like highways and beltways, but the underlying purple cluster represents smaller, more rural, roadways. As the number of cluster increases, the regions pertaining to a cluster become smaller and more specific meaning that it makes it easier to narrow down specific roadways and sections of Virginia to recommend; in Figure 4, we can see a cluster along the western edge of the state corresponding to the Appalachian Mountains and more specifically, Interstate 81.

In addition we can consider the classification model along with the clustering results. Our classification model, the ANN, was able to predict the crash severity with an approximate of 70% accuracy. This means that given certain parameters of a cluster similar to those existing in the dataset, the ANN can predict the crash severity. Tying this all together, along with being able to identify clusters to recommend for self-driving testing we can then also use the information available in each of the clusters to predict the scale of crash severity. In doing so, our project is then able to successfully identify specific roadways of Virginia safe for self-driving cars thereby eliminating the current issue of only 70 miles of road currently being used for testing.

However, it should be noted that our research project in its current form contains a number of shortcomings. The biggest shortcoming is the difficulty of actually being able to recommend any specific instances, or individual roads for self-driving testing. Our accuracy score of about 70% is adequate enough for us to make general policy decisions concerning self-automated driving, but is not high enough of a score to guarantee the outcome of a test on any specific road. This implication would then be potentially dangerous if we were to consider the idea of people using the self-driving cars, albeit this would be avoided during testing but is still a concern. Thus, while our model is accurate enough to provide recommendations, we do not advise the use of the model for definitive or final decisions. We would recommend further investigation.

As our algorithm relies only on the data that is able to be collected prior to an accident occurring, such as weather conditions, speed limit, and time-of-day, the algorithm is applicable to almost any location. However, in other geographic locations certain factors may have a greater weight, thus, we will need to gather data from the new locations as well and feed them into the model. As self-driving vehicles enter public roads, it is vitally important to protect the safety of the public. Thus, informed policy decisions are crucial. For Virginia, we recommend the I-81 interstate and western edge of Virginia for the next expansion of the self-driving vehicle testing network of roads. This is an important step, as the current Automated Driving Corridors in Virginia are located around Blacksburg and Northern Virginia. This expansion will connect the two, creating a significantly larger contiguous network available for testing. To further expand on this project, we hope to spend more time optimizing the ANN model. While the results computed were sufficient for the purpose of our project, we believe that if given more time, we could further improve the model and accuracy score. Additionally, in the future, this research and resulting algorithm can be expanded to other states and countries, to inform their respective policy decisions.

6 Contributions

- Jennifer Huynh: Responsible for the regression modeling and SVM.
- Liem Budzien: Responsible for cleaning the preprocessing the data, k-means clustering, and ANN modeling.
- Kevin Dang: Responsible for classification models and graphing results.

7 Figures

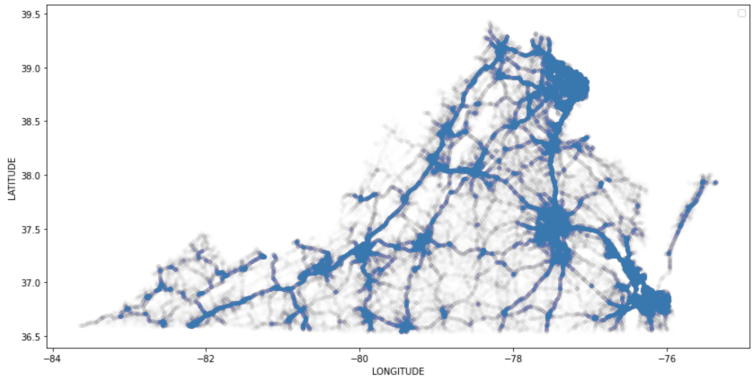

Figure 1: Virginia map with Cleaned Dataset Points

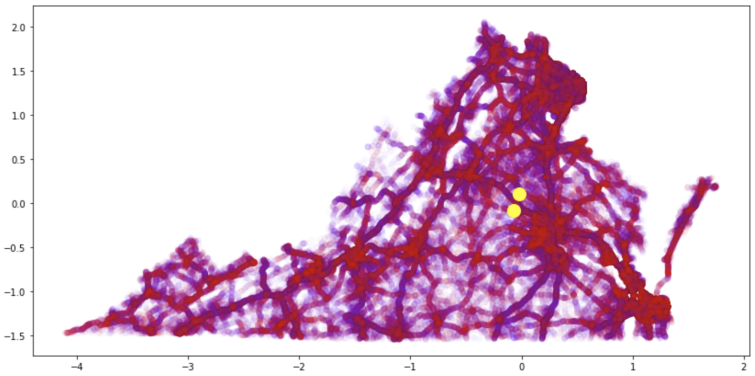

Figure 2: Virginia map with 2 clusters.

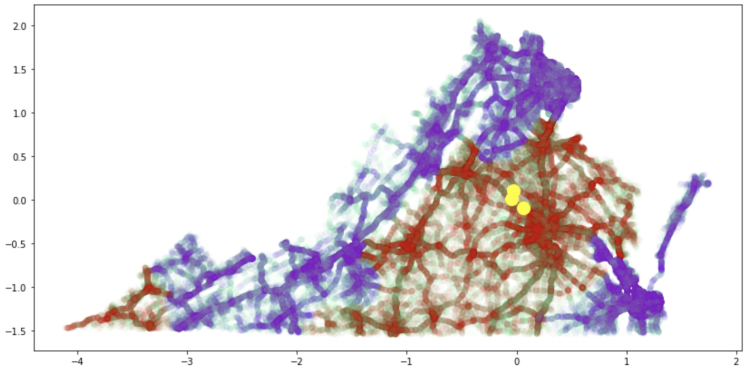

Figure 3: Virginia map with 3 clusters.

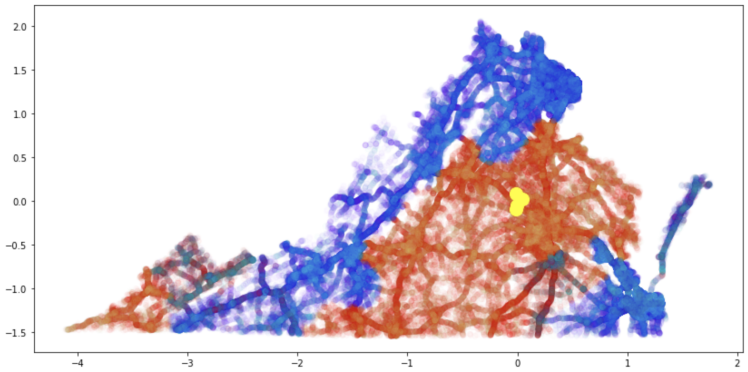

Figure 4: Virginia map with 4 clusters.

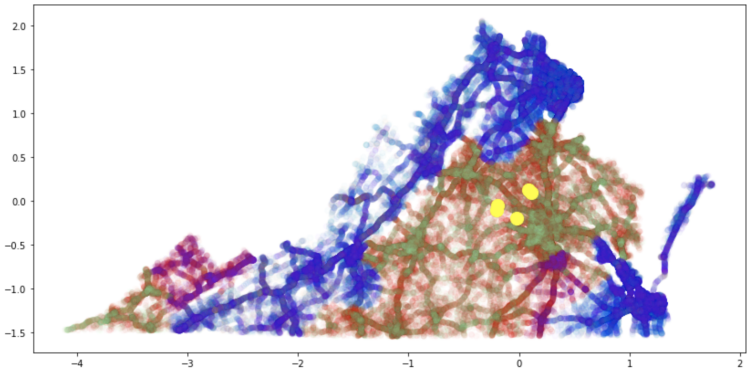

Figure 5: Virginia map with 5 clusters.

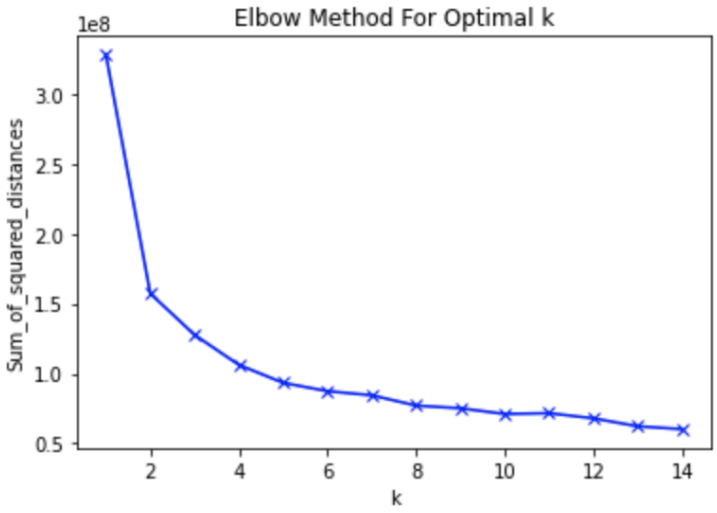

Figure 6: Elbow Curve of Clusters

References

[1] Abdishakur. "Predicting Vehicle Accidents with Machine Learning." Medium, Towards Data Science, 1 Jan. 2019, towardsdatascience.com/predicting-vehicle-accidents-with-machine-learning-ce956467fa74.

[2] Antonio, Meraldo. "Live Prediction of Traffic Accident Risks Using Machine Learning and Google Maps." Medium, Towards Data Science, 24 Oct. 2019, towardsdatascience.com/live-prediction-of-traffic-accident-risks-using-machine-learning-and-google-maps-d2eeffb9389e.

[3] Chong, Miao & Abraham, Ajith Paprzycki, Marcin. (2005). Traffic Accident Analysis Using Machine Learning Paradigms.. Informatica (Slovenia). 29. 89-98.

[4] Kharkovyna, Oleksii. "Top 10 Machine Learning Algorithms for Data Science." Medium, Towards Data Science, 18 Apr. 2019, towardsdatascience.com/top-10-machine-learning-algorithms-for-data-science-cdb0400a25f9.

[5] Wahab, Lukuman, and Haobin Jiang. "A comparative study on machine learning based algorithms for prediction of motorcycle crash severity." PloS one vol. 14,4 e0214966. 4 Apr. 2019, doi:10.1371/journal.pone.0214966