MACHINE LEARNING FOR VIRGINIA CAR ACCIDENTS

Ben Stansell
University of Virginia
bjs4sy

Arthur Harris
University of Virginia
rah9eu

Chris Kazakis
University of Virginia
ck7aj

April 30, 2020

ABSTRACT

For this project, we investigate when and where crashes result in injury. We have many other variables such as if the driver is buckled, the weather conditions, type of crash, if the driver was drunk, etc. We want to see how a driver can reduce their chances of being injured as a result of an accident, and how emergency responders can be prepared for accidents. We are attempting to predict if a crash will have serious injuries or a fatality. The ideal outcome for this project would be to gain insight into why crashes happen and how to avoid more deadly crashes.

1 Motivation

From deaths, to injury, to other less-severe accidents, there is a lot of potential pain that can be caused on roadways. We plan to use a machine learning algorithm to predict the frequency and severity of crashes across Virginia, so that emergency response can be better prepared to take care of calls, and to make drivers more aware of relative dangers. We also seek to decrease the frequency and impact of crashes in any other way possible. Our project is primarily an applied project, since our goal is to help predict the likelihood of crashes and the number of injuries in a crash, should a crash be predicted.

2 Introduction

Roadway accidents are an unfortunate side effect of modern life in Virginia. While there is some other transportation infrastructure, particularly around the Washington DC area, few people across the state can go without automobiles. Since the year 1946, over 30,000 people have died every year from a motor vehicle accident. Looking at these statistics is grim and discouraging. It begs the question of if it is really using this mode of transportation. Modern car manufacturers have greatly improved the safety of vehicles in the unlikely event of a collision, but these improvements are still far from guarantees. Some context for this improvement is that in 1946 for every 100 million vehicle miles traveled, there were 9.35 fatalities. In 2018, this number was lowered to 1.13. This is obviously much lower, but calling it "good" still downplays the lives that are lost every year. In 2018, 36,560 people or one out of every 8,947 people died in Virginia from a roadway accident. Our goal going into this project was to reduce this number in any way possible. One way that we thought we could accomplish this was by predicting what scenarios would be more likely to have a serious accident. This could both alert drivers to be more cautious. Also emergency responders could be alerted when particularly dangerous scenarios were present. Our goal was given information about a crash, how serious would it be.

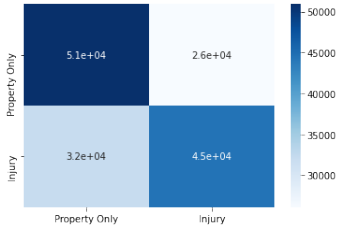
3 Method

The dataset that we used is from VDOT and had several oddities, to say the least. To start with, the dataset had 893663 entries and 66 features. About half of the entries had at least one feature with a NULL. We used one categorical variable, 'Crash Severity' as the y-value for the data. Crash severity was measured at 5 levels in order of increasing severity: property damage only, non-visible injury, visible injury, severe injury, and fatality. A crash's label is ultimately determined by the most severe injury resulting from the incident. For our X values, we dropped about 40 variables we assumed had nothing to do with our y-value (such as Document Number), or had too many categories (such as Route

Number)or would taint our results with information about the outcome we sought to predict (such as Pedestrian Injury Type). We then one-hot-encode the remaining categorical variables, and normalized the continuous variables. After this process, we were left with over 230 features, and did some initial, unsuccessful model building. As a result, we dropped features with less than 3000 occurrences, (about .3 percent), and dropped features with injury rates closest to the population rate of about .34, and had 40 features left to work with. Once the data was cleaned, we started with models that attempted to distinguish between property damage and injury, with the intent of creating a 5-category classifier later on. Initially we wanted to predict whether crashes would result in fatalities, but only about 5000 observations (less than 1 percent) were fatalities, so we focused on the other problems with more data in each class. We elected to mainly focus our efforts on creating neural networks to accomplish this goal, as they seemed like the best method to tackle such a large amount of data. We also used other modeling methods to supplement our efforts such as k-means, logistic regression, support vector classifiers, rand forest classifiers, and ensemble classifiers.

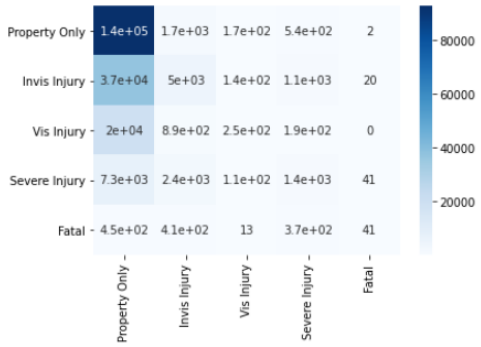
4 Experiments

We first tried a K-means algorithm. The traditional "elbow method" of finding a good number of clusters showed a distinct drop in error at k=2, so we ran several iterations with 2 clusters. The initial results were difficult to interpret with 40 features, and clusters with distinct injury rates rarely emerged. When they did emerge, we obtained a few pieces of information that were already apparent from our variable selection procedure (namely that accidents involving pedestrians or bikes are more likely to result in injury and hitting parked cars and animals result in injury less often. Some clusters emerged along highways, but mainly due to differing speeds and speed limits, and not due to increased injury rates. Initially, we attempted to predict whether a crash would be fatal with random forest and SVC models, but the relative infrequency of fatalities (about .6 percent of all accidents) made analysis difficult, as models could get over 99 percent accuracy by always guessing 'no fatality'. We then moved on to predicting whether accidents resulted in any injury at all with several using neural networks. We chose neural networks to solve this problem. We tried several different models with different numbers of nodes, layers, learning rates, and features. After being met with little success, We dropped many columns from the data set and only kept columns that were more highly correlated with injuries. This improved our results. We also randomly sampled data from the rest of the data to ensure that we would have an even 50 percent of data with injuries and 50 percent without injuries. We ran into this problem earlier when we were trying to predict fatalities and this seemed to be a good remedy to this issue. We also read that amsgrad, a slight variation on the adam optimizer, sometimes converges to better results more quickly than normal adam (Gugger and Howard). The confusion matrix for our best Neural Network can be seen below:



The resulting best Neural Net gave us 62.5 percent accuracy. 62.5 percent accuracy was the best we had legitimately gotten at this point, but still not enough. Based on the confusion matrices, it appeared that all the models were still heavily favoring guessing 'no-injury' over 'injury'. The train test and validation accuracies were all very similar, so we concluded that our models were underfitting, and attempted to make adjustments to drop-out layers and batch sizes, and added more nodes, layers, and epochs. We tweaked and optimized many Neural Nets attempting to improve the accuracy and better learning the subtle differences in the data. This had little effect on accuracy, so we investigated other techniques, primarily particle swarm optimization and ensemble learning. Particle swarm optimization yielded about the same results as the prior neural networks, so we moved on to ensemble learning. Here we found that ensemble learning can sometimes outperform neural nets, and were especially good if some certain cases were very difficult to identify, as the average of three models might make those cases easier to predict (Brownlee). We chose logistic regression, svc, and random forest classifiers for the individual components of the model using default parameters, as they were the main options we had left. The ensemble learning model outperformed the individual components, but had accuracy at about 61 percent, which was lower than our best neural net.

We elected not to continue improving this model, as the accuracies were very similar among all models, and their confusion matrices suggested that they were making roughly the same incorrect predictions as the neural net, and the similar performance of the voting model to its components suggested the 3 models were all similar to one another. Once our work on the binary classification reached its end, we attempted the 5-class problem described above, using the best neural network from the previous part of the project. A confusion matrix for the model can be seen below:

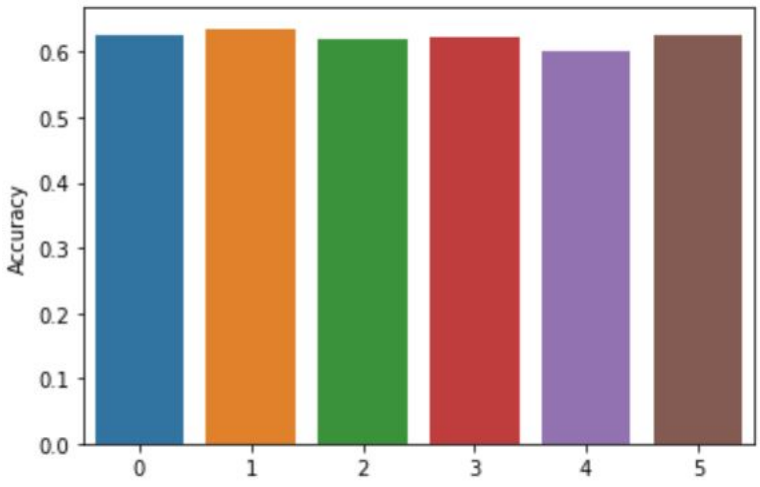


Given our previous experiences and attempts at improving our models, we concluded the underfitting was a result of a lack of relevant data in our source material, and not from inappropriate modeling. The neural nets and ensemble learning model, as well as their results can be found in this notebook: https://colab.research.google.com/drive/17lnNHocqTtx2ZOEKMaS0X91vjHOO41RW.

5 Results

The overview of these models is in experiments. The exact details of all models can be seen in the colab notebooks. We had many models which were able to predict the severity of an accident in the low 60 percent. There were many more that had even less glorious results.

	Name	Accuracy	Precision	Recall	Loss	Hidden Layers	Optimizer	Batch Size	Epochs	Activation	Learning Rate	Amsgrad
0	Batch Normalization	0.6252	0.6252	0.6252	0.6206	2.0	Adam	100.0	30.0	Softmax	0.0007	TRUE
1	More Batch Layers, No Amsgrad	0.6362	0.6362	0.6362	0.6106	3.0	Adam	120.0	30.0	Softmax	0.0070	FALSE
2	Even more layers, SGD, Reduced Batch Size	0.6199	0.6199	0.6199	0.6246	4.0	SGD	50.0	30.0	Softmax	0.0050	-
3	Neural Network with Nadam	0.6219	0.6219	0.6219	0.6230	2.0	NAdam	100.0	30.0	Softmax	0.0009	-
4	Neural Net with Adam, High Learning Rate	0.6006	0.6006	0.6006	0.6340	2.0	Adam	100.0	30.0	Softmax	0.0100	FALSE
5	Neural Net with Adam, Low Learning Rate, More ...	0.6246	0.6246	0.6246	0.6211	3.0	Adam	100.0	30.0	Softmax	0.0040	FALSE



When looking at a breakdown of each feature's importance to one of our most successful models, we see that the features with the highest correlation to severe crashes are (from highest to lowest) motorcycles, pedestrians, lack of seat belts, bicycles, head on collisions, and overturned vehicles.

6 Conclusion

As a whole, we were not able to predict outcomes as accurately as we had originally hoped. However, we did find some meaningful relationships between certain variables and chance of injury. Accidents involving motorcycles, bikes, and pedestrians almost always result in injury, so we recommend that these accidents take priority for first responders, and also recommend that VDOT investigate ways to improve safety for these groups. This could be in the form of changing traffic patterns or roadway markings. Also, certain collision types, such as head-on collision, and crashes that result in overturned cars were much more likely to produce injuries, so it would be best if first responders prioritized those as well. Injuries were also more likely when seatbelts were not worn, so we recommend that VDOT increase awareness of the importance of wearing seatbelts (such as click-it or ticket). On the other side of things, we discovered that crashes involving animals and parked vehicles rarely led to injury, and therefore responding to other accidents should take precedence over these.

The validity of these findings is limited due to our underperforming models, which was ultimately a result of the data lacking the information needed to answer our question. One of the main issues with the data was that there were many missing values, which had to be filled in. Additionally, some of the columns did not have 'No' values, but rather had empty fields that implied 'No' or 0, so it was impossible to distinguish between 'No' vs. 'No data' sometimes. Another factor that contributed to our lack of precision in modeling was all the factors at play in a roadway accident. If different people were put in the same crash, the seriousness of the injury could vary. Also different cars have different collision safety ratings, which depend on which angles the collision is in. There is an incredible amount of data that would be needed to be able to more accurately model crash severities. So much also depends on the split second decision making of the drivers involved in the accidents. It is certain that for every crash in this data set, there was a near crash in a nearly identical scenario that drivers were able to avoid. With the evolution of self-driving cars, much more data could be made available and the responses could be optimized. When data from self-driving cars becomes available, a study similar to this could have different and more revolutionary results. Unfortunately, that was not the data that was available.

Potential future work might also include investigation into what circumstances causes pedestrian / cyclist / motorcyclist accidents, and how those might be prevented, as they were by far the strongest indicators for injury caused by accidents in this data. We also found few differences in injury rates based on geography, so perhaps a state similar to Virginia has data, with different variables, that could be analyzed and then the conclusions could be extrapolated to Virginia. We did not find any revolutionary patterns, but we do think that there will be significantly fewer roadway injuries and fatalities in Virginia in 2020. It is possible that our research will have had something to do with that.

7 Code Repositories

Note that all Jupyter Notebooks can be found in the zip file that corresponds with this document.

https://colab.research.google.com/drive/1T5tHvEBhvRtbCDy111WULFVG3go163u4

https://colab.research.google.com/drive/1a5YAQaGjdta3-1584hxs6E3uuhWyFyjC

https://colab.research.google.com/drive/1iVlVWm7l1jxhAQRD5Uwz9utFV8FZhHVH

https://colab.research.google.com/drive/1Q6HBhvt59d9yTTCBLVPzrvbbY9fADNrF

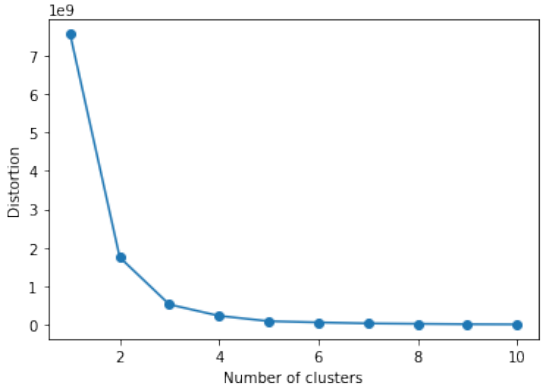
https://colab.research.google.com/drive/18vRzLSel3x4AvVd2vTOawGiplYIVlciC

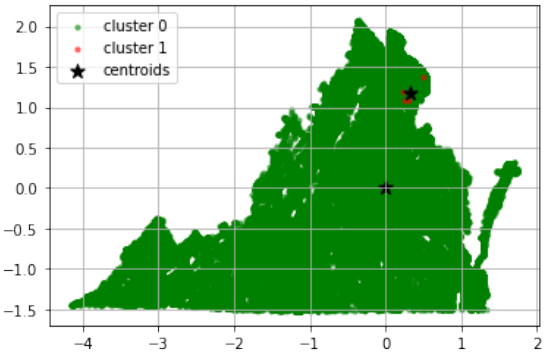
https://colab.research.google.com/drive/1HwP7zGZHO6SHVBfO5ibPPSqPQnT58J9d

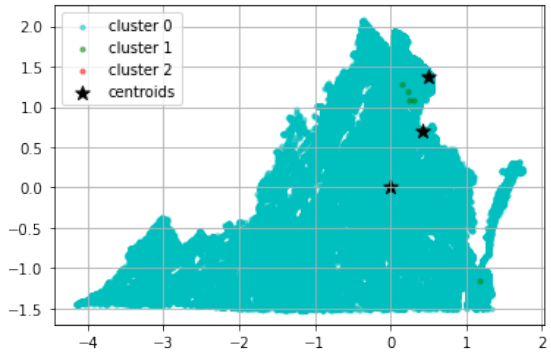
8 Graphs

These graphs were not particularly relevant to our goals in the end, but we found them interesting nonetheless. With 2 clusters, we see mostly just a split between East and West, which is not terribly interesting for our purposes. But when we do three clusters, while there still seems to be an East-West split, the third cluster forms along the major interstates and highways across Virginia. This suggests that there is a significant difference in these crashes from others, which may prove useful later (or not).

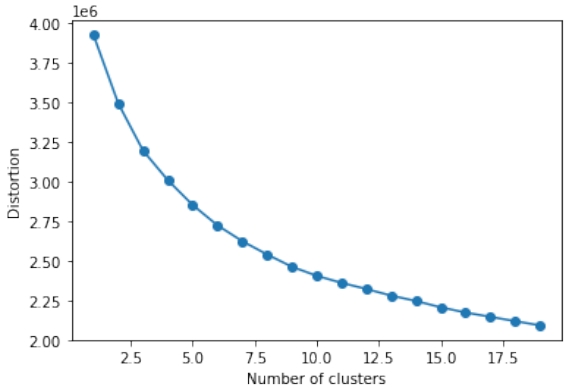
kmeans with mean replacement:

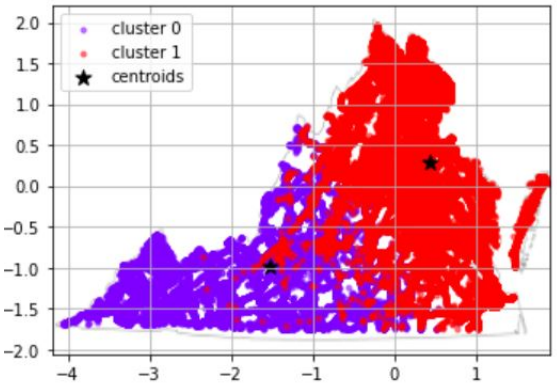


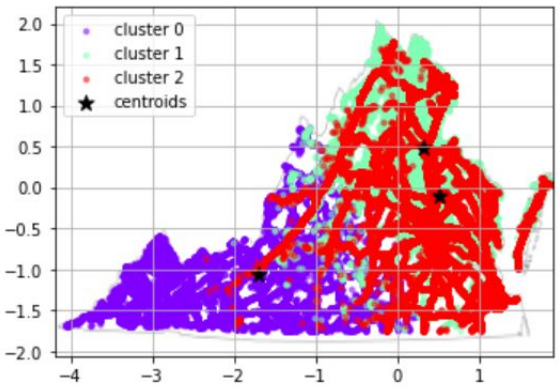




Kmeans with drops:







9 Contributions

For the preliminary stages, each group member had a role in deciding which variables from the dataset would be kept, which would be converted to dummy variables, and which would be removed. Arthur spearheaded the data cleaning, but all of us took part in it by writing several python lambda expressions and using the pandas dummy functionality to get data we could work with. Ben trained the various models for classification and documented the results, while Arthur trained the k-means algorithm and generate the images seen in the report using matplotlib. All of us wrote portions of the report, while Chris edited it and made sure that it conformed to the checkpoint guidelines.

For the remainder of the project, Arthur and Chris reprocessed the data after a group consensus that the original data processing was not working well enough. Ben was in charge of using this new data to retry older methods and to get updated plots using the K-Means clustering, while Chris and Arthur tinkered with various deep neural networks in an attempt to get better results. The report was once again a team effort, with help from each member in collecting figures and summarizing our findings and process.

References

[1] Abdel-Aty, Mohamed, and Anurag Pande. “Crash Data Analysis: Collective vs. Individual Crash Level Approach.” Journal of Safety Research, National Safety Council and Elsevier, 22 Oct. 2007, www.sciencedirect.com/science/article/pii/S0022437507001065.

[2] Basic PSO Parameters. https://web2.qatar.cmu.edu/ gdicaro/15382-Spring18/hw/hw3-files/pso-book-extract.pdf

[3] Brownlee, Jason. Ensemble Machine Learning Algorithms in Python with scikit-learn. Machine Learning Mastery. 3 June 2016. https://machinelearningmastery.com/ensemble-machine-learning-algorithms-python-scikit-learn/

[4] Brownlee, Jason. How To Improve Deep Learning Performance. Machine Learning Mastery. 21 Sept. 2016. https://machinelearningmastery.com/improve-deep-learning-performance/

[5] Gugger and Howard. "AdamW and Super-convergence is now the fastest way to train neural nets." Fast AI. 2 Jul. 2018, https://www.fast.ai/2018/07/02/adam-weight-decay/

[6] Simmons, Tien. “Virginia Crashes.” Virginiaroads.org, VDOT, 2 Apr. 2020, www.virginiaroads.org/datasets/virginia-crashes.