The Severity of Car Accidents - How Can we Predict Danger?

Marcus Chiang
October 3, 2020

1 - Introduction

1.1.) Background

Car accidents are fairly common with the average number of accidents per year in the U.S being six million, with a total of ninety deaths a day according to this site. Three million people annually are injured in these accidents, almost one percent of the country's population. This may not sound like a lot, but car accidents are a leading cause of death in the United States, meaning many preventive measures have been taken by the government to ensure driver's safety. The sad truth is we will never be able to completely eradicate car accidents; we can, however, predict and possible prevent a large amount with the right technology and code. With enough data, we can see what variables affect the severity of an accident and warn people.

1.2.) Problem

Using the example dataset of Seattle city, I will locate the data columns that most affect car accident severity and pinpoint their exact conditions, allowing me to recognize which places may be safer and which places may be more dangerous. I hope to be able to determine the risk of any given journey with just things like weather conditions, the location, or the junction.

1.3.) Interest

My project will be of interest to many people, especially those frequently driving or living near a place with lots of vehicular traffic. Although the dataset is limited to Seattle at the moment, with enough data from other cities or towns, the model should be able to predict and work just as efficiently. In order to stay safe, many drivers would utilize the predictions to avoid routes that could potentially pose risk of danger or otherwise.

2 - Data

2.1.) Data Sources

I used the Data Collisions dataset introduced in week one of the Applied Data Science Capstone. I thought very briefly about trying to find one of my own, but as someone who's only experience with data science is this particular course, I decided to simplify things as much as possible by using a dataset many others have used, in case I had any questions. Although the data may be slightly outdated, I checked the overall statistics of the frequency and severity of accidents, and they remained not too unlike the example dataset. It's significant enough of a difference to possibly affect the outcome of my model, but with a more recent dataset my project should work fine. I imported the data frame along with several libraries such as pandas for the data, matplotlib for the visualization, sklearn for AI, and numpy for math. Using seaborne to see which data columns correlated the most with the accident severity, I was now ready to begin my project in full.

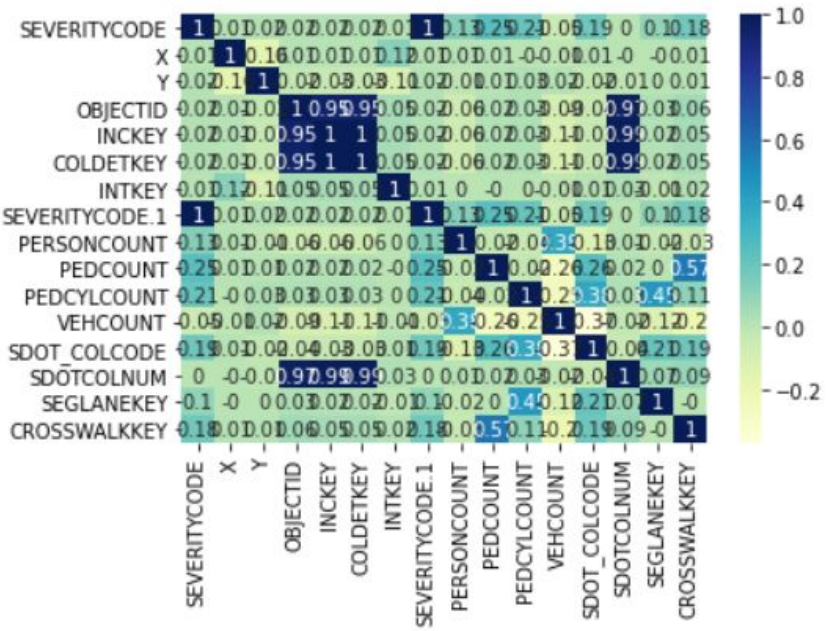


Figure 1. The Correlation Chart Created Displaying the Correlations to SEVERITYCODE

3 - Methodology

3.1.) Data Cleaning

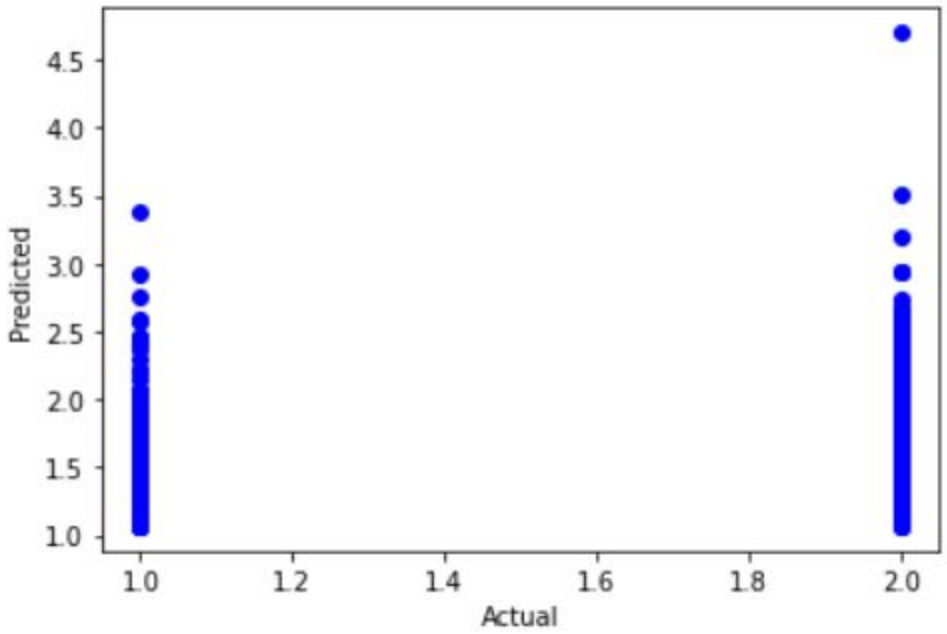
Immediately observable from the correlation chart were the five main factors that determined accident severity. PEDCOUNT, the amount of pedestrians, PEDCYLCOUNT, the amount of bicycles in the a accident, SDOT_COLCODE, the code given to the accident by the Seattle Department of Transportation, PERSONCOUNT, the amount of people involved, and CROSSWALK KEY, the key of the crosswalk the accident occurred at. After this, I replaced all NaN values with the mean of the data to make sure my regression model would run smoothly.

3.2.) Model Fitting

I defined data frame X and Y to begin training the model, which originally gave me some trouble. The model refused to fit, displaying NaN errors, something that confused me. Reaching out to friends I knew that had previous experience from Python, I was able to discover that I had accidentally moved some cells, preventing the model from being able to fit. After that, I trained the X and Y variables, which resulted in an RMSE score of 0.425 and a R2 score of 0.1333, which were less than ideal results. Nevertheless, I decided to move on to testing.

3.3.) Model Testing

The testing process showed similar results to the training process, with both scores improving slightly. The RMSE score increased to 0.427, while the R2 score increased to 0.135. Using matplotlib.pyplot to plot the graph, it ended up like this:



At this point, I became slightly confused, but decided to finish by testing specific inputs and outputs. Creating a set for each one of the data columns that correlated the most

with accident severity, I plugged in the numbers of an actual accident to see how accurate the model was.

4 - Results

4.1.) Final Project Results

Unfortunately, the model was only semi-accurate. Out of the six attempts I tested, almost all of them were a significant one or two points off. As part of the final return code, I changed the final prediction cell to divide the product by three due to an error in my R2 section, which greatly improved the model's accuracy. Testing against preexisting accidents and new ones similar to those, the project wasn't as accurate as I hoped it to be, but I had run out of time.

5 - Discussion

5.1.) Errors

As someone who's only experience with Python and even coding in general was the IBM Data Science Professional Certificate course, the capstone project was a difficult final test. I often worked with friends or family members that were more experienced than I, and even then, many code errors popped out, forcing me to deal with them. I had some trouble working to create the correlation model and training the data, but as I mentioned before the biggest problem I had was fitting my regression model. I spent several days attempting complicated solutions I couldn't comprehend before eventually realizing (with the help of several friends) why my code wasn't working. After that, however, the project went much smoother.

5.2.) Proceedings of the Capstone

As I started this final course, I severely underestimated how much work the project would require and ended up rushing to turn in the assignment late. On top of schoolwork, the project seemed impossible, but with help I was able to finish. If I had more time, I would have liked to improve the model, or even possibly use a new dataset with something slightly more interesting to me than car accident severity.

6 - Conclusion

Although the model may be inaccurate, it certainly can tell us a lot about the severity of any given accident. With the PEDCOUNT, PEDCYLCOUNT, SDOT_COLCODE, PERSONCOUNT, and CROSSWALKKEY data columns, the code I wrote could predict and give a rough idea of the severity of a car accident. The code used in the project and information learned could certainly be improved and built upon with more time, perfecting the model that I started.