**IBM Data Science Capstone Project: Traffic Accident Severity Classification**

**Introduction/Business Problem**

In the United States alone, an average of 6 million car crashes occur annually. This results in over 37,000 deaths each year, making it the leading cause of death for healthy U.S. citizens. In addition, auto accidents cost the U.S. over 200 billion dollars per year. Ironically, these tragedies are mostly preventable, with over 90 percent occurring because of driver carelessness. For many decades, government officials have worked to remedy this issue. Despite their efforts, the problem of severe car accidents is still extremely prevalent today.

In this project, we will build a machine learning model based on various driving conditions to predict the severity of a given car accident. This model can be used to alert drivers of potentially dangerous road situations and encourage them to drive more cautiously based on the level of risk predicted. While this approach is unlikely to eliminate the issue altogether, by understanding the factors that contribute to the severity of a car crash, local officials can concentrate their efforts on building a data driven solution to the problem.

**Data Description**

The data that will be used for this project is titled *US Accidents (3.5 million records)* by Sobhan Moosavi, from Kaggle. It contains information on accidents during the years 2016-2020 spanning the united states. The data was collected using APIs that source data from the US Department of Transportation and law enforcement agencies across the country. There are over 3 million entries, with 49 columns to describe the report. Some of the attributes that will be used for this project include:

* **Location:** Street, City, County, Zipcode, State
* **Weather:** Temperature(F), Wind\_Chill(F), Humidity(%), Wind\_Speed(mph), Precipitation(in)
* **Road Type:** Crossing, Junction, No\_Exit, Railway, Roundabout

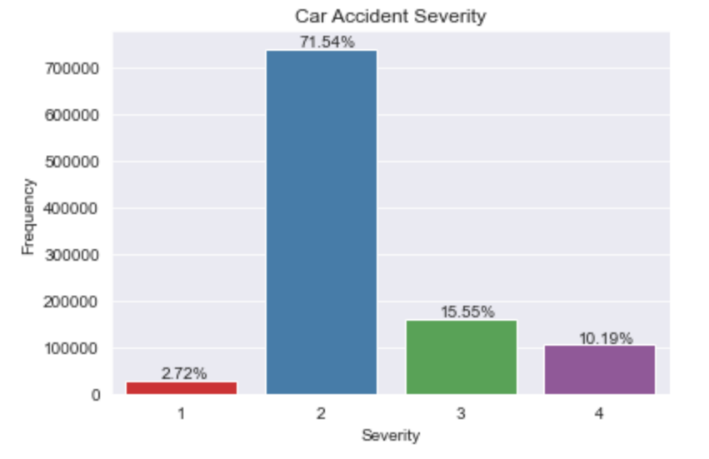
**Methodology**

1. **Data Preprocessing**

I began the process by extracting only the necessary information from the dataset. This meant I only included columns that could be used to predict the severity of a car accident before it occurred, and I dropped any rows where severity was not recorded. In addition, I only included accidents that came from the source “Bing” as different sources categorize severities in different ways.

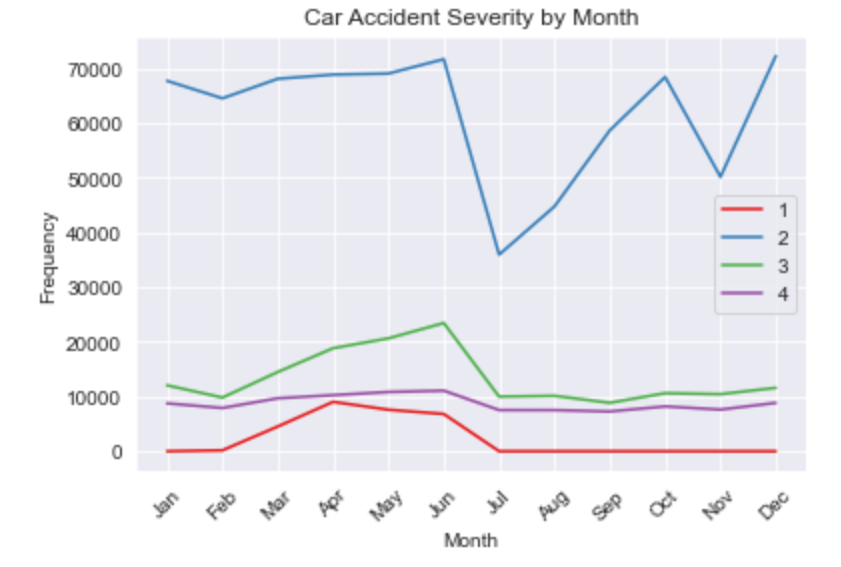
The only minor change that needed to be made for the data to be explored had to do with the Start\_Time column. This feature represented the time the accident started, represented as a string in YYYY-DD-MM HH:MM:SS format. I first converted this column to datetime, and extracted the month, day, weekday, and time attributes from this. With this out of the way, I was now ready to move on to the exploratory data analysis phase.

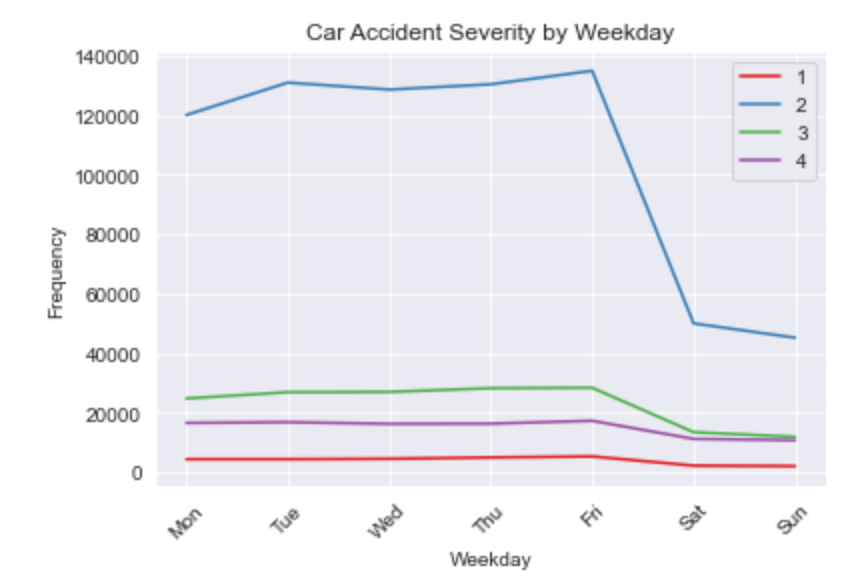
1. **Exploratory Data Analysis**

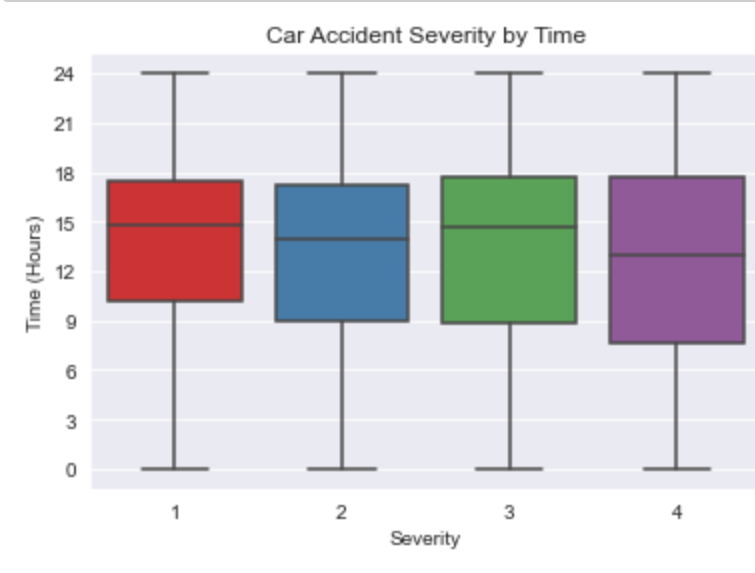
 First and foremost, I plotted the distribution of accident severities in the dataset. The distribution was heavily imbalanced. Roughly 70% of the cases were of severity 2, followed by about 15% of severity 3, 10% of severity 4, and less than 3% of severity 1.

Then, I plotted various features related to the time of the accident. These included month, weekday, and hour of the day. From this, I was able to make a few important insights:

1. The fewest number of cat accidents seem to occur at the beginning of the summer
2. Most car accidents occur during weekdays
3. The median time for a car accident to occur seems to be in the afternoon, around 2 PM.

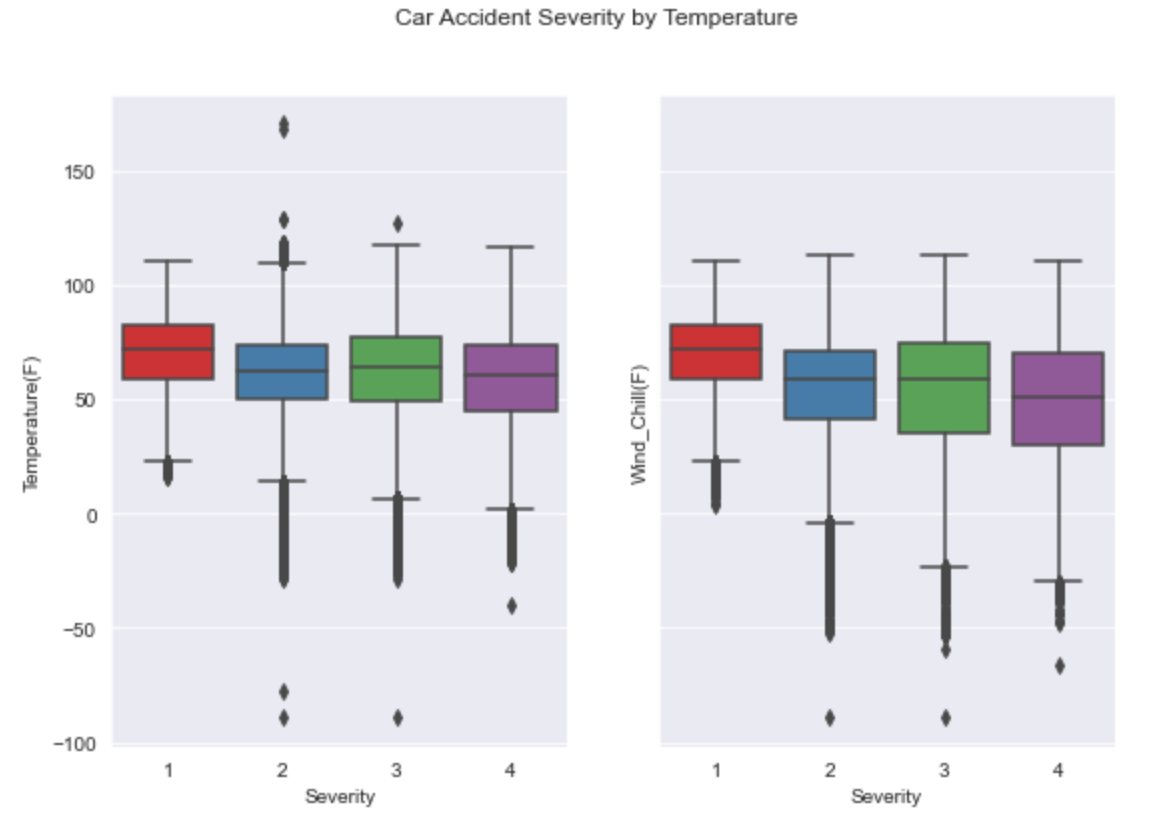


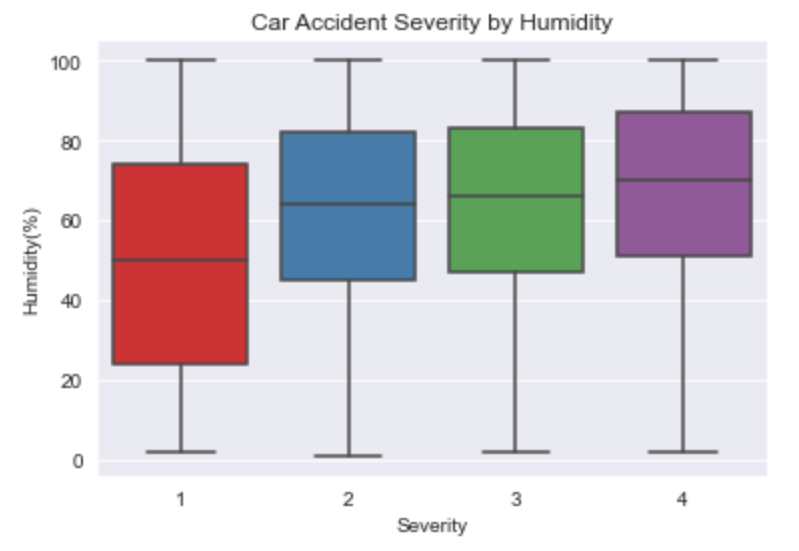


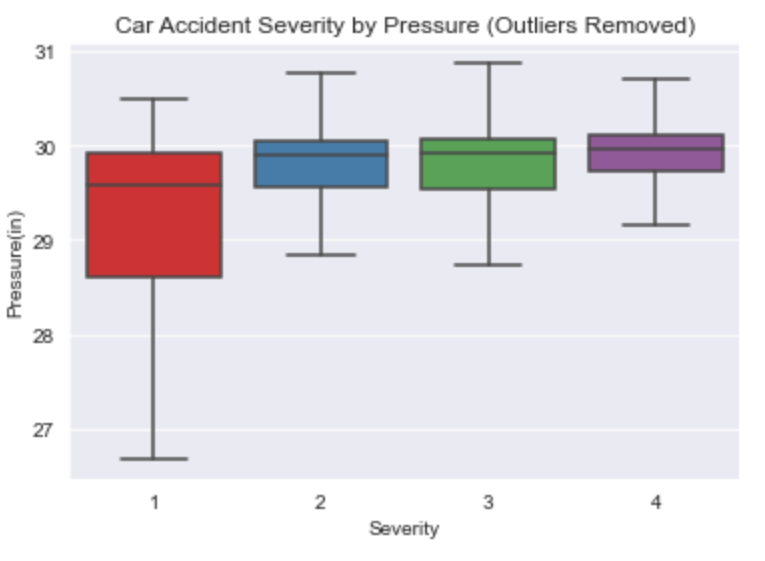


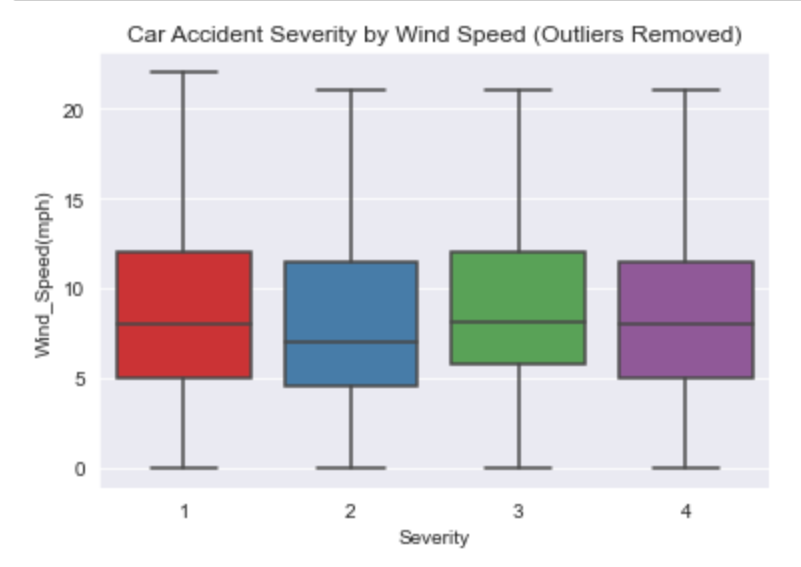
Next, I looked at the columns related to weather. Some of the attributes had huge ranges, meaning that their outliers had to be removed in the plots for the sake of readability. Nonetheless, a few key observations:

1. As temperature decreases, accident severity tends to increase
2. As humidity increases, accident severity tends to increase
3. As pressure increases, accident severity tends to increase



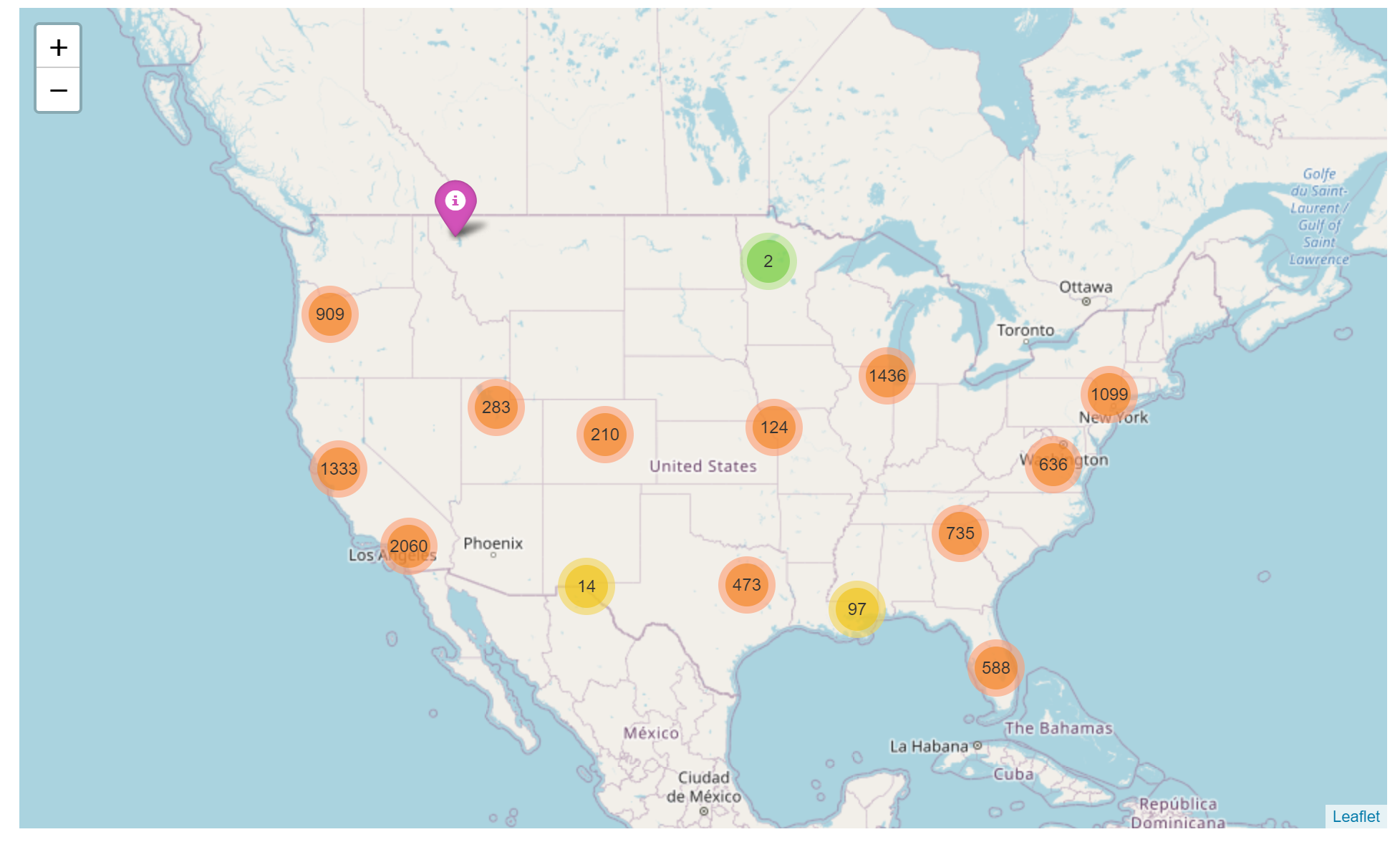


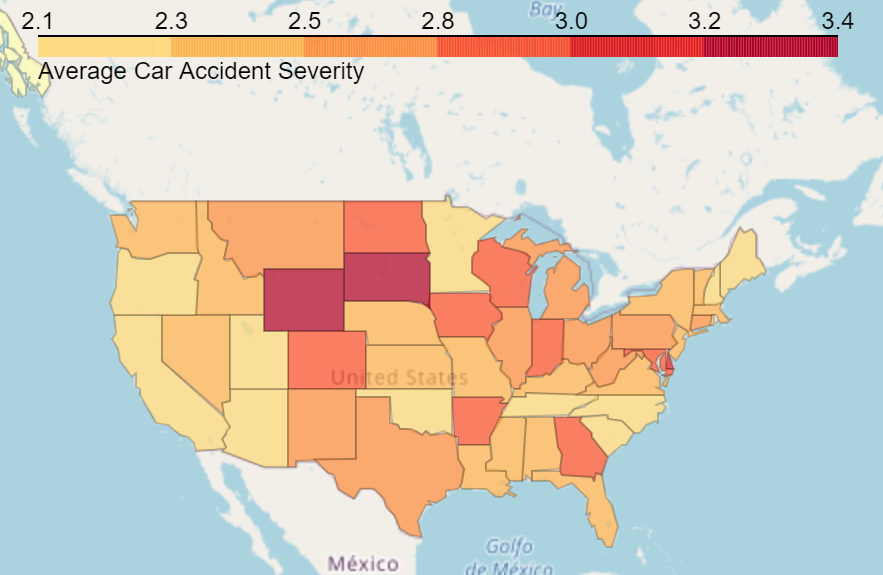


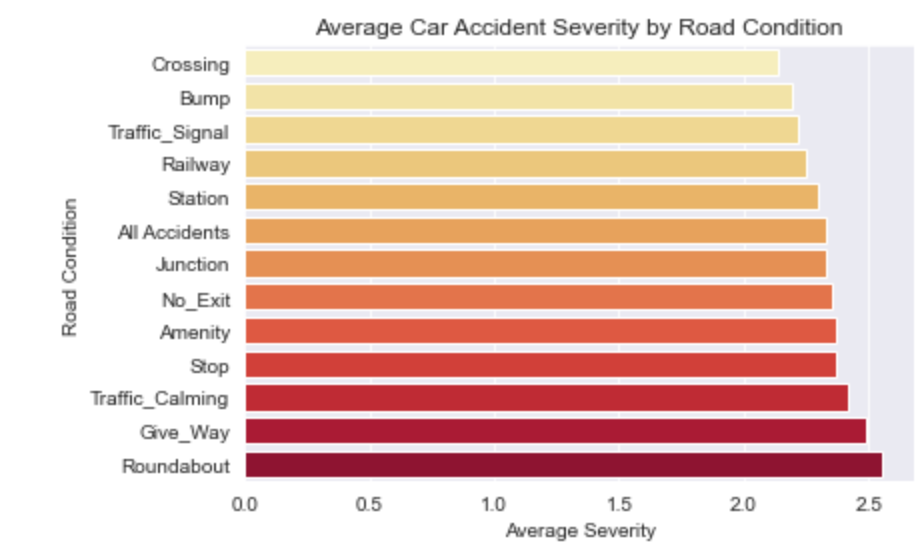


Finally, I explored the location where the car accidents occurred. I created an interactive map showing the distribution of car accidents across the United States, a map showing the average accident severity per state, bar plot showing the average accident severity per road condition. We can see that:

1. Car accidents are clustered towards the coasts, where most of the population is
2. There doesn’t seem to be a clear correlation between geographical region and average accident severity
3. Roundabouts and give ways on average have the most severe car accidents







**3. Data Preparation**

Now that I had a good understanding of the data, I moved on to the data preparation phase. This would begin with feature engineering. I started by diving into the street name feature.

I was surprised about how much information could be gained just from the name of the street. For instance, I was able to extract a direction from many of the streets, as they often contained the words ‘North’ or ‘South’, or sometimes just a character representing this (N, S, W, E). In addition, I classified the highway types of streets where appropriate – I-xx maps to interstate highway, US-xx maps to United States highway, etc. Finally, I grouped streets by common abbreviation like Rd, Dr, St, Fwy, and Expy.

Then, I revisited the Start\_Time column. From this, I created a new feature signifying whether a car accident occurred on a US holiday or not. I also created another feature that combined Start\_Time and Weather\_Timestamp to determine the elapsed time between the observance of a weather condition and the beginning of a car accident – I called this column Time\_Elapsed\_Weather. After this, I dropped the Start\_Time and Weather\_Timestamp columns as I extracted much from these columns already, and I believed they would not be useful on their own.

Finally, I combined the Wind\_Direction column and the street direction columns I created earlier to determine if the cars faced a headwind, tailwind, or sidewind during the accident. I did this by comparing the primary and secondary cardinal direction from Wind\_Direction against the direction of the street. Due to this method of calculation, it was possible for some accidents to have faced 2 of these at once. This concluded the feature engineering phase, and now it was time to deal with missing values in the dataset.

To start with the missing value handling, I choose the incomplete columns related to location – City, Airport\_Code and Timezone. It did not make sense that these were missing, as every location must have a closest city, airport, and time zone. Luckily, the columns for longitude and latitude of the accident were complete, so I used K-Neighbors Imputer to fill in the missing data based on the nearest accident.

Then I moved on to the weather-related features. I filled the categorical columns with missing data – Weather\_Condition and Wind\_Direction – with ‘None’. These would be difficult to guess based on the other features, and the consequences of misclassifying these would likely be great, so the missing values were given a label all to themselves. Perhaps labeling them as such will yield some predictive power if these groups have attributes in common. Next, I imputed the continuous weather features – such as temperature, humidity, precipitation, etc. – based on mean of a subgroup. The subgroups were based on State, County, Month, Day and weather condition. Finally, I filled in missing values for Time\_Elapsed\_Weather with 0s. This value was missing when Weather\_Condition was missing. I figured if no condition was reported, it makes logical sense that the difference between the time of the accident and the observance of the weather condition was 0.

The final columns that needed to be imputed were the astrological based ones – Sunrise\_Sunset, Civil\_Twilight, Nautical\_Twilight, and Astronomical\_Twilight. These features have values of either Day or Night. Very few of these were actually missing (about 60 in each column), so I filled these in with Day, which represented the vast majority of the entries. And with that, all the columns in the dataset were now complete.

With over 1,000,000 rows in the dataset, the sheer size of the dataset would likely cause many computational troubles when fitting it to machine learning models. To deal with this, some dimensionality reduction would need to take place:

1. I would use a sample of 50,000 randomly selected rows to be used for machine learning
2. The Street and Zipcode columns would be dropped. Combined, these represent almost 200,000 unique values. We already extracted lots of information from the street name, and we should have a pretty good estimate of accident location based on the State and County columns, so I wouldn’t expect for too much information to be lost

All there was left to do was standardize the continuous columns and dummy the categorical columns, and split the data into a train and test set. After doing this, the data was all ready to go for machine learning.

**4. Machine Learning**

The scoring metric that will be used for this task is f1\_macro. This task is a multi-classification, with heavily imbalanced labels, so this should be an appropriate function. The machine learning models chosen for this project are:

1. Logistic Regression
2. K-Neighbor Classification
3. Random Forest Classification

I have chosen these models because they all support classification, and they represent a wide variety of model types. For instance, Logistic Regression is a linear based model, K-Neighbor Classifier will make predictions based on similarity, and Random Forest Classification uses a Decision Tree to make its classifications.

**Results**

The best performing model for this dataset is Logistic Regression. It scores an average macro score of .46, with an accuracy of .59. While guessing a severity of 2 for all accidents would result in a higher accuracy, it would misclassify all other severities, hence resulting in a lower average macro score. The second-best model was K-Neighbors Classifier with a average macro score of .4, followed by Random Forest Classification with a macro score of .31.

**Discussion**

When scoring for f1\_macro, the machine learning models made a tradeoff between average macro score and accuracy. Essentially, the model was able to correctly identify more outlier severities by sacrificing some overall accuracy. Because of this, the results for this project would be the perfect fit to build a driver alert system, as it would more liberally assign accidents of higher severity, thus encouraging drivers to be safer. Also, this model may be useful in predicting future severe accidents in sections of the country that are considered safe. Perhaps false predictions of high accuracy may be a warning sign to local officials that with future increased traffic, the risk of severe accidents is high.

**Conclusion**

I hope that the results of this project will yield valuable information that can contribute to life saving policies and technologies. I think certain ways to expand upon this project would be for more data to be collected related to speed limits and amount of traffic during an accident. In addition, the use of a powerful cloud computing system to aid in the machine learning process would have been of great assistance. The 1,000,000 rows of data we originally had was reduced to 10,000 to save on memory and computation time. Using the full dataset, perhaps even more powerful predictions could be made.

**Acknowledgements**

<https://www.kaggle.com/sobhanmoosavi/us-accidents>

<https://www.thewanderingrv.com/car-accident-statistics/#:~:text=Annual%20United%20States%20Car%20Crash%20Statistics,-This%20section%20of&text=On%20average%2C%20there%20are%206,22%2C471%20caused%20only%20property%20damage.>

<https://arxiv.org/abs/1906.05409>

["Accident Risk Prediction based on Heterogeneous Sparse Data: New Dataset and Insights."](https://arxiv.org/abs/1909.09638)