# Analysis and Prediction of US Traffic Accidents

## Problem Definition

Traffic accidents every year have been causing severe societal and economic losses to the United States(US) amounting to about one trillion dollars, particularly the serious ones. Hence, if the key factors and the patterns of how the factors affect the severity of accidents could be identified, we will be able to predict accident severity and location ahead, which are possible to make contributions to society.

Therefore, researching the background of US car accidents in depth and figuring out key factors influencing accidents' severities would be the first problem we try to solve. And developing a machine learning model to predict the severity of the accident accurately would be the second. Since some factors that might influence the severity of accidents such as durations, distances, vehicle types, and drivers' attitudes and behaviors vary significantly among accident cases and are hard to obtain before the accident happened, these factors may not be a good choice for a prediction model. So we will only choose the observable factors that do not change with every accident as our model's features.

The main datasets we use: a) US Accidents (2016 - 2020)[1], which is offered as a CSV file. The dataset covers traffic accidents in 49 states of the US from February 2016 to December 2020 and provides streaming traffic crashes using two APIs("MapQuest Traffic" and "Microsoft Bing Map Traffic"). There are 47 columns in the dataset including severity, longitude, latitude, time, temperature, humidity, traffic signal situation and many other environmental related factors that would allow us to analyze. b) Population data from the U.S[2]. government.

As the severity of an accident is a highly complex and integrated issue, many factors seem to have an impact on it. Thus, selecting the right features for the machine learning model is the first challenge of analyzing our project problems. The second one is to find a more intuitive way to present the result of the model instead of just demonstrating the accuracy rate or relationship figures.

## Methodology

#### Data Cleaning and Processing

This project utilizes Spark for ETL process. After first loading the raw data CSV file, multiple columns were found containing missing values. To ensure the data quality for visualization and machine parts later, we created data cleaning scripts to modify the original file to replace all of the missing values. We chose Python and Spark because they can handle the large size of data, and the data source is a structured CSV file, which works well with Spark's dataframe and the Struct Schema we define as we need for later analysis.

1: The dataset can be acquired from https://www.kaggle.com/sobhanmoosavi/us-accidents

2: https://www.census.gov/data/tables/time-series/demo/popest/2010s-nationaltotal.html#par_textimage_2011805803

**Data Storage**

After loading the cleaned version of the raw data using Spark, the Accident Analysis project's data exploration scripts then aggregate data into several aspects: address, time, weather, and point of interest. Aggregated results are stored into a shared cluster in the Cloud version of MongoDB. We chose MongoDB for storage because it can scale easily when the data becomes big and we want to learn this new technology. In addition, because this is a team project, we want to store the data to a system that can be accessed and modified concurrently by our team.

**Visualization**

Now the aggregate results are stored in MongoDB, we will use Python to access the data source and use Plotly to create visualizations based on these results. Plotly is easy for us to learn because it has detailed documentation and working examples of all the graph types we need. In addition, we are using Python for all of the scripts except the Front End part, so we want to use Plotly which has powerful Python support.

**Front End**

For the front end web page to demonstrate our results, we used React for development and Tailwind for styling. In addition, Tailwind provides us with ample default choices to make the website stylish in a short time. As all of our results and visualizations are divided into five parts, and all of them have the same format of visualizations, title and descriptions, a lot of the components of the website can be shared and reused, which is why we choose React which can easily abstract repeated used codes into Components.

**Features Engineering**

The features were selected for training models based on the relative importance of each feature by using '. featureImportances' in Pyspark. We first tested 25 features including features related to time, address, weather, and point of interests, and then chose the following 11 most influential features: Year (8%), Month (6%), Weekday (5%), Hour (7%), Start_Lat (18%), Start_Lng (44%), Temperature (1%), Pressure (3%), Humidity (1%), Junction (2.4%), Traffic Signal (1%).

The time-based features, "Month", "Weekday" and "Hour", were numbers but represented categories that had fixed ranges. If we treat these features as continuous features, their values will affect the model weights. Using one hot encoder could improve the model, which could map a category to a binary vector to make the model perform better.

We tried both the undersampling and the combination of undersampling and oversampling method to resample the imbalanced data. A builtin '.sampleBy' function in PySpark and its 'fractions' parameter allow us to assign the proportions of data for different values, which is a good implementation for undersampling. While for oversampling, we used the '.sample' function to duplicate instances by setting the value of its 'withReplacement' parameter to 'True'.

**Decision Tree Model**

One of the main advantages of the decision tree algorithm is it can handle multi-label classification for a large dataset. In addition, the decision tree accepts both numerical and categorical data. The analysis of factors associated with the accident severity levels shows there

is a more complex relationship than a linear one between them. The decision tree is a suitable model for classification and prediction for our purpose.

## Problems

#### Data ETL: from raw data to clean and balanced data

Even though the data was already cleaned by the publisher, we spent enormous efforts to fully understand the data, then realigned and double cleaned it. For example, when we analyzed the number of accidents per year by severity, we noticed that the severity 1 only occurs in data of 2020. We then delved deeper into the different years' data by month and found that there are only 2 accidents occurring in July 2020 while the amount data in any other years' months looks normal. Based on the results above, to eliminate these kinds of outer points' influence, we decided to narrow down our data by dropping the data in 2020. We also lowered all values in the 'Severity' column by 1 to better demonstrate and interpret results.

Besides, even though we obtain a high accuracy from the first version of the machine learning model, out of 68% (95929/140188) predicted results are from severity 1, which has a large bias towards the actual data. After checking the number of accidents classified by different severity levels, we found that severity 1 contains more than 70% (510929/708242) of the data. Thus, we confirmed that there is an imbalanced classification problem in our model. We came up with two solutions to this problem: undersampling(randomly deleting instances from the over-represented class) and the combination of undersampling and oversampling(randomly duplicating instances from the under-represented class). And the prediction results become more precise after applying resampling methods.

#### User Interface

Prior to the start of this project, none of the team members have previous experience of web development, so coordination and technical challenges arise when we design and implement the web UI for the project. To solve this, we searched many existing solutions online and decided to use React and Tailwind for our system because they both have detailed documentation and community support for any questions encountered. To ensure better coordination, we designate one of the team members mainly responsible for the front end related tasks, and that member specifies the data sources needed from the team to develop the front end system. We believe that by delegating a specified person instead of everyone developing and presenting the front end separately, all the team members know what end results are needed from each other as well as avoiding repeated work.

## Results

#### Location Analysis

Cities differ significantly both in severity and accident counts, Los Angeles for example has 20,054 records which is almost as twice as the second most accident city Charlotte, but LA's average severity is only 1.13, which is a lot smaller compared with the city with highest average severity Griffin's 2.525. However, as a categorical feature, there are two many distinctive values for cities, so if such location information is aggregated to State level, a similar pattern can be found: California's records are almost four times as much as the second State Oregon, and both

of them are in the West Coast. Timezone aggregation also confirms such a pattern: about 70 percent of the total accidents take place in the Pacific time zone and Eastern time zone.

**Weather Analysis**

Having winds in any direction would increase the proportions of having a serious car accident when it compares with the Clam air; The density distribution of level 1 accidents is different from the other severities in both temperature and humidity, especially for the area below the first quartile in temperature and the area above the third quartile in humidity. This illustrates that a serious accident is more likely to occur in low temperature and high humidity in comparison with other temperature and humidity conditions; The proportion of level 3 accidents increases as weather changes from clear & cloudy (10.8%) to rain (11.2%) to fog (14.1%) to storm (16.2%) to snow (17.2%). The results somehow confirm the conclusion we draw from the previous analysis: the occurrences of level 3 accidents are more frequent with increasing humidity and decreasing temperature. Besides, it seems that level 1 accidents would occur in a greater range of pressure than the others. Wind speed and visibility do not have clear impacts on the accident's severity.

**Time Analysis**

The data show the following: (1) Winter accidents are significantly higher and peak in December. (2) The number of accidents in winter is significantly higher in the north (latitude > 40) of the eastern US, while the number of accidents in the south (latitude < 35) does not differ significantly across seasons. (3) Traffic accidents are higher on most working days of the week (Tuesday to Friday). To our surprise, traffic accidents are the least on Monday. (4) There are two peaks in accident counts during the day (7-8 am and 16-17 pm), which correspond to the rush hours we know, as expected. (5) Among the cities with the highest number of accidents, we further uncovered which days had the most traffic accidents in those cities. The results show that Los Angeles had the most accidents (138) on December 23, 2019. We explored the weather on that day and found that 89 accidents (64%) occurred on that day when it was raining.

**Red vs. Blue**

We already see the difference between red states and blue states in the US from taking vaccines, so are they different in driving? We found five states (CA, NY, IL, MA, NJ) that Biden won in the 2020 election, and five states (OH, TX, TN, IN, MO) that Trump won. We do not consider swing states, where the winner of our chosen state receives more than 5% more votes than the loser, and the total number of voters exceeded 2 million in each state. The number of accidents per capita in blue states (0.002741) is twice as high as in red states(0.001022).

**Point of Interest Analysis**

According to the analysis of points of interest, 28.2% of car accidents happened near some points of interest. Most accidents near points of interest occurred at junctions (126,382), followed by traffic signals and crossings (59,709 and 27,961, respectively). Although there were numerous accidents, they had a minor impact on traffic. There were only three car accidents near a roundabout in this dataset, and no incidents occurred near a turning loop. It might be because the roundabouts and turning loops only had one way, which reduced car accidents. But the accidents near roundabouts were more prone to cause significant traffic delays. As a result, the majority of car accidents that occurred near traffic signals or signs had no significant impact on traffic. There

is a relationship between nearby points of interest and the accident severity, though points of interest may not be a critical factor affecting traffic.

**Severity Prediction and Model Implementation**

The features included in our machine learning models are: Year, Month, Weekday, Hour, Start Latitude, Start Longitude, Temperature(F), Pressure(in), Humidity(%), Junction, Traffic Signal, which all can be easily accessed before the accidents happen and are not changing completely among different accidents. Besides those observable features, there are many other factors that influence an accident's severity, notably drivers' attitudes and behaviors, the driving speed, or the vehicle type, so this could be one important reason why our model's prediction accuracy is 65%.

However, if some convenient and low-cost preventive measures including setting up corresponding warns though map apps and traffic broadcasting and temporary placing warning signs for areas where level 3 accidents to be expected to happen can be implemented beforehand to arise driver's attention and remind them of safety driving, we believe that a number of incoming serious accidents would be warned or even avoided through this way. While the cost of implying those preventive approaches are fairly small when compared with the large losses caused by severe accidents.

Although our model's prediction accuracy is not very high on average, in some areas (based on the points in map graph), for example like the east side of the US, we have a considerably high precision of the prediction. Hence, when our model is applied in the eastside of America, the accuracy of the model's prediction accuracy would be over 65% and even reached. So it is more suggested to apply our model in those east-side areas of the US.

## Project Summary

- Getting the data: We downloaded data from Kaggle, and also combined it with other data sources such as population and election results, which are also found online. 2
- ETL: We first realigned and double cleaned the data using Pandas and Pyspark and then performed Extract-Transform-Load work with Pyspark. 3
- Problem: We researched the background of US car accidents in depth, figured out key factors influencing accidents' severities, and developed a machine learning model to predict the severity of the accident. 3
- Algorithmic work: We used machine learning techniques to build our prediction model. 3
- Bigness/parallelization: We used a shared cluster in the Cloud version of MongoDB to store data. 1
- UI: We wrote a frontend web app in React to display our results running locally. 3
- Visualization: We used Plotly to present our results with various forms of diagrams. 3
- Technologies: React, Tailwind, MongoDB, Pandas, Plotly. 2

## References

Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, and Rajiv Ramnath. “A Countrywide Traffic Accident Dataset.”, arXiv preprint arXiv:1906.05409 (2019).

Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, Radu Teodorescu, and Rajiv Ramnath. “Accident Risk Prediction based on Heterogeneous Sparse Data: New Dataset and Insights.” In proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2019.