Predicting the severity scale of a potential car accident

Yassine Fahim

September 19th, 2020

I. Introduction

1. Background

Car accidents are terrible events that can take someone's life or cause significant delays. The severity is a factor that measures the impact of a car accident on the traffic. So, anticipating the severity of a potential car accident can be a major step to save lives and time. Car drivers can be more vigilant in areas where the predicted severity is higher than usual because some POI features such as speed bump, traffic signal might not be present, or weather conditions.

2. Problem

Car accidents can be reduced by predicting their severity. When driving in dangerous areas under dangerous weather conditions, car drivers can be alerted with the severity of a potential car accident so that they can reduce the risk by applying safety measures. The weather conditions, time features such as day of the week or night/day are very important features in predicting the severity of an accident. Moreover, there're other point of interest (POI) features related to the environment of the car such as bumps, crossings, and traffic signals that can predict the severity of the car accident.

3. Interest

Local authorities, highway companies or other traffic related companies may alert car drivers of the severity of potential car accidents to take safety measures by creating user applications that can use this model. In addition, this model can be used through existing user applications such as Google maps or Waze by collecting data about the time, weather, location of the car; thus, predict the severity of a potential car accident at any moment.

II. Data acquisition and cleaning

1. Data sources

The dataset used in this project is a subset of the dataset of 3.5 million traffic accidents that took place in the United States, from February 2016 to June 2020. The dataset was retrieved from Kaggle. It includes 49 different features of the accidents that occurred in the US during this period. There are features related to the surroundings of the accidents such as the crossing, amenity, bump, junction...etc. Also, there're features about the location of the accidents such as the street, city, state, county, longitude, latitude...etc. Moreover, there're features related to the weather conditions in the time of the accident. In addition, there're features related to the time of the accident, and the description of the accident. The dataset includes, of course, our target feature which is the severity.

2. Data cleaning

The data was downloaded to a Jupiter notebook. I uploaded almost 120 thousand rows, which represents almost 42 MB of data. First, I started by dropping duplicate rows, then I dropped the missing values. I ended with a table of almost 60 thousand rows. After that, I dropped the “Turning Loop” as it had only one value. The “Start time” feature was converted to three features: Hour, Day of week, and Month. After selecting the relevant features, I converted them to float values. The next task was detecting outliers, some values were abnormal. For example, having a wind speed of 241 mph in a clear day was inaccurate, or pressure below 28(in). All the outliers were dropped because they represent a risk of bias to the model. Finally, I downloaded the cleaned dataset.

3. Feature selection

The model is destined to work for any car driving in USA at any time. It is a kind of live prediction that can be updated anytime based on USA data such as the location of the car and the weather conditions. The model will predict the severity according to the weather conditions, time features, and some POI features.

The selected features are: 'Severity', ‘Hour’, ‘Day of week’, ‘Month’, 'Temperature(F)', 'Humidity(%)', 'Pressure(in)', 'Visibility(mi)', 'Wind_Direction', 'Wind_Speed(mph)',

'Weather_Condition', 'Traffic_Signal', 'Traffic_Calming', 'Roundabout', 'Sunrise_Sunset', 'Crossing','Amenity', 'Give_Way', 'Junction', 'No_Exit', 'Railway', 'Station', 'Stop'.

III. Methodology

1. Exploratory Data Analysis

1.1. Severity groups:

I started the analysis by grouping the dataset by severity. This way I can inspect how the mean of each feature changes across severity.

Severity	Temperature(F)	Humidity(%)	Pressure(in)	Visibility(mi)	Wind_Direction	Wind_Speed(mph)
1.0	63.596078	64.058824	30.006078	9.411765	15.313725	8.347059
2.0	66.213265	58.586086	29.956070	9.371615	14.448234	8.124448
3.0	66.421100	59.476312	29.962552	9.389343	14.930213	8.298948
4.0	61.845000	68.600000	29.956500	8.800000	15.100000	8.195000

Figure 1: The Accident dataset grouped by severity

1.2. Correlation:

In order to have an insight on the most correlated features with severity in the dataset we can plot a correlation heat map such as:

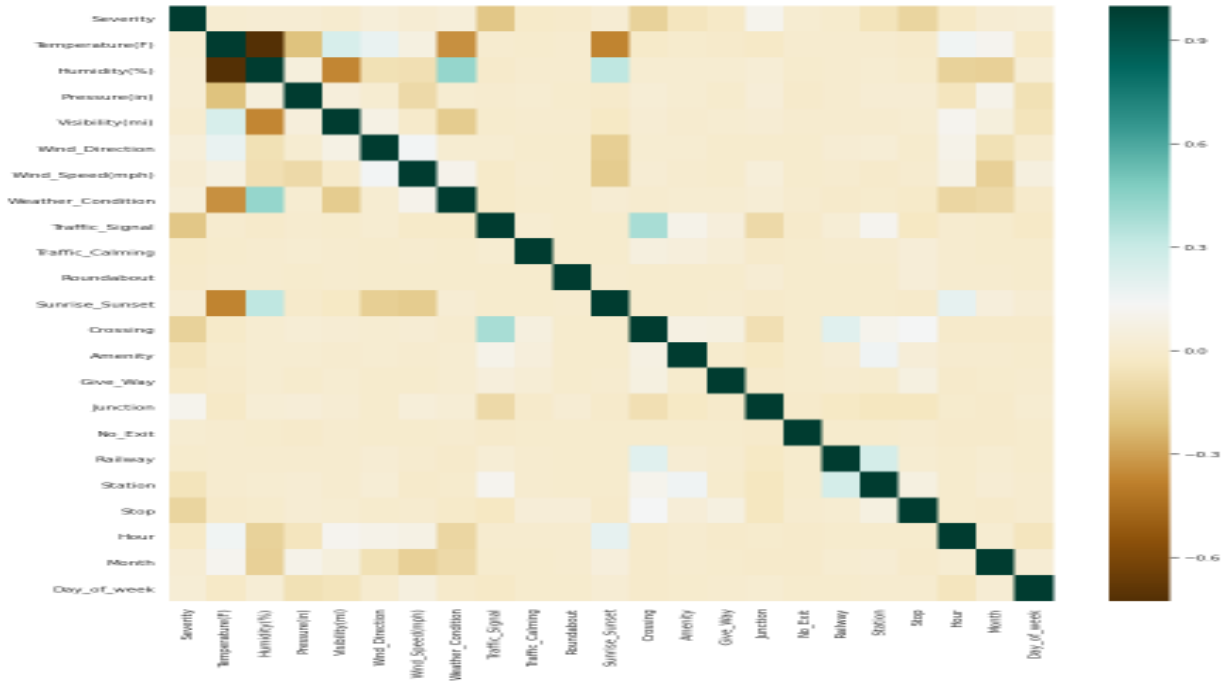

Figure 2: The correlation heatmap of all the selected features in the accident dataset

This table shows the correlation coefficient of each feature with severity in a descending way.

```
Traffic_Signal        -0.186074
Crossing              -0.136795
Stop                  -0.122309
Station               -0.053948
Amenity               -0.048250
Give_Way              -0.019850
Hour                  -0.019750
Traffic_Calming       -0.014488
Roundabout            -0.006266
Railway               -0.003294
Visibility(mi)         0.004249
Temperature(F)         0.008104
No_Exit                0.008733
Pressure(in)           0.015864
Humidity(%)            0.018906
Month                  0.018953
Sunrise_Sunset         0.019692
Wind_Speed(mph)        0.019730
Day_of_week            0.021675
Weather_Condition      0.039891
Wind_Direction         0.040140
Junction               0.103243
Severity               1.000000
Name: Severity, dtype: float64
```

1.3. Time features and Severity:

I analyzed the time features relationship with severity by plotting each feature. For example, the hour in which most of the car accidents occur is at 11am.

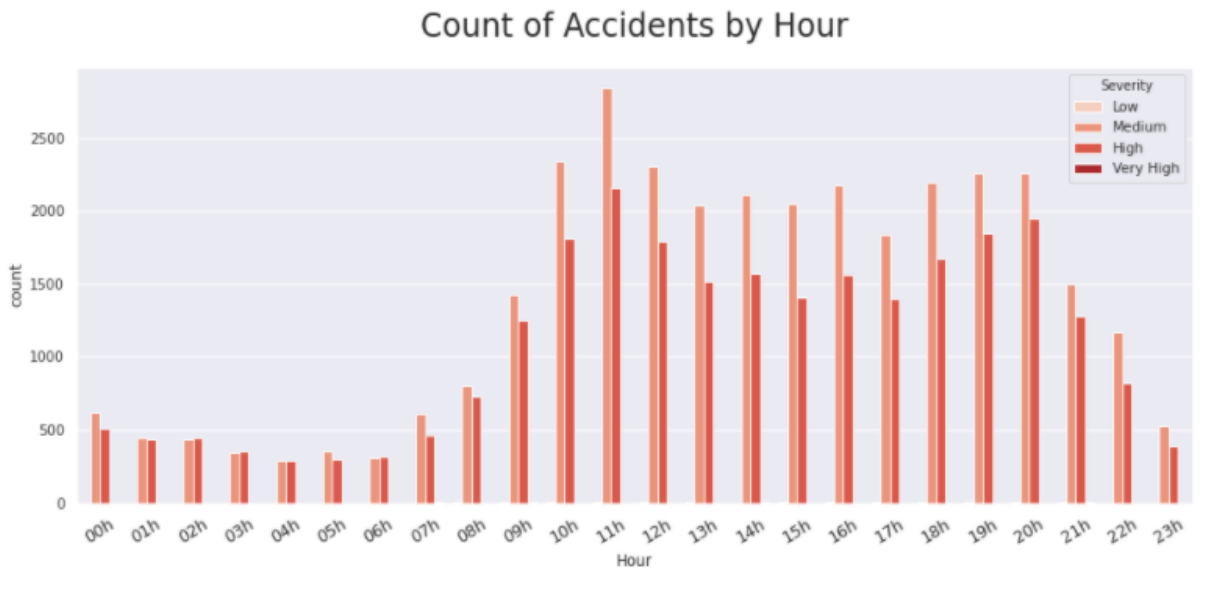

Figure 3: The count of accidents in the dataset by hour and severity

Also, the months in which more accidents happen are January, July, and November.

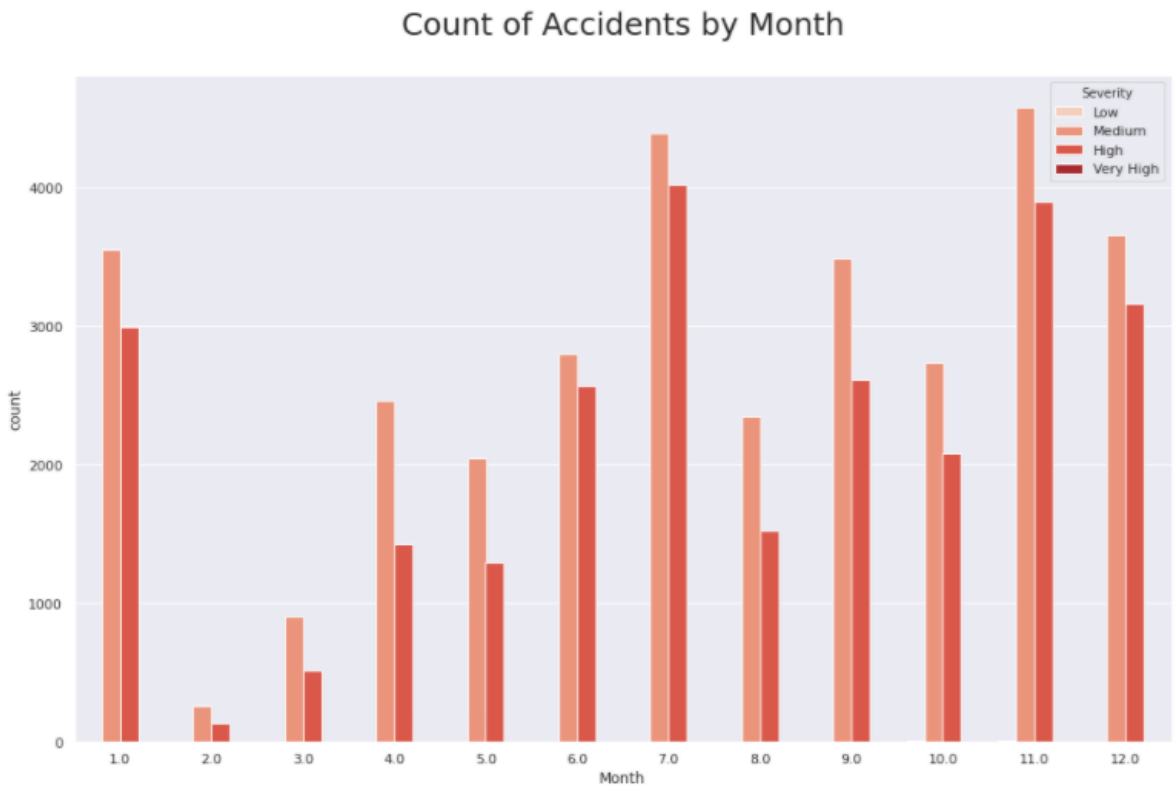

Figure 4: The count of accidents in the dataset by month and severity

Moreover, the days of the week in which most of accidents happen are normal week days, not weekends.

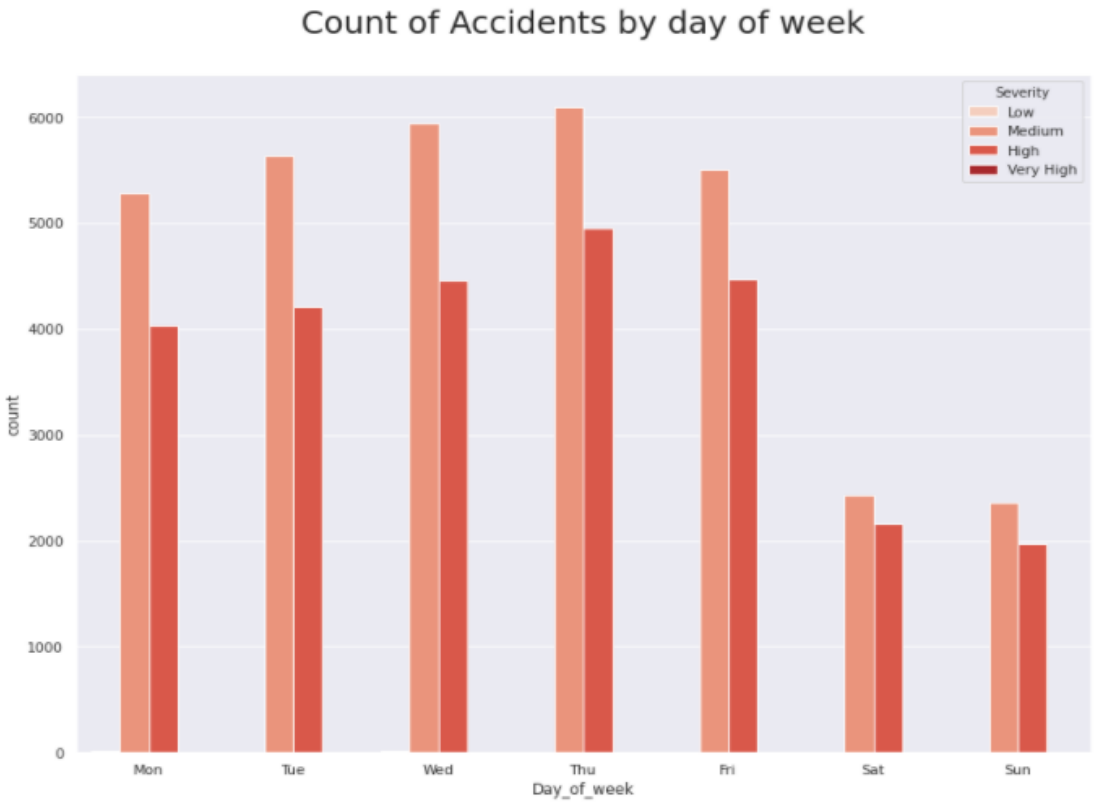

Figure 5: The count of accidents in the dataset by day of week and severity

Another important time feature is the Day/Night based on sunset and sunrise.

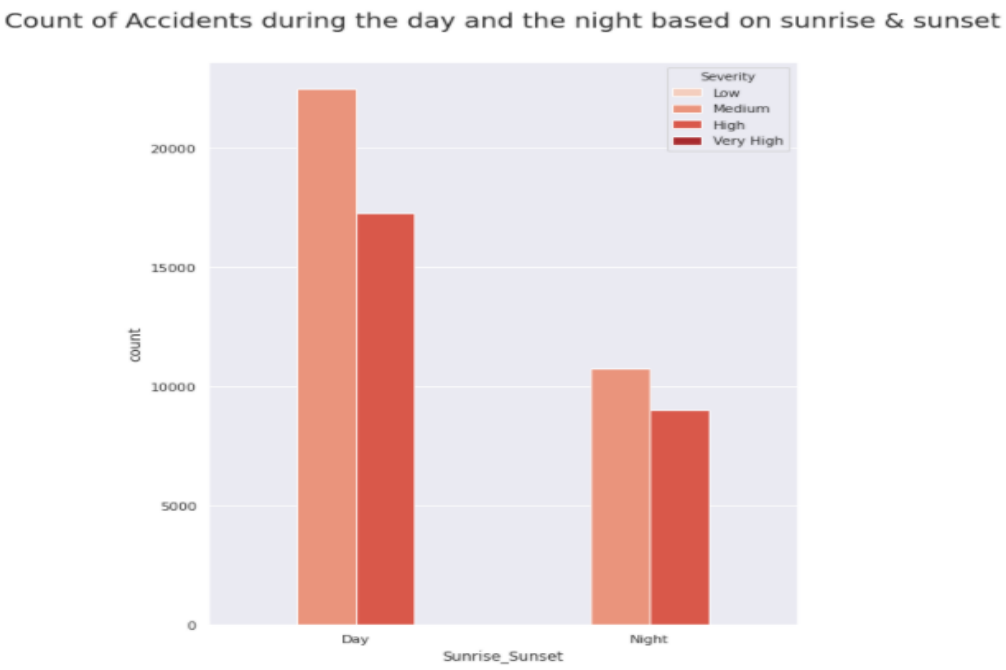

Figure 6: The count of accidents in the dataset by severity and day/night based on the sunrise_sunset

We can clearly see that most of the accidents occur during the day.

1.4. Severity and weather conditions:

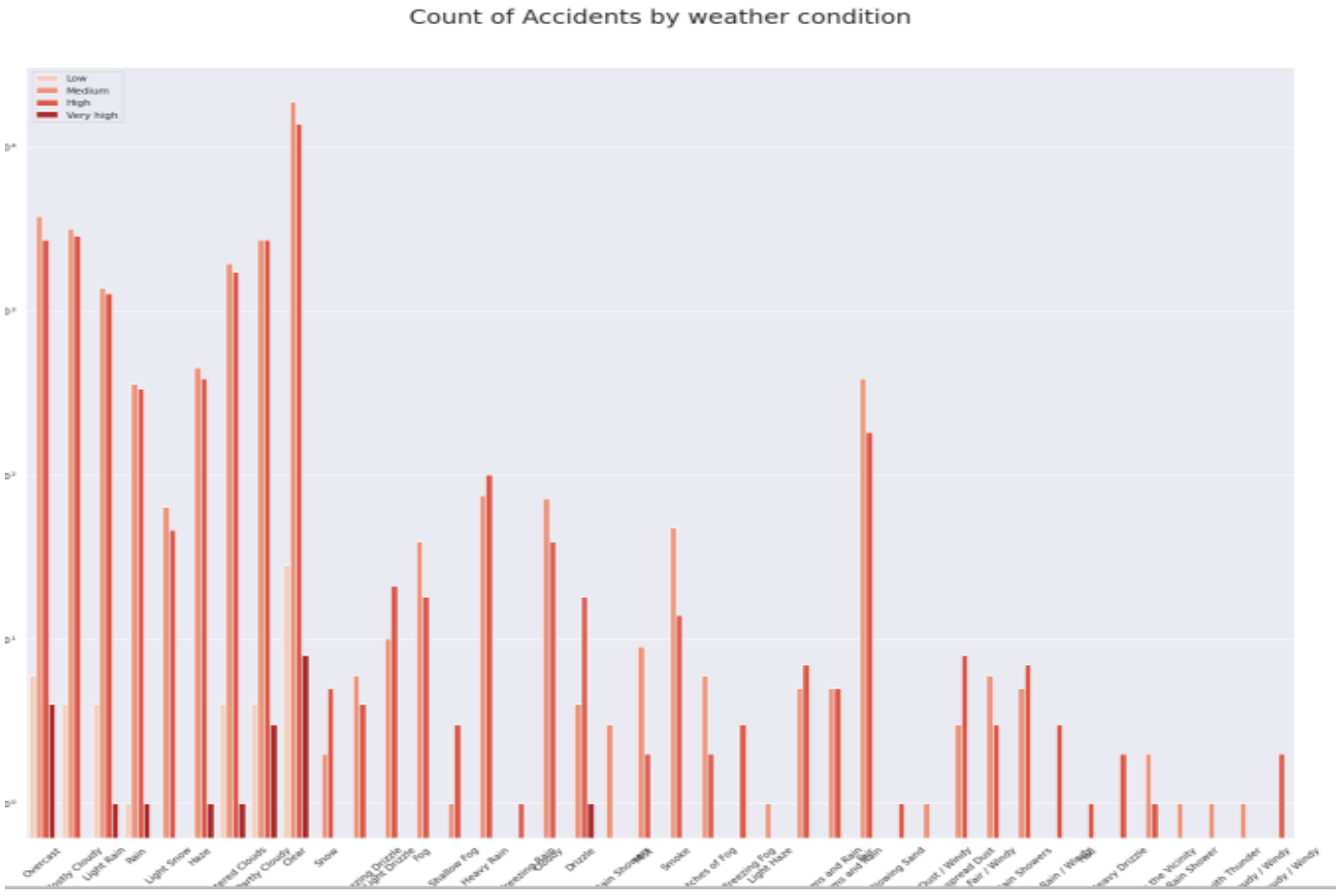

Figure 7: The count of accidents in the dataset by each weather condition and severity

The plot follows a logarithmic scale in the y-axis. It shows that there're some weather conditions that have more accidents than others. Overcast, partly cloudy, Haze… Thunderstorm and rain are weather conditions that witness more accidents. However, we can notice that the clear condition has the biggest number of the severest accidents and accidents in general

1.5. POI features and severity:

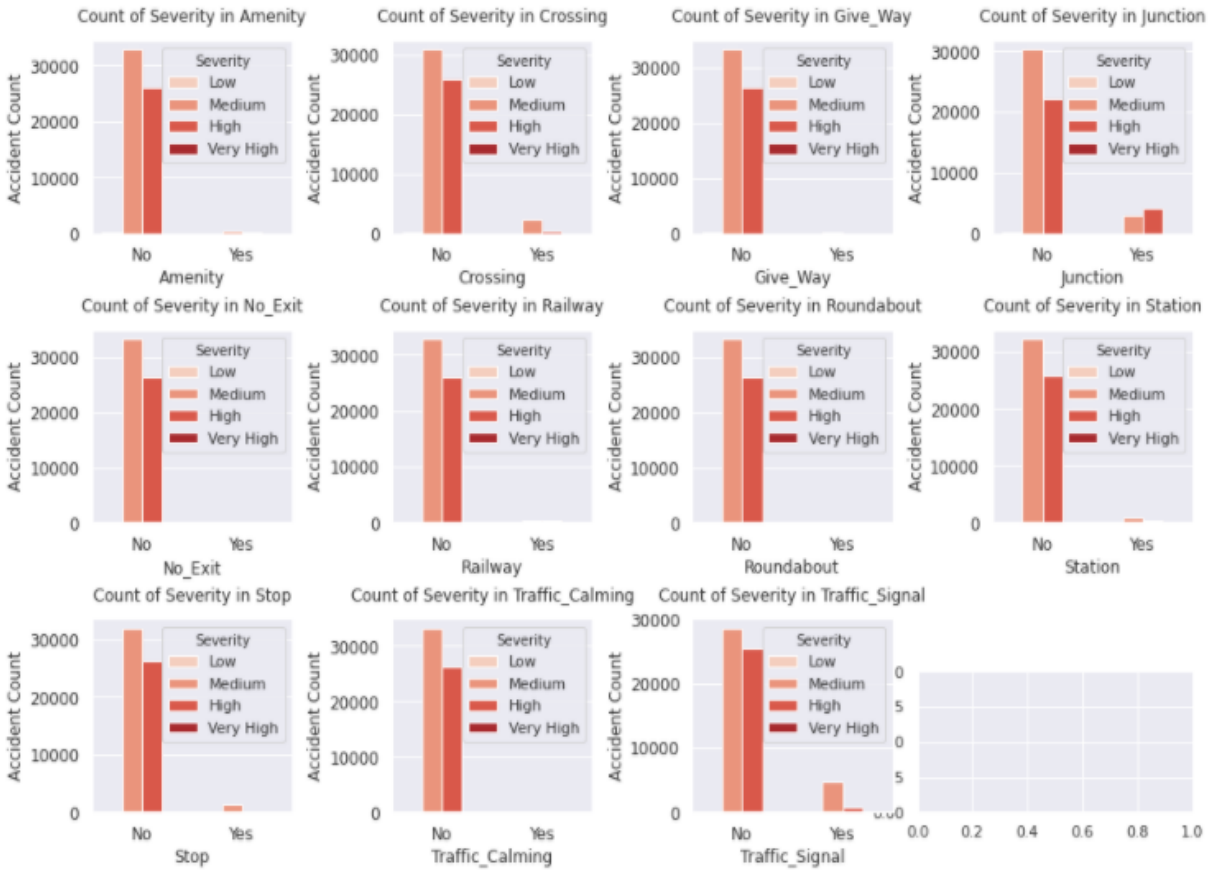

Figure 8: The count of accidents in the dataset by severity and existence of POI in a 5km radius area

We can clearly notice that the presence of POI features is strongly related to the severity of accidents and accidents in general. Almost all the accident happens when there's no POI feature around.

2. The Classification Models:

I started by importing the clean dataset, then by splitting it to a train and test parts. I used 85% of the dataset for train and 15% for test which resulted in the best accuracy scores. The machine learning

algorithms that I used are KNN and Decision tree. These two algorithms fit very well with large data. They can handle the outliers better than other algorithms.

2.1. K Nearest Neighbor (KNN)

We start this model by finding the best k based on its accuracy

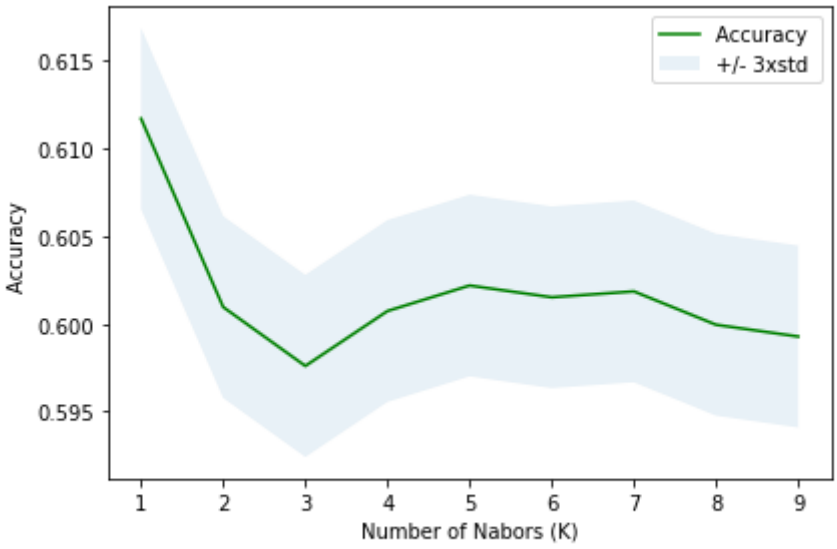

Figure 9: The accuracy of the KNN model using different values for k

After that, I built the model and performed the prediction on the test data.

```
from sklearn.neighbors import KNeighborsClassifier
k = 1
#Train Model and Predict
neigh = KNeighborsClassifier(n_neighbors = k).fit(X_train,y_train)
neigh
```

Figure 10: The python syntax of the KNN model using k = 1

2.2. Decision Tree:

I used the same train and test data on all the models. In the decision tree, there're some hyperparameters that can be tuned in order to obtain the highest accuracy. In fact, I tuned the decision tree model based on the criterion, splitter, and max depth. I obtain the following configuration:

```
from sklearn.tree import DecisionTreeClassifier
DT_model = DecisionTreeClassifier(criterion="entropy", splitter = "best", max_depth = 30)
DT_model.fit(X_train,y_train)
DT_model
```

Figure 10: The python syntax of the Decision tree model

After that, I built the model and performed the prediction on the test data.

IV. Results

1. Performance of the models

Models	Jaccard	F1s
KNN	0.611708	0.611855
DecisionTree	0.601746	0.600777

Figure 11: The Jaccard and F1 scores of the KNN and Decision tree models

Both Models scored almost the same accuracies. 61% for KNN and 60% for the Decision tree. We can compare the two models based on the actual vs predicted values by plotting the count of each column; however, it doesn't give information about the true accuracy.

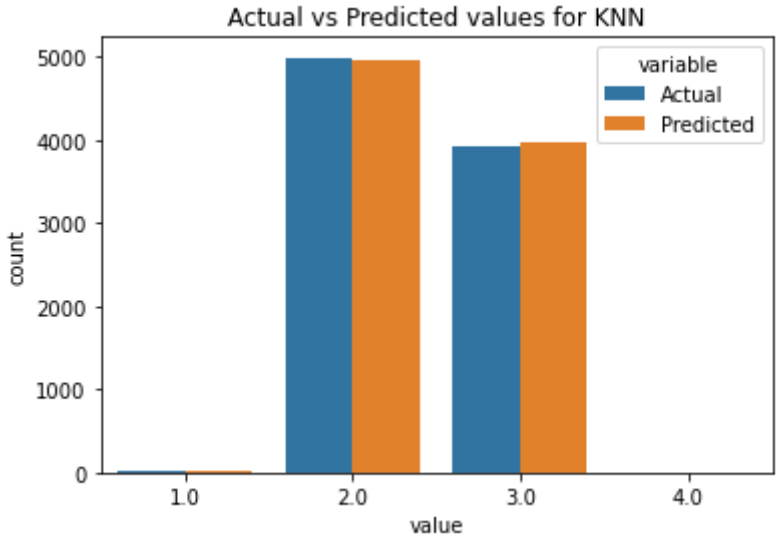

Figure 12: The count of the actual and the predicted values by severity of the KNN model

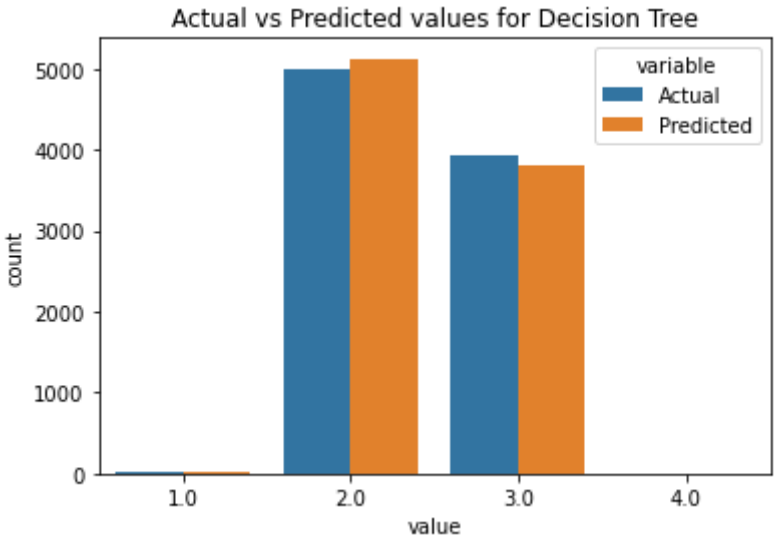

Figure 12: The count of the actual and the predicted values by severity of the Decision tree model

2. Confusion Matrices

The confusion matrix is a good visualization of a classification model. It gives an insight on how the model performs. For example, in the KNN model, 2230 rows were classified as high severity which is correct. The actual values give the same values. However, the model predicted 3 very high severities as medium which is inaccurate. The KNN scored 5465 true predictions, whereas the decision tree scored 5392 true predictions.

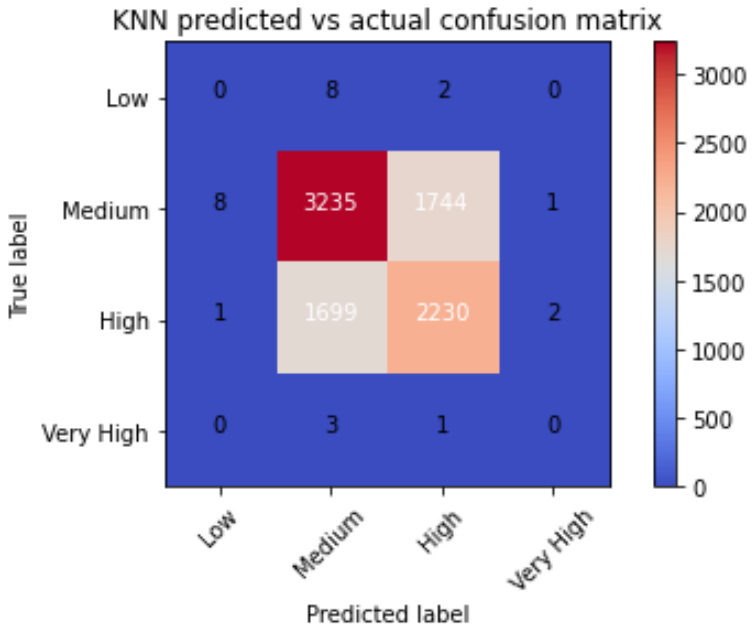

Figure 13: The confusion matrix of the actual and the predicted values by severity of the KNN model

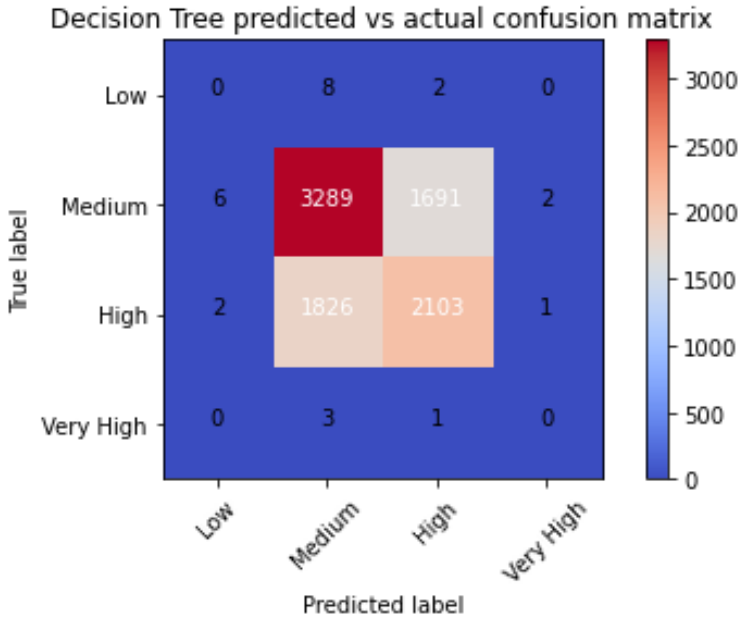

Figure 13: The confusion matrix of the actual and the predicted values by severity of the Decision tree model

3. Discussion

The results obtained by these two models are far from being satisfying. He accuracy should be increased. To increase the accuracy, we can add more data to the models. Also, the computational power wasn't very helpful. The Jupiter notebook interface doesn't allow uploading very big csv files. Another way to increase the accuracy of the models is by adding more relevant features such as the speed at the time of the accident, the type of the car, the state of the car…

V. Conclusions

In this project, I developed a classification model that predicts the severity of car accidents based on time, POI, and weather features. First, I downloaded the dataset of all the accidents in USA from 2016 to 2020. Then, I dropped the duplicates, missing values, irrelevant features, and outliers. After that I analyzed the relationship of time features, weather conditions, and POI features with the severity of accidents by performing a correlation heatmap and descriptive statistics of each feature. After that I visualized the features to get a better understanding. At the end, I built two different models (KNN and Decision tree) to predict the severity of a potential car accident based on those features. The models scored about 61% out of sample accuracy. To conclude, the models need more data and more relevant features to perform better.