US ACCIDENTS

A project report submitted to ICT Academy of Kerala

in partial fulfillment of the requirements

for the certification of

CERTIFIED SPECIALIST

IN

DATA SCIENCE & ANALYTICS

submitted by

Team 10

Anju Sudheendran

Caroline Mary M

Himesh

Johnson Mathew

Preetha M V

ICT ACADEMY OF KERALA
THIRUVANANTHAPURAM, KERALA, INDIA
May 2021

List of Figures

Mean Severity based on Humidity(%)

Mean Severity based on Pressure(in)

Figure 6: Graph Based on Point Of Interest Attributes

Accident Severity Near Crossing

Accident Severity Near Junction

Accident Severity Near No_Exit

Accident Severity Near Roundabout

Accident Severity Near Station

Accident Severity Near Stop

Accident Severity Near Traffic_Signal

Accident Severity Near Turning_Loop

Accident Severity Near Give_Way

Figure 7: Percentage of side of Accidents

List of Abbreviations

Distance(mi) :- Distance in Miles

Temperature(F) :- Temperature Fahrenheit

Wind_Chill(F) :- Wind_Chill Fahrenheit

Pressure(in) :- Pressure Inches

Visibility(mi) :- Visibility Miles

Wind_Speed(mph) :- Wind_Speed Miles Per Hour

Precipitation(in):- Precipitation Inches

Humidity(%) :- Humidity in Percentage

KNN:- K-Nearest Neighbour

Table of Contents

Abstract

Reducing traffic accidents is an important public safety challenge. However, the majority of studies on traffic accident analysis and prediction have used small-scale datasets with limited coverage, which limits their impact and applicability; and existing large-scale datasets are either private, old, or do not include important contextual information such as environmental stimuli (weather, points-of-interest, etc.). In order to help the research community address these shortcomings we have - through a comprehensive process of data collection, integration, and augmentation - created a large-scale publicly available database of accident information named US-Accidents. US-Accidents currently contain data about 2.25 million instances of traffic accidents that took place within the contiguous United States, and over the last three years. Each accident record consists of a variety of intrinsic and contextual attributes such as location, time, natural language description, weather, period-of-day, and points-of-interest. We present this dataset along with a wide range of insights gleaned from this dataset with respect to the spatiotemporal characteristics of accidents. Dataset **US Accidents** include **4232541** records and **49** fields. The classification goal is to predict the severity of the accident

1. Project Overview

Reducing traffic accidents is an important public safety challenge around the world. A global status report on traffic safety notes that there were 1.25 million traffic deaths in 2013 alone, with deaths increasing in 68 countries when compared to 2010. Accident prediction is important for optimizing public transportation, enabling safer routes, and cost-effectively improving the transportation infrastructure, all in order to

make the roads safer. Given its significance, accident analysis and prediction has been a topic of much research in the past few decades. While a large body of research has been focused on small-scaled datasets with limited coverage (e.g. a small number of road-segments, or just one city), the value and impact of predictive solutions may be better studied when using large-scale data. Although some studies conducted their work based on large-scale motor-vehicle crash datasets, their data is usually private or poses strict rules to be shared with outside researchers, which makes their framework and results unproducible. While there are still a few publicly available large-scale accident datasets, their data is either old, limited to one state or one city, or incomprehensive (regarding data attributes or average reports per year). In order to mitigate these challenges and to provide a context for future research on traffic accident analysis and prediction, we present a new dataset, we name it US-Accidents, which includes about 2.25 million instances of traffic accidents that took place within the contiguous United States1 between February 2016 and March 2019. Unlike some of the available large-scale accident datasets, US-Accidents offers a wide range of data attributes to describe each accident record including location data, time data, natural language description of event, weather data, period-of-day information, and relevant points-of-interest data (traffic signal, stop sign, etc.). Very importantly, we also present our process for creating the above dataset from streaming traffic reports and heterogeneous contextual data (weather, points-of-interests, etc.), so that the community can validate it, and with the belief that this process can itself serve as a model for dataset creation. Using US-Accidents, we performed a variety of data analysis and profiling to derive a wide-range of insights. Our analyses demonstrated that about 40% of accidents took place on or near high-speed roadways (highways, interstates, etc.) and about 32% on or near local roads (streets, avenues, etc.). We also derived various insights with respect to the correlation of accidents with time, points-of interest, and weather conditions. A variety of insights gleaned through analyses of accident hotspot locations, time, weather, and points-of-interest correlations with the accident data; that may directly be utilized for applications such as urban planning, exploring flaws in transportation infrastructure design, traffic management and prediction, and personalized insurance.

1.2 Problem Statement

This data science project aims to help data scientists develop an intelligent ML model to predict the severity of the accidents happening,which will aid in reducing the number of accidents occuring.

Accidents are a common cause of death, disability, collateral destruction and a significant public health and road safety problem. Accidents are also a significant cause of congestion and delays in the flow. On average, nearly 5,000 people are killed and over 418,000 people are injured in weather-related crashes each year. Being one of the major steps of accident management, accident severity prediction can provide crucial information for emergency responders to evaluate the severity level of accidents, estimate the potential impacts, and implement efficient accident management procedures.

1.3 Domain and Data: Data Description

A. Source and Size of Data

Dataset link: https://www.kaggle.com/sobhanmoosavi/us-accidents

This is a countrywide car accident dataset, which covers 49 states of the United States. The accident data is collected from February 2016 to December 2019, using several data providers, including two APIs that provide streaming traffic incident data. These APIs broadcast traffic data captured by a variety of entities, such as the US and state departments of transportation, law enforcement agencies, traffic cameras, and traffic sensors within the road-networks. Currently, there are about 3.0 million accident records in this dataset.

1.4 Overview the dataset

Details about features in the dataset:

Traffic Attributes (12):

- **ID:** This is a unique identifier of the accident record.
- **Source:** Indicates source of the accident report (i.e. the API which reported the accident.).
- **TMC:** A traffic accident may have a Traffic Message Channel (TMC) code which provides a more detailed description of the event.
- **Severity:** Shows the severity of the accident, a number between 1 and 4, where 1 indicates the least impact on traffic (i.e., short delay as a result of the accident) and 4 indicates a significant impact on traffic (i.e., long delay).
- **Start_Time:** Shows start time of the accident in the local time zone.
- **End_Time:** Shows end time of the accident in the local time zone.
- **Start_Lat:** Shows latitude in GPS coordinate of the start point.
- **Start_Lng:** Shows longitude in GPS coordinate of the start point.
- **End_Lat:** Shows latitude in GPS coordinate of the end point.
- **End_Lng:** Shows longitude in GPS coordinate of the end point.
- **Distance(mi):** The length of the road extent affected by the accident.
- **Description:** Shows natural language description of the accident.

Address Attributes (9):

- **Number:** Shows the street number in the address field**.**
- **Street:** Shows the street name in the address field.
- **Side:** Shows the relative side of the street (Right/Left) in the address field.
- **City:** Shows the city in the address field.
- **County:** Shows the county in the address field.
- **State:** Shows the state in the address field.
- **Zip Code:** Shows the zip code in the address field.
- **Country:** Shows the country in the address field.

- **Timezone:** Shows timezone based on the location of the accident (eastern, central, etc.).

Weather Attributes (11):

- **Airport_Code:** Denotes an airport-based weather station which is the closest one to location of the accident.
- **Weather_Timestamp: S**hows the time-stamp of a weather observation record (in local time).
- **Temperature(F):** Shows the temperature (in Fahrenheit).
- **Wind_Chill(F):** Shows the wind chill (in Fahrenheit).
- **Humidity(%):** Shows the humidity (in percentage).
- **Pressure(in):** Shows the air pressure (in inches).
- **Visibility(mi):** Shows visibility (in miles).
- **Wind_Direction:** Shows wind direction.
- **Wind_Speed(mph):** Shows wind speed (in miles per hour).
- **Precipitation(in):** Shows precipitation amount in inches, if there is any.
- **Weather_Condition:** Shows the weather condition (rain, snow, thunderstorm, fog, etc.).

POI Attributes (13)

- **Amenity:** A Point-Of-Interest (POI) annotation which indicates presence of amenity in a nearby location.
- **Bump:** A POI annotation which indicates presence of speed bump or hump in a nearby location.
- **Crossing:** A POI annotation which indicates presence of crossing in a nearby location.
- **Give_Way:** A POI annotation which indicates presence of give_way sign in a nearby location.

- **Junction:** A POI annotation which indicates presence of a junction in a nearby location.
- **No_Exit:** A POI annotation which indicates presence of no_exit sign in a nearby location.
- **Railway:** A POI annotation which indicates presence of railway in a nearby location.
- **Roundabout:** A POI annotation which indicates the presence of a roundabout in a nearby location.
- **Station:** A POI annotation which indicates presence of station (bus, train, etc.) in a nearby location.
- **Stop:** A POI annotation which indicates presence of a stop sign in a nearby location.
- **Traffic_Calming:** A POI annotation which indicates presence of traffic_calming means in a nearby location.
- **Traffic_Signal:** A POI annotation which indicates presence of traffic_signal in a nearby location.
- **Turning_Loop:** A POI annotation which indicates presence of turning_loop in a nearby location.

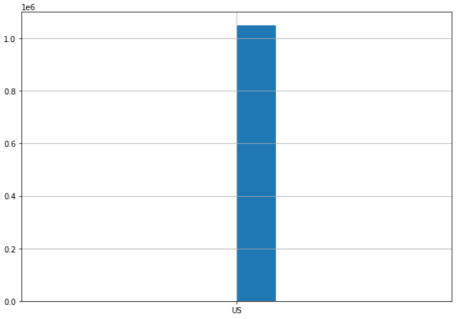
Period-of-Day (4)

- **Sunrise_Sunset:** Shows the period of day (i.e. day or night) based on sunrise/sunset.
- **Civil_Twilight:** Shows the period of day (i.e. day or night) based on civil twilight.
- **Nautical_Twilight:** Shows the period of day (i.e. day or night) based on nautical twilight.
- **Astronomical_Twilight:** Shows the period of day (i.e. day or night) based on astronomical twilight.

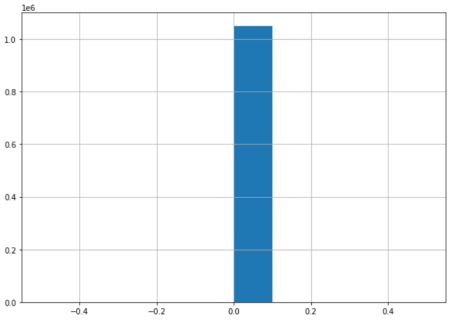
2. Data Exploration

2.1 Eliminating Unnecessary Features

- The feature ***Country*** contains only one entry — USA, which is quite apparent since we are dealing with the USA's dataset. Hence, deleting this feature.



- ***Turning_Loop*** also contains one value — False. Hence, deleting this feature.



- ***Source*** feature. It represents the API that reported the accident.Hence not relevant.

- In US officially the entire 50 states and the District of Columbia have **six** main ***time zones*** ,which is not relevant when the location/city of accident is present.
- **Traffic Message Channel (TMC)** is a technology for delivering traffic and travel information to motor vehicle drivers.plot for the number of accidents with respect to the *TMC* feature depicts that most numbers of accidents have a TMC of 201.Removing since this feature doesn't seem to be relevant.
- ***Distance(mi)*** feature tells the length(in miles) of the road extent affected by accident.
- ***Amenity*** feature indicates the presence of amenity in a nearby location,which is 98% false as per dataset.This is not much required feature.
- ***End_Lat*** and ***End_Lng*** features are having almost all values NULL so deleting these features

3. Data Preprocessing

3.1 Outlier Removal

A. Drop rows with negative time_duration

Negative time duration is illogical data so this has to be removed.

B. Fill outliers with median values

The values which are negative has to filled with median data ,which makes this feature more usable to generate the accurate prediction

For each numerical data we tried to plot the boxplot to find the outliers. We hardly removed outliers which is less than 1% of the whole data. For the Temperature attribute the total number of outliers found were **34207**. But since we had to retain the outliers, we had maintained them and so only we had discarded just **14** of them.ie, we had used the formula **low_lim=Q1-3*IQR** and **Up_lim=Q3+3*IQR** to remove the Temperature outliers.

For the Wind_Speed(mph) attribute the total number of outliers found were 17390. But since we had to retain the outliers, we had maintained them and so only we had discarded just **3778** of them.ie, we had used the formula **low_lim=Q1-7*IQR** and **Up_lim=Q3+7*IQR** to remove the Wind_Speed(mph) outliers.

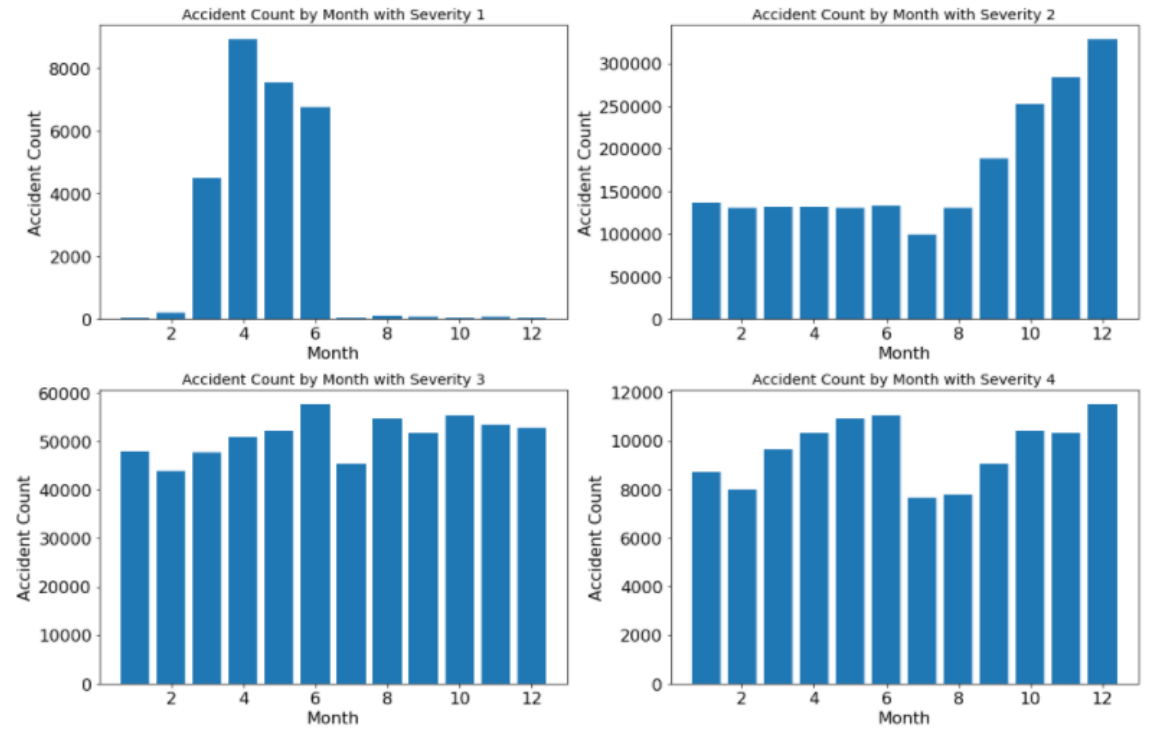
For Wind_Chill(F) attribute the total number of outliers found were **379286**. But since we had to retain the outliers, we had maintained them and so only we had discarded just **162** of them.ie, we had used the formula **low_lim=Q1-7*IQR** and **Up_lim=Q3+7*IQR** to remove the Wind_Chill(F) outliers.

For the Pressure attribute the total number of outliers found were 243445. But since the values of all Q1,Q2 and Q3 were 29.6, 29.92 ,30.07 and so IQR is 0.490, we have not removed any of the Pressure outliers.

For the Distance attribute the total number of outliers found were **4962**. But since the values of all Q1,Q2 and Q3 were 0.0 and so IQR is 0, we have not removed any of the Distance outliers.

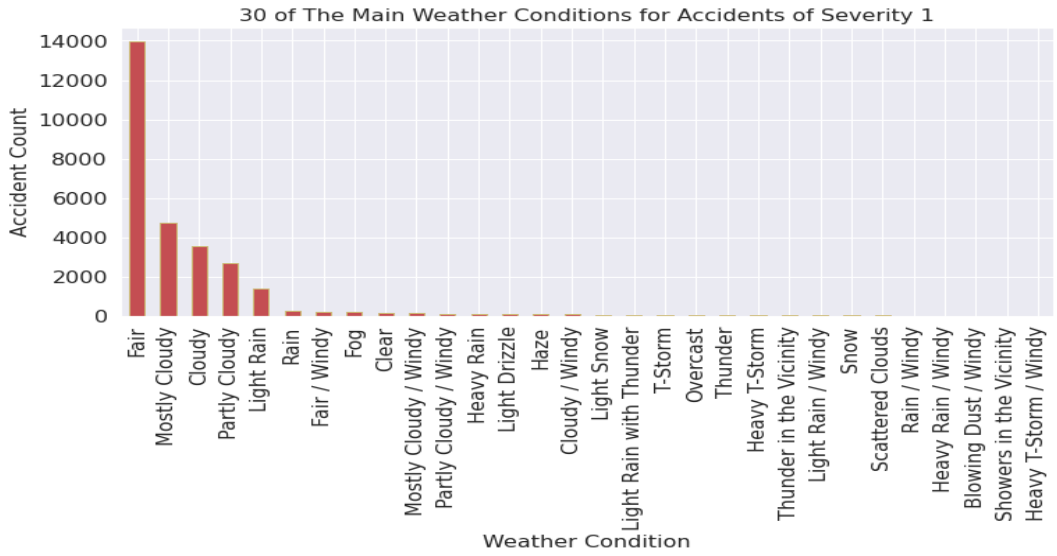
4. Exploratory Data Analysis

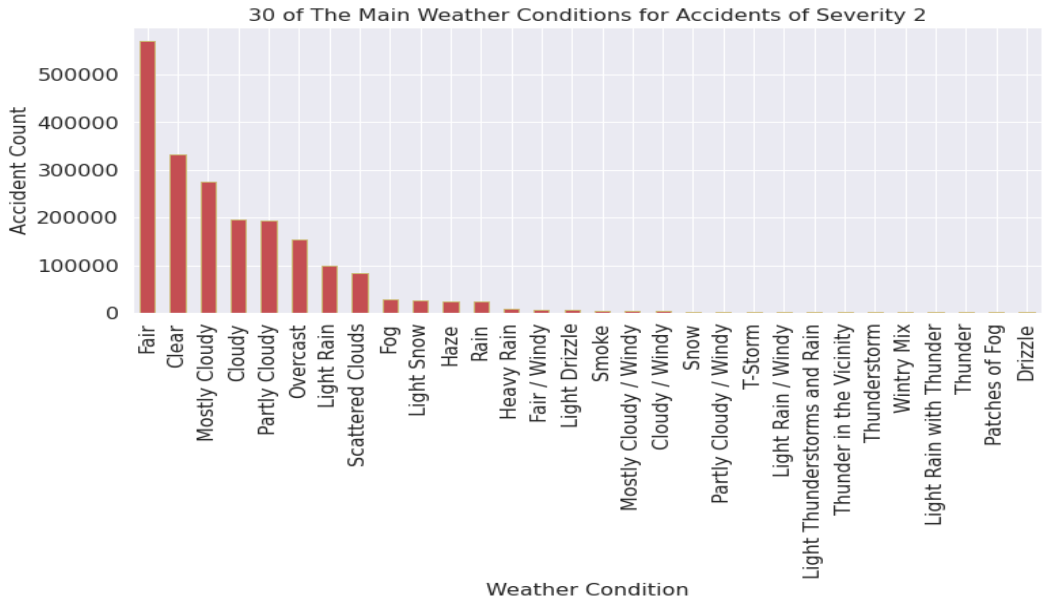
A. Plot accident count by month and severity

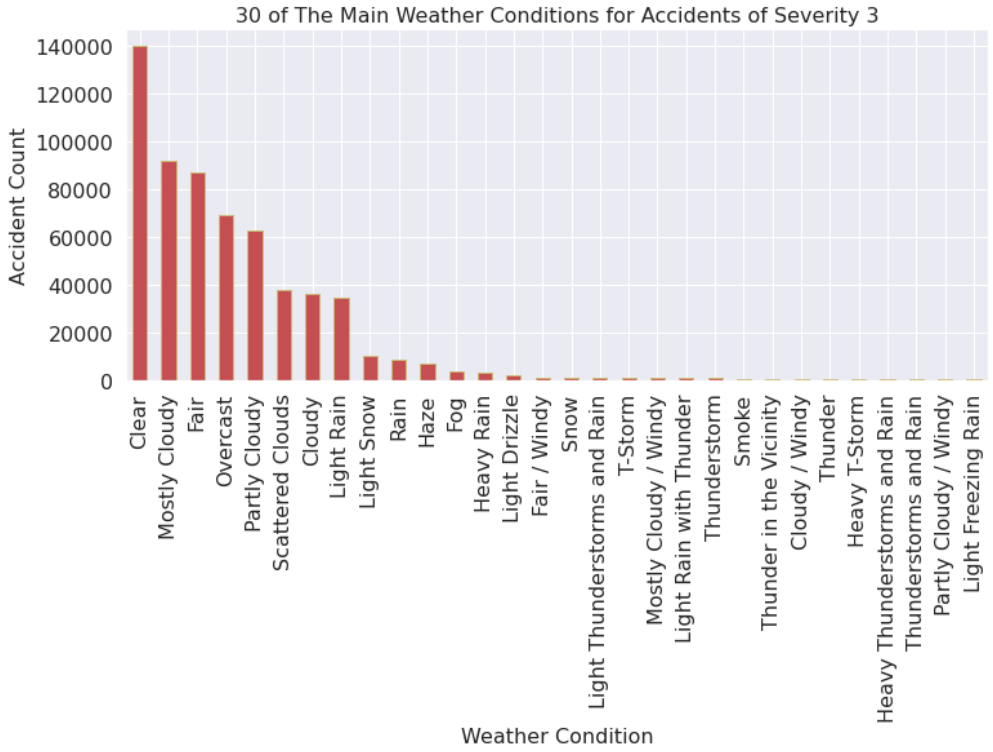


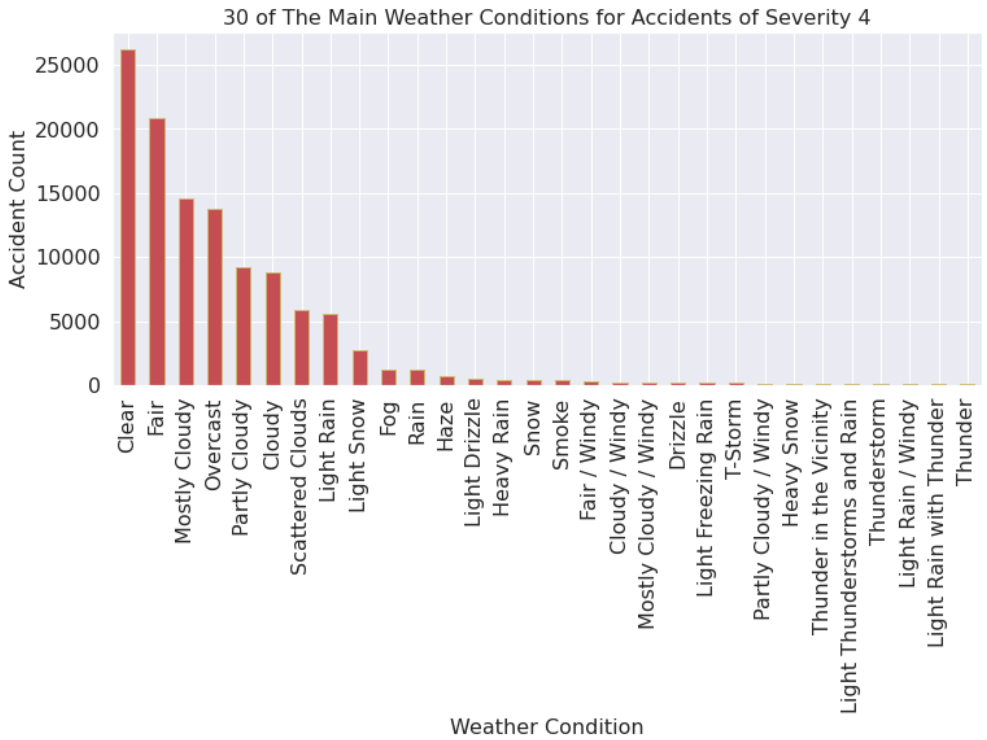
From the above bar plot,we can infer that October ,November and December have the most number of accidents. This may be because of the winter vacation and christmas time,because of this the traffic will be high and the amount of snowfall will also be high compared to other months

B. Plot bar plots according to various weather conditions.



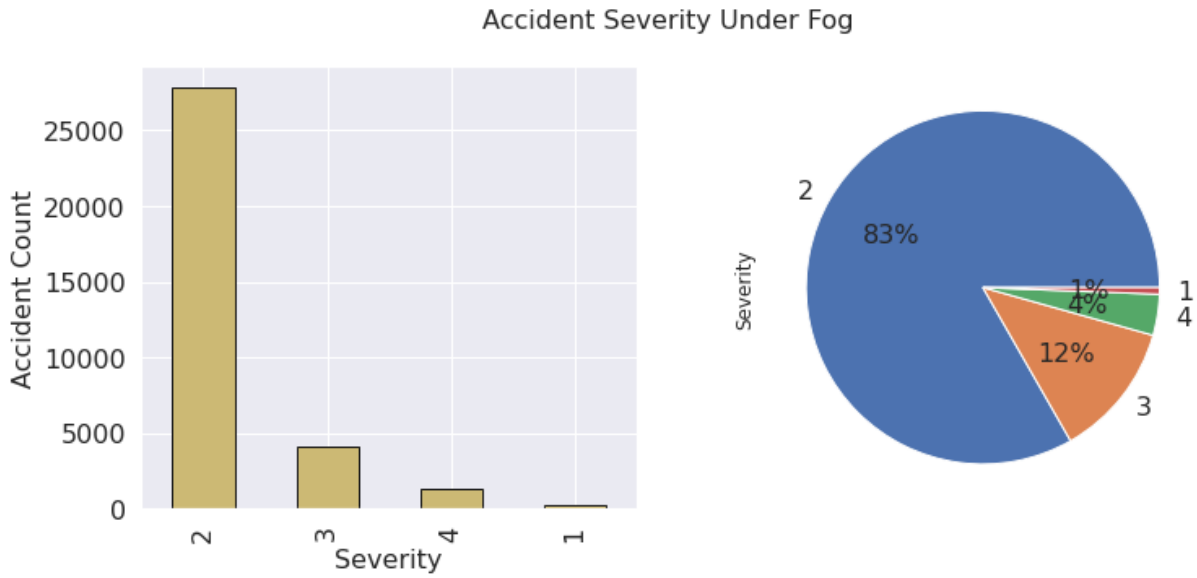


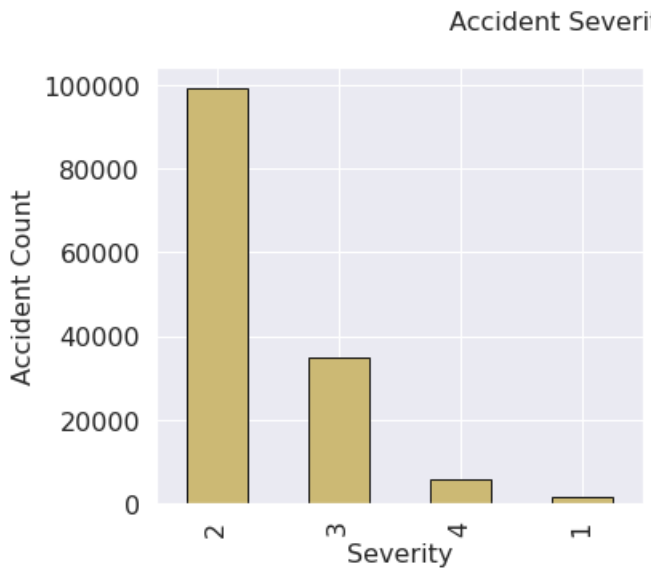

30 of The Main Weather Conditions for Accidents of Severity 3
Accident Count
140000
120000
100000
80000
60000
40000
20000
0
Clear
Mostly Cloudy
Fair
Overcast
Partly Cloudy
Scattered Clouds
Cloudy
Light Rain
Light Snow
Rain
Haze
Fog
Heavy Rain
Light Drizzle
Fair / Windy
Snow
Light Thunderstorms and Rain
T-Storm
Mostly Cloudy / Windy
Light Rain with Thunder
Thunderstorm
Smoke
Thunder in the Vicinity
Cloudy / Windy
Thunder
Heavy T-Storm
Heavy Thunderstorms and Rain
Thunderstorms and Rain
Partly Cloudy / Windy
Light Freezing Rain
Weather Condition

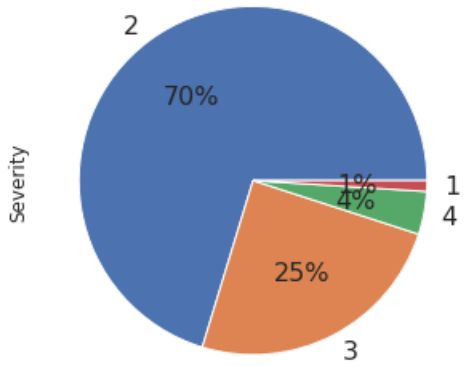

30 of The Main Weather Conditions for Accidents of Severity 4
Accident Count
25000
20000
15000
10000
5000
0
Clear
Fair
Mostly Cloudy
Overcast
Partly Cloudy
Cloudy
Scattered Clouds
Light Rain
Light Snow
Fog
Rain
Haze
Light Drizzle
Heavy Rain
Snow
Smoke
Fair / Windy
Cloudy / Windy
Mostly Cloudy / Windy
Drizzle
Light Freezing Rain
T-Storm
Partly Cloudy / Windy
Heavy Snow
Thunder in the Vicinity
Light Thunderstorms and Rain
Thunderstorm
Light Rain / Windy
Light Rain with Thunder
Thunder
Weather Condition

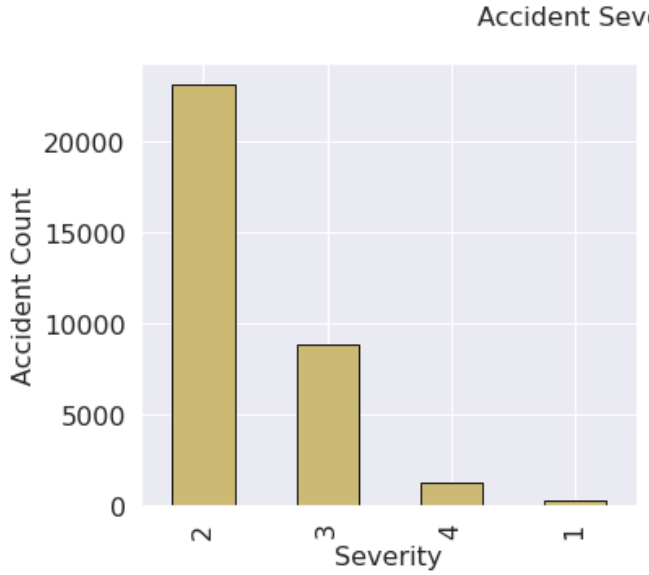
From the above plots, we can observe irrespective of severity, most of the accidents happened in clear weather conditions followed by overcast conditions. So during Overcast weather,there are chances of drizzle which causes tires to skid which may be the reason for accidents. But the accidents in clear weather conditions proves that other factors like roads, signals, and driver's experience are playing a major role here.

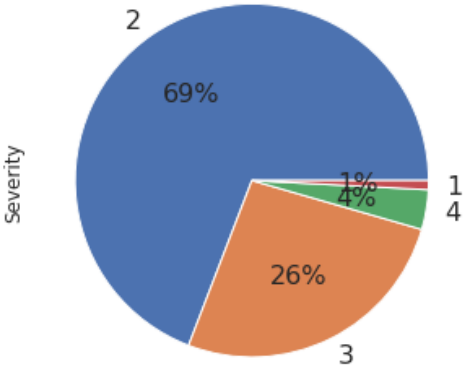
C. Severity by Fog, Light Rain, Rain, Heavy Rain and Snow

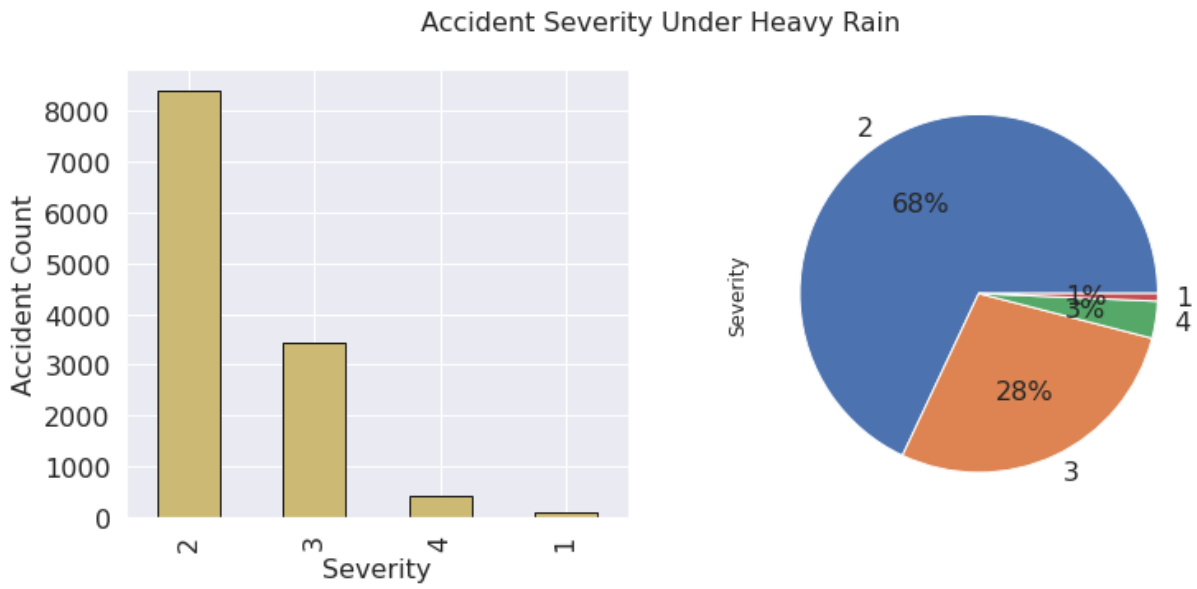


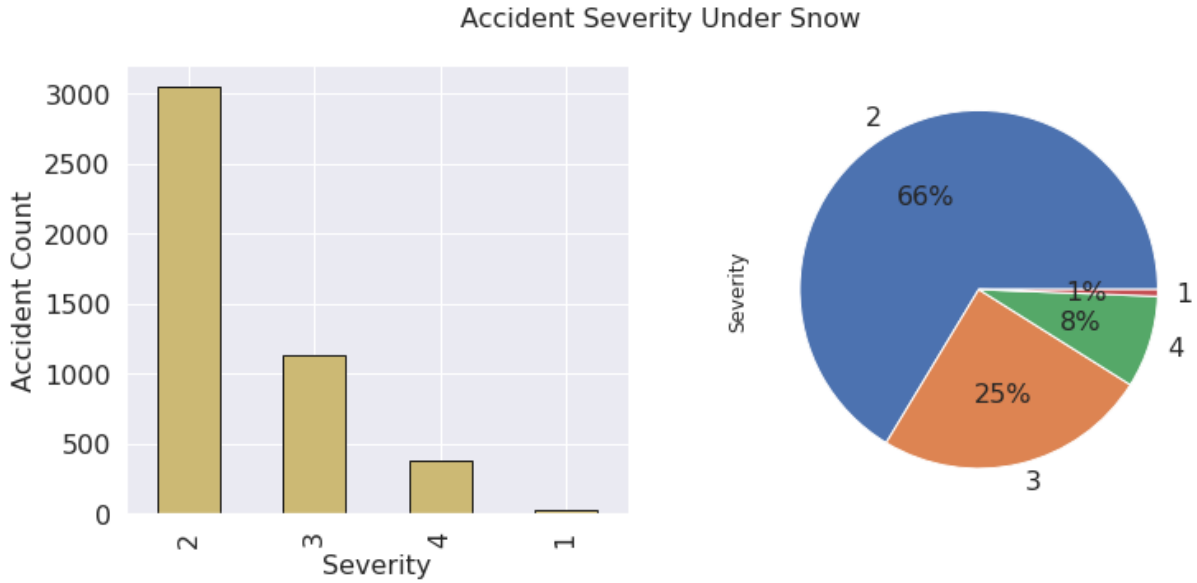

Accident Severity Under Light Rain
Accident Count
100000
80000
60000
40000
20000
0
2
3
4
1
Severity


Severity
2
70%
1%
1
4%
4
25%
3


Accident Severity Under Rain
Accident Count
20000
15000
10000
5000
0
2
3
4
1
Severity

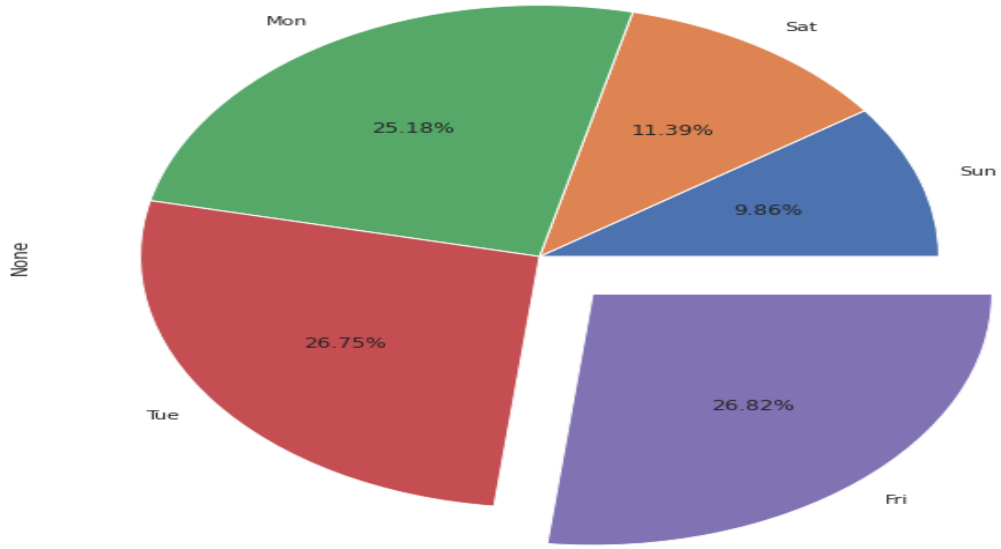

Severity
2
69%
1%
1
4%
4
26%
3





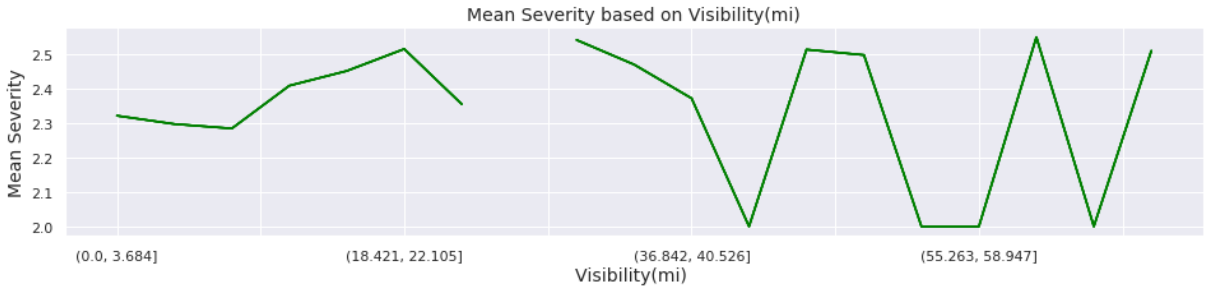
The proportion of level 3 and 4 accidents increases as weather changes from fog to light rain to rain to heavy rain to snow.So that means Weather Conditions certainly have an effect on accident severity.

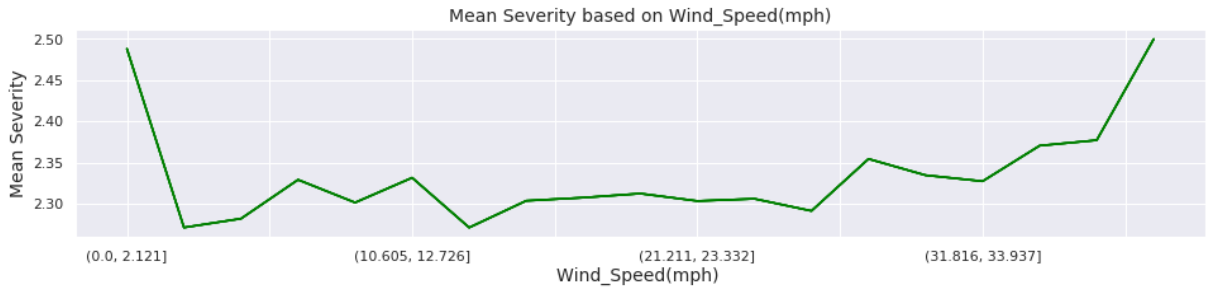
D. When the most number of accidents occurs in weekdays



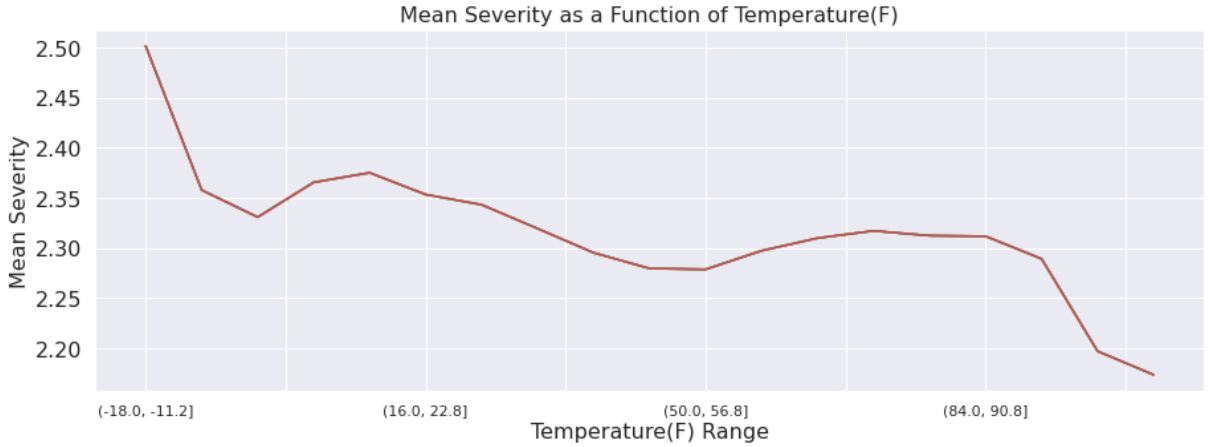
We can see that Monday, Thursday and Friday reported more accidents which are peak days of the week. Friday being weekend start day, reports the highest number of accidents may be due to rash driving after parties or rush to go vacations and may be because since Saturday and Sunday being off days,majority of students and working people will be travelling to their native places. However sunday and saturday reports lowest cases as they are rest days and majority of people will prefer to stay @ Home.

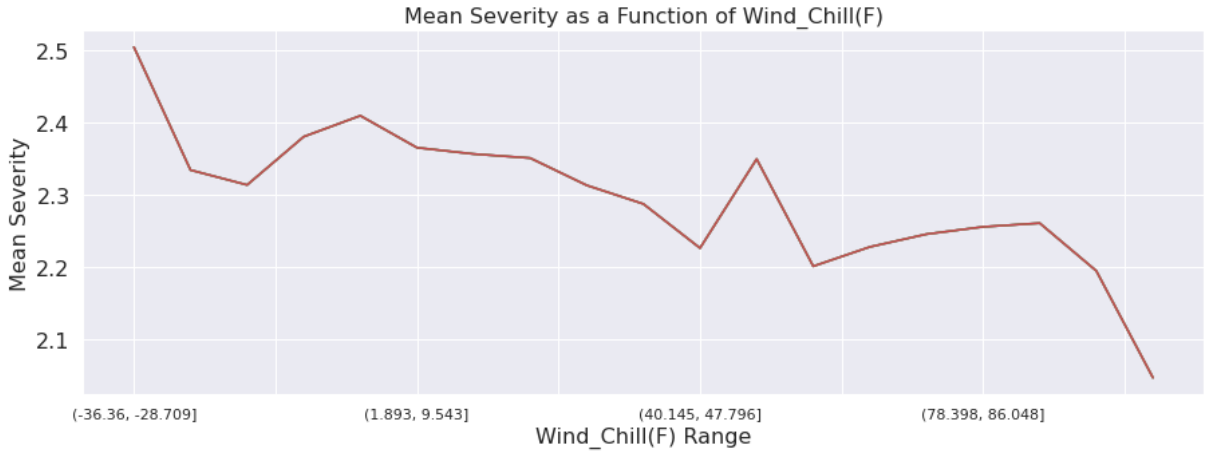
E. Checking Weather Factors

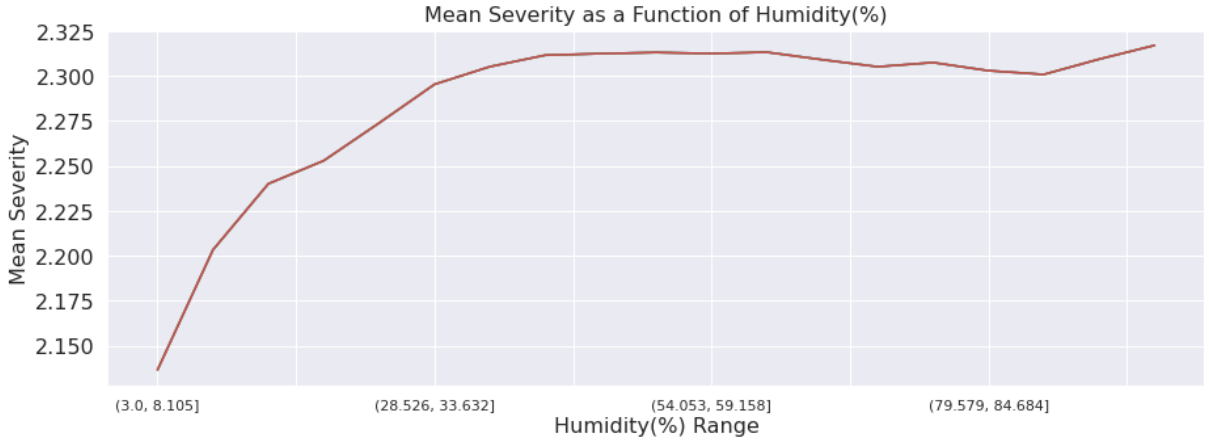


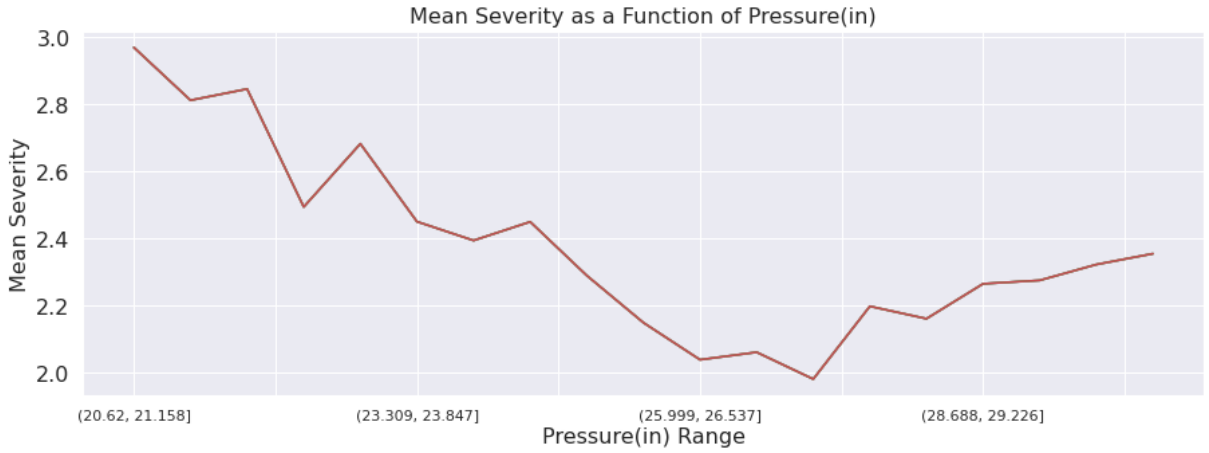


Here, increasing wind speed(mph) and diminished visibility causes accidents of severity 2, which implies even wind speed is also not the major factor for severity 4 accidents.



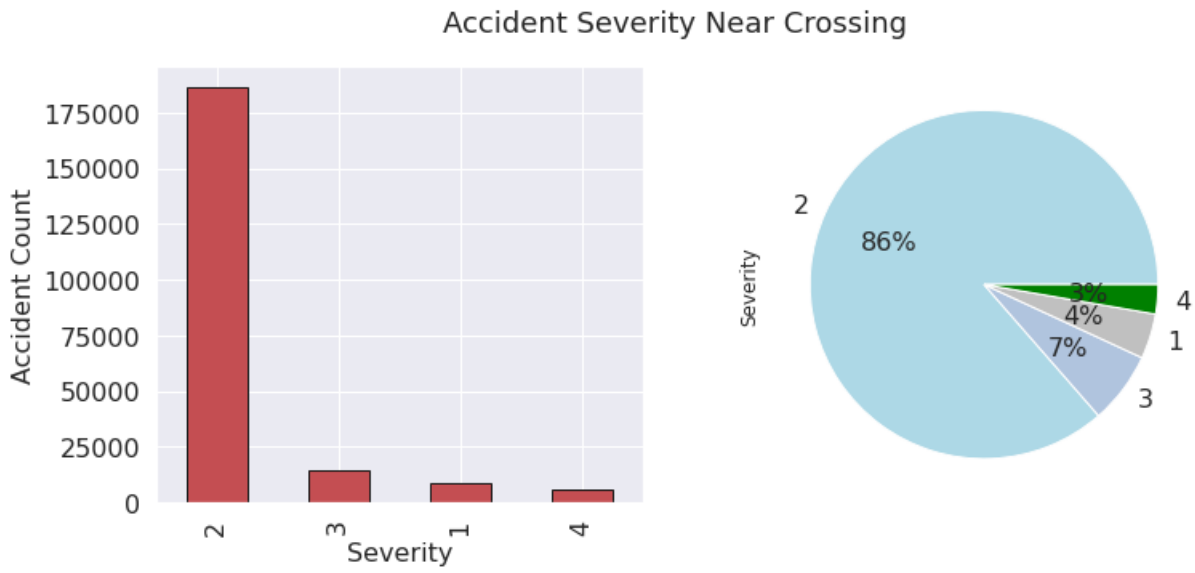

Mean Severity as a Function of Wind_Chill(F)
Mean Severity
2.5
2.4
2.3
2.2
2.1
(-36.36, -28.709]
(1.893, 9.543]
(40.145, 47.796]
(78.398, 86.048]
Wind_Chill(F) Range

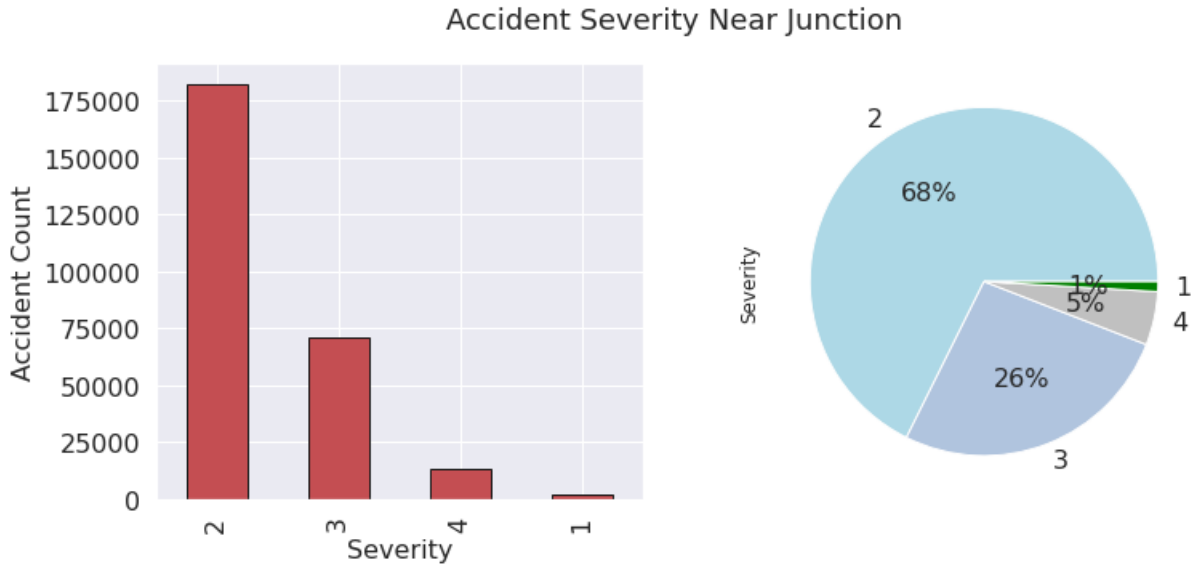

Mean Severity as a Function of Humidity(%)
Mean Severity
2.325
2.300
2.275
2.250
2.225
2.200
2.175
2.150
(3.0, 8.105]
(28.526, 33.632]
(54.053, 59.158]
(79.579, 84.684]
Humidity(%) Range

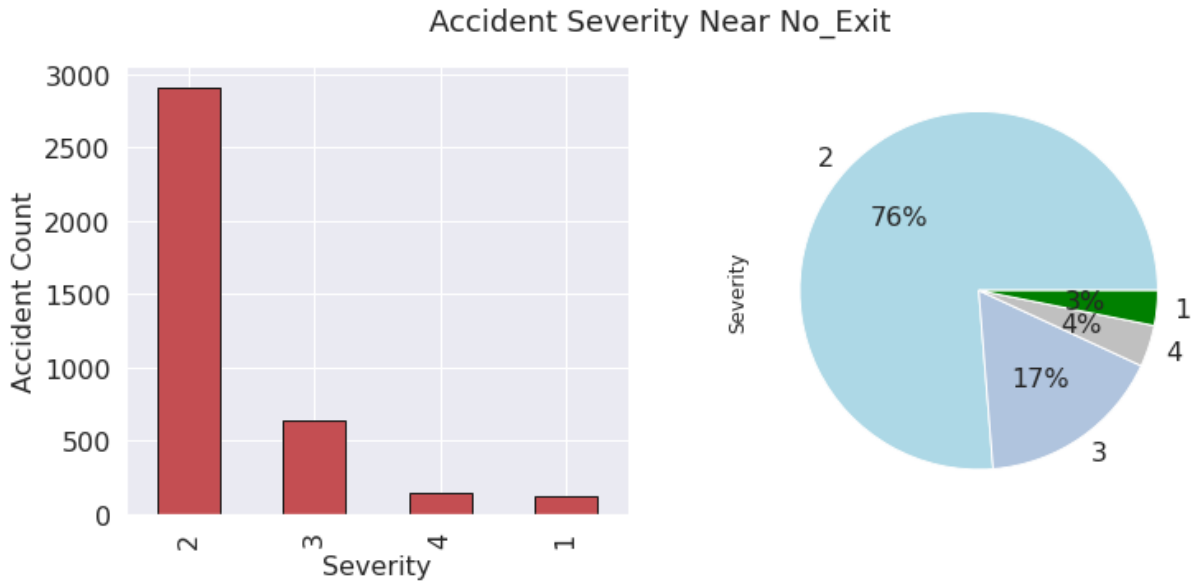

Mean Severity as a Function of Pressure(in)
Mean Severity
3.0
2.8
2.6
2.4
2.2
2.0
(20.62, 21.158]
(23.309, 23.847]
(25.999, 26.537]
(28.688, 29.226]
Pressure(in) Range

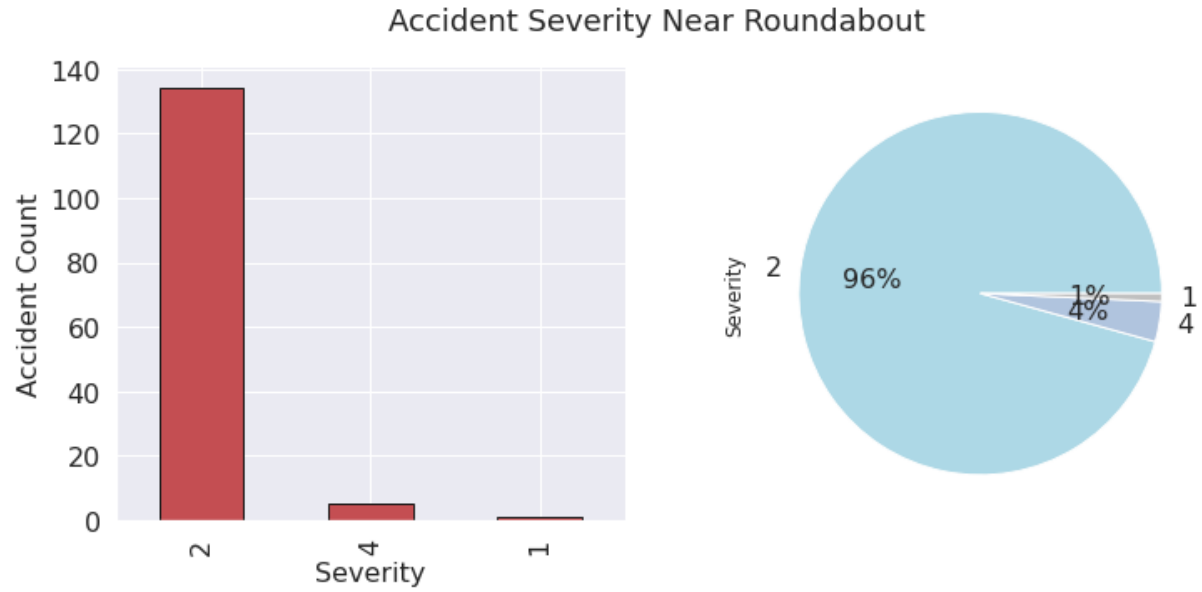
We can observe that rain and snow have a higher proportion of level 3 and 4 severity. These conditions include decreasing temperature, wind chill, and air pressure as well as increasing humidity. Severity also increases as a function of wind speed.

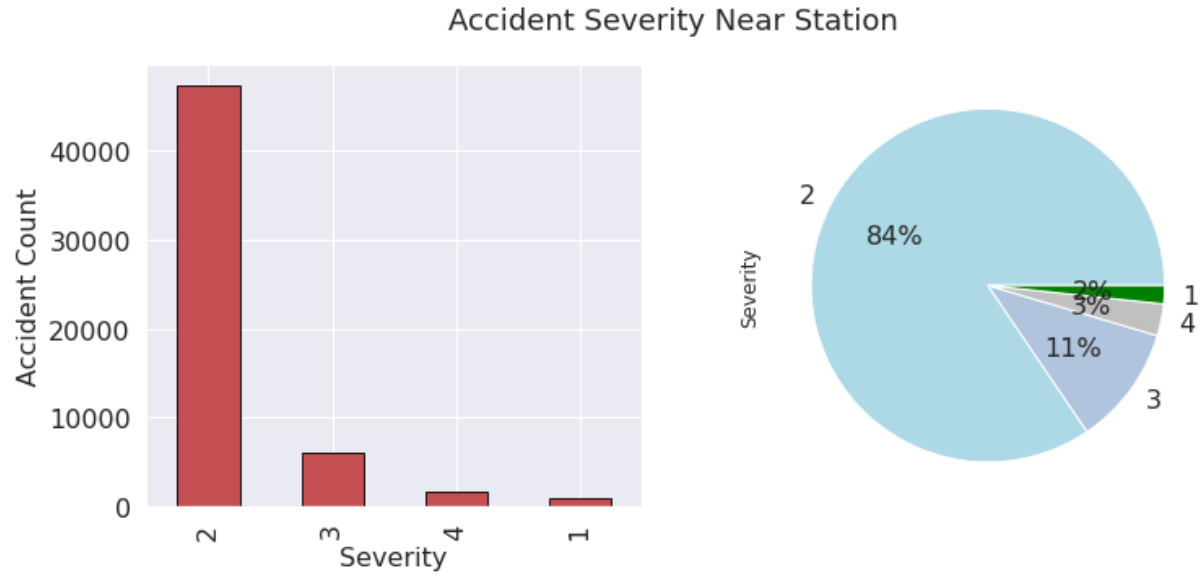
F. Graph Based on Point Of Interest Attributes

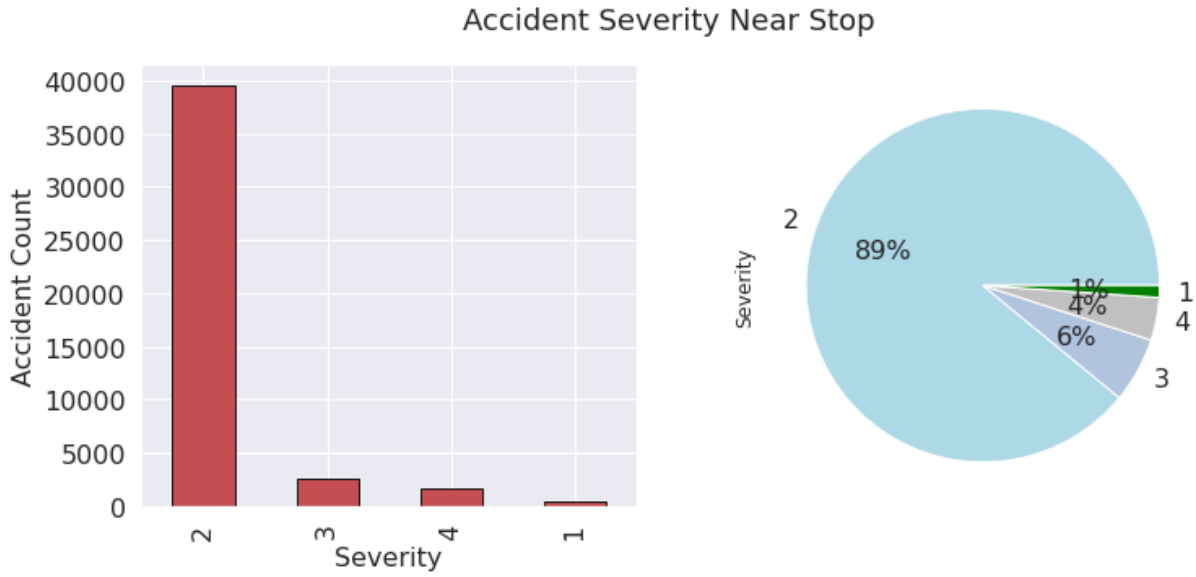


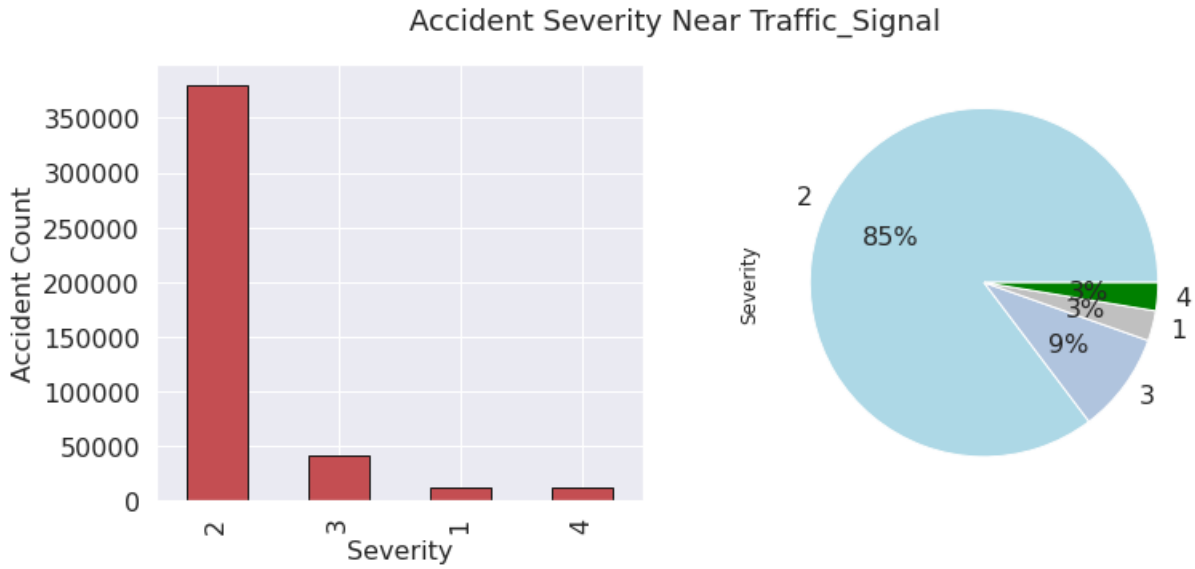

Accident Severity Near Junction
Accident Count
175000
150000
125000
100000
75000
50000
25000
0
2
3
4
1
Severity
Severity
2
68%
1%
5%
1
4
26%
3

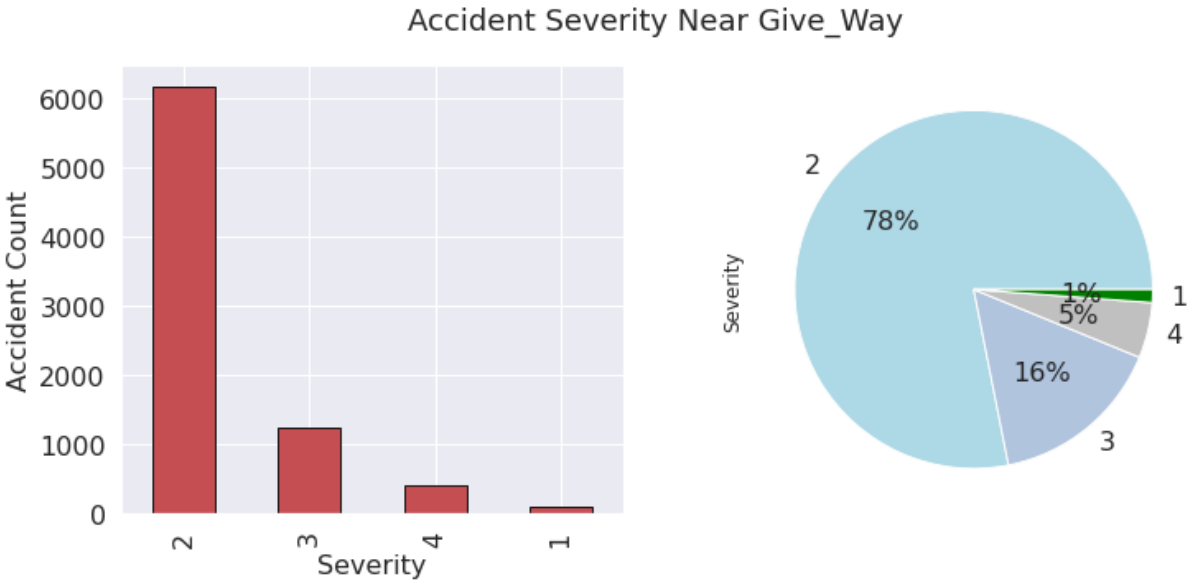

Accident Severity Near No_Exit
Accident Count
3000
2500
2000
1500
1000
500
0
2
3
4
1
Severity
Severity
2
76%
3%
4%
1
4
17%
3


Accident Severity Near Roundabout
Accident Count
140
120
100
80
60
40
20
0
2
4
1
Severity
Severity
2
96%
1%
4%
1
4


Accident Severity Near Station
Accident Count
40000
30000
20000
10000
0
2
3
4
1
Severity
Severity
2
84%
2%
3%
11%
1
4
3

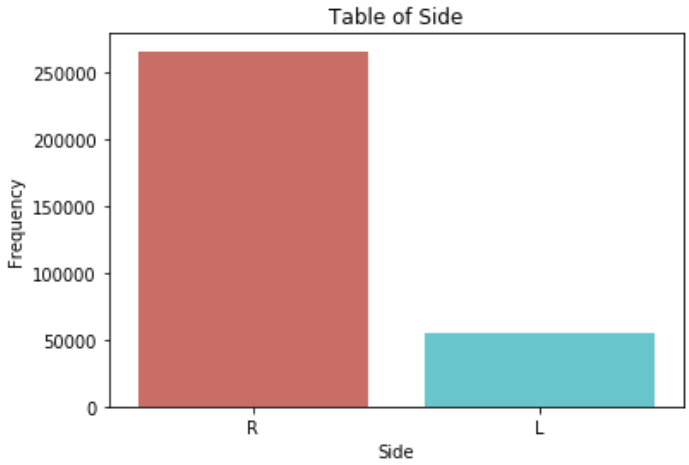

Accident Severity Near Stop
Accident Count
40000
35000
30000
25000
20000
15000
10000
5000
0
2
3
4
1
Severity
Severity
2
89%
1%
4%
6%
1
4
3


Accident Severity Near Traffic_Signal
Accident Count
350000
300000
250000
200000
150000
100000
50000
0
2
3
1
4
Severity
Severity
2
85%
3%
3%
9%
4
1
3



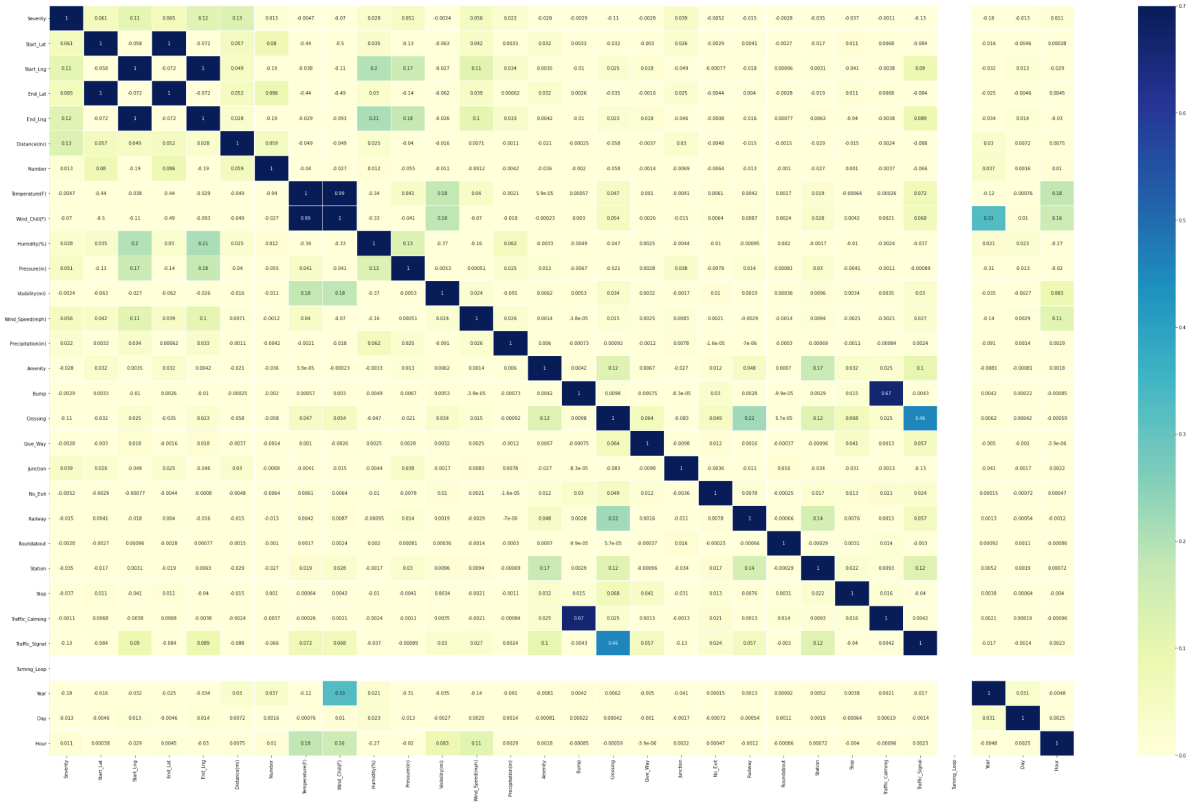
From the above diagram,we can see that the majority of severity 3 and severity 4 accidents are happening near junctions,near no exits,near traffic signals and near give ways. We can infer that absence of traffic signals or people violating rules are major reasons for these accidents.

G. Table Based on side



4.1 Correlation Matrix of the Numerical Features

Correlation is a **statistical** measure that indicates the extent to which two or more variables fluctuate together. A positive **correlation** indicates the extent to which those variables increase or decrease in parallel; a **negative correlation** indicates the extent to which one variable increases as the other decreases. This can be a very useful tool to quickly check which features are more correlated and which pair of features are not.



5. Splitting the data

One of the important steps is to split your dataset into train and test datasets .We have split the dataset in a 70–30 ratio, which is a common practice in data science.

6. Metrics

Functions to model and measure

6.1. Confusion Matrix

Each confusion matrix row shows the Actual/True labels in the test set and the columns show the predicted labels by classifier.

In the specific case of a binary classifier, we can interpret these numbers as the count of true positives (TP), false positives(FP), true negatives(TN), and false negatives(FN)

		Predicted	
		Negative	**Positive**
Actual	**False**	True Negative (TN)	False Positive (FP)
	True	False Negative (FN)	True Positive (TP)

6.2. Accuracy

Accuracy (ACC) is the number of correct predictions, divided by the total number of predictions. It is the fraction of predictions our model got right.

$$ACC = \frac{TP+TN}{TN+FN+TP+FP}$$

6.3. Precision

Precision (PR) is a measure of the accuracy provided that a class label has been predicted.

$$PR = \frac{TP}{TP+FP}$$

6.4. Recall

Recall(REC) is the true positive rate.

$$REC = \frac{TP}{TP+FN}$$

6.5. F1 Score

The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (which represents perfect precision and recall) and its worst at 0.

$$F1SCORE = 2 \times \frac{PR \times REC}{PR + REC}$$

7. Models

In order to carry out a good analysis, the metric that we are going to choose is crucial. Our main objective in this project is to find the clients most likely to subscribe to a term deposit, in order to carry out specific marketing campaigns efficiently. Since that, we are going to focus on the f1 score to evaluate the models. As it was defined as the harmonic mean of Precision and Recall, it penalizes the extreme values. So, it is a good measure to reduce the incorrectly classified cases, in order to achieve a balance between false negatives and false positives.

7.1. Linear Regression

Linear Regression is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope. It's used to predict values within a continuous range, (e.g. sales, price) rather than trying to classify them into categories (e.g. cat, dog). There are two main types:

Simple regression

Simple linear regression uses the traditional slope-intercept form, where m and b are the variables our algorithm will try to "learn" to produce the most accurate predictions. x represents our input data and y represents our prediction.

$$y = mx + b$$

Multivariable regression

A more complex, multi-variable linear equation might look like this, where w represents the coefficients, or weights, our model will try to learn.

$$f(x, y, z) = w_1 x + w_2 y + w_3 z$$

The variables x,y,z represent the attributes, or distinct pieces of information, we have about each observation. For sales predictions, these attributes might include a company's advertising spend on radio, TV, and newspapers.

$$Sales = w_1 Radio + w_2 TV + w_3 News$$

The following output is obtained.

```
Linear Regression:
R2 score: 0.08291355358977348
Mean Squared error: 0.28098892144979215
```

7.2. KNN

The k-nearest-neighbors algorithm is a classification algorithm that takes a bunch of labelled points and uses them to learn how to label other points. This algorithm classifies cases based on their similarity to other cases. In k-nearest neighbors, data points that are near each other are said to be "neighbors". K-nearest neighbors are based on this paradigm: "Similar cases with the same class labels are near each other". The following output is obtained.

```
KNN:
Classification Report
              precision    recall  f1-score   support

           0       0.53      0.43      0.48      5709
           1       0.80      0.92      0.86    415612
           2       0.56      0.36      0.44    122496
           3       0.51      0.15      0.23     23082

    accuracy                           0.76    566899
   macro avg       0.60      0.47      0.50    566899
weighted avg       0.73      0.76      0.74    566899

Accuracy is  0.7621392875979672
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1321:
  % (pos_label, average), UserWarning)
Precision is  0.7621392875979672
Recall is  0.8919332825569048
f1 score is  0.8409502708388409
```

7.3. Logistic Regression

Logistic Regression is a variation of Linear Regression, useful when the observed dependent variable, **y**, is categorical. It produces a formula that predicts the probability of the class label as a function of the independent variables.

Logistic regression fits a special s-shaped curve by taking the linear regression and transforming the numeric estimate into a probability with the following function, which is called sigmoid function σ:

$$h_\theta(x) = \sigma(\theta^T X) = \frac{e^{(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots)}}{1 + e^{(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots)}}$$

In this equation, $\theta^T X$ is the regression result (the sum of the variables weighted by the coefficients), exp is the exponential function and $\sigma(\theta^T X)$ is the sigmoid or logistic function , also called logistic curve. It is a common "S" shape (sigmoid curve).

So, briefly, Logistic Regression passes the input through the logistic/sigmoid but then treats the result as a probability. The objective of Logistic Regression algorithm, is to find the best parameters θ, for $h\theta(x) = \sigma(\theta^T X)$, in such a way that the model best predicts the class of each case. The following output is obtained.

```
Logistic Regression:
Classification Report
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1272:
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      5709
           1       0.73      1.00      0.85    415612
           2       1.00      0.00      0.00    122496
           3       0.00      0.00      0.00     23082

    accuracy                           0.73    566899
   macro avg       0.43      0.25      0.21    566899
weighted avg       0.75      0.73      0.62    566899

Accuracy is  0.7331341208927868
Precision is  0.7331341208927868
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1321:
  % (pos_label, average), UserWarning)
Recall is  0.9999993821255174
f1 score is  0.8460215455979804
```

7.4. DECISION TREE

A decision tree is a decision support tool that is built using recursive partitioning to classify the data. The algorithm chooses the most predictive feature to split the data on. The objective is to determine “which attribute is the best, or more predictive, to split data based on the feature. The functions to measure the quality of a split are: “gini” for the Gini impurity and “entropy” for the information gain. The following output is obtained.

```
Decision Tree:
Classification Report
              precision    recall  f1-score   support

           0       0.57      0.61      0.59      5709
           1       0.88      0.87      0.87    415612
           2       0.63      0.63      0.63    122496
           3       0.39      0.42      0.41     23082

    accuracy                           0.80    566899
   macro avg       0.62      0.63      0.62    566899
weighted avg       0.80      0.80      0.80    566899

Accuracy is  0.7979622472433361
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1321:
  % (pos_label, average), UserWarning)
Precision is  0.7979622472433361
Recall is  0.8498508113815587
f1 score is  0.8521077404511604
array([[  3469,   1654,    454,    132],
       [  1956, 361790,  41331,  10535],
       [   477,  40198,  77393,   4428],
       [   133,   9263,   3974,   9712]])
```

7.5. RANDOM FOREST

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees. Cross-validation is not necessary when using random forest, because multiple bagging in the process of training random forest prevents overfitting. The following output is obtained.

```
Random Forest Classifier:
Classification Report
              precision    recall  f1-score   support

           0       0.75      0.57      0.65      5709
           1       0.85      0.92      0.89    415612
           2       0.68      0.56      0.61    122496
           3       0.73      0.33      0.45     23082

    accuracy                           0.82    566899
   macro avg       0.75      0.59      0.65    566899
weighted avg       0.81      0.82      0.81    566899

Accuracy is  0.8169427005515973
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1321:
  % (pos_label, average), UserWarning)
Precision is  0.8169427005515973
Recall is  0.9008051597715667
f1 score is  0.8706282989976021
```

7.6. GRADIENT BOOSTING

Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Gradient Boosting for classification.

GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage classes regression trees fit on the negative gradient of the binomial or multinomial deviance loss function. Binary classification is a special case where only a single regression tree is induced.The following output is obtained.

```
Gradient Boosting Classifier:
Classification Report
              precision    recall  f1-score   support

           0       0.77      0.25      0.38      5709
           1       0.76      0.97      0.85    415612
           2       0.56      0.15      0.24    122496
           3       0.50      0.00      0.01     23082

    accuracy                           0.75    566899
   macro avg       0.65      0.34      0.37    566899
weighted avg       0.70      0.75      0.68    566899

Accuracy is  0.7460976293837174
Precision is  0.7460976293837174
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1321:
  % (pos_label, average), UserWarning)
Recall is  0.955574242756999
f1 score is  0.844679901452515
```

7.7. XGBOOST

XGBoost is a specific implementation of the Gradient Boosting method which uses more accurate approximations to find the best tree model. XGBoost stands for “Extreme Gradient Boosting”. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements Machine Learning algorithms under the Gradient Boosting framework. It provides a parallel tree boosting to solve many data science problems in a fast and accurate way. The following output is obtained.

```
Classification Report
              precision    recall  f1-score   support

           0       0.84      0.11      0.20      5665
           1       0.75      0.99      0.85    415682
           2       0.58      0.08      0.14    122589
           3       0.55      0.00      0.00     22996

    accuracy                           0.74    566932
   macro avg       0.68      0.29      0.30    566932
weighted avg       0.70      0.74      0.65    566932

Accuracy is  0.7410624201844312
Precision is  0.7410624201844312
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_
ive') is ignored when average != 'binary' (got 'micro').
  % (pos_label, average), UserWarning)
Recall is  0.9785175157388716
f1 score is  0.8463947530358673
```

8. Result

Sr No	Algorithm	Accuracy
1	Logistic Regression	0.73
2	Decision Tree	0.79
3	RANDOM FOREST	0.816
4	KNN	0.76
5	Gradient Boosting	0.76
6	Xtreme GB	0.74

9. Coding Details

9.1 Python Flask

Flask is called a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself.

9.2 Web Hosting using Python Flask

We have a web application designed to show the project structure for a machine learning model deployed using flask.This application acts as an interface for a user to submit new queries.

http://carolinemary.pythonanywhere.com/

9.3 Other Tools/Language Used for programming

Python

Python is an easy to understand and commonly used programming language. Python is an object-oriented language used for data analysis.

Benefits of Python:

- Data Analysis
- Development of Website
- Development of Application

Why Python?

- Python is available on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.).
- Python is more straightforward as the English language, i.e., simple coding

- Python has a syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be rapid
- Python can be treated procedurally, an object-orientated way, or a practical way.

In this project, python has been used for data cleaning and data manipulation.Jupyter notebook has been used for python. All analysis is done in Python 3.0.

Spyder

Spyder is an open source cross-platform integrated development environment (IDE) for scientific programming **in the Python** language.

- An editor with syntax highlighting, introspection, code completion.
- Support for multiple IPython consoles.
- The ability to explore and edit variables from a GUI.
- A Help pane able to retrieve and render rich text documentation on functions, classes and methods automatically or on-demand.

Visual Studio

Visual Studio is an integrated development environment that is used to develop computer programs for Windows. Visual studio can also be used for developing web sites, web applications, and web services. It uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

10. Conclusion

This project describes US-Accidents, a unique, publicly available motor vehicle accident dataset, and its process of creation – that includes several important steps such as real-time traffic data collection, data integration, and multistage data augmentations using map-matching, weather, period-of-day, and points-of-interest data. To the best of our knowledge, US Accidents is the first countrywide dataset of this scale, containing about 2.25 million traffic accident records collected for the contiguous United States over three years. From this dataset, we were able to derive a variety of insights with respect to the location, time, weather, and points-of-interest of an accident. We believe that US Accidents provide a context for future research on traffic accident analysis and prediction. In terms of our own future work, we plan to employ this dataset to perform real-time traffic accident prediction

All in all, through this project, we have gained deeper insights about road accidents in the United States based on data from 2016 to 2019. Through exploratory data analysis on the data set, many intriguing conclusions were drawn. For example, it is evident that more road accidents occur during morning commutes (7–9 am) and on weekdays compared to weekends. Additionally, road accidents in the United States tend to be moderate to severe in terms of the delays they cause. Moreover, from 2016–2019, California had the most road accidents of all US states while Houston led all US cities. Finally, of all roadway infrastructure features, the highest proportion of high severity accidents occur at junctions between multiple roads.

In terms of building a ML model, we took the approach of formalizing a binary classification problem where accident severity values of 1 and 2 were considered a low severity and accident severity values of 3 and 4 were considered a high severity. We experienced the most success with a Random Forest Model in which we considered all types of features including location data (start latitude and longitude of the accident), time data (hour in the day and day of the week for the accident), POI information, road infrastructure details, and weather data. We were able to further improve the performance of the model by adding features based on natural language processing of

keywords in the accident descriptions in the data set as well as keywords in the descriptions of Traffic Message Channel codes.

11. Future Work

In terms of my future work, we will be making this platform available to state governments and the public so it can be easily accessible. Different versions of this platform can be created, which can address specific problems of state government and people. This application needs to be launched on a website, and also live data will need to be fed to get a more up to date analysis. Creating a machine learning model can help the public predict an accident location based on the source and destination location along with the date and time of travel. This type of prediction model can help reduce the number of accidents happening in the US. The prediction model can incorporate several neural network-based components that use a variety of data attributes, such as traffic events, weather data, points-of-interest, and time information.

12. References

- Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, and Rajiv Ramnath.arXiv preprint arXiv:1906.05409 (2019).
- Najjar, A., Kaneko, S., & Miyanaga, Y. (2017). Combining Satellite Imagery and Open Data to Map Road Safety. Thirty-First AAAI Conference on Artificial Intelligence.
- Yuan Z., Zhou X., Yang T., Tamerius J, and Mantilla R.(2017). Predicting traffic accidents through heterogeneous urban data: A case study. In Proceedings of the 6th International Workshop on Urban Computing , Halifax, NS, Canada, Vol. 14.