Study of Severity of US Car Accidents Using Machine Learning

Shambhu KC

September 21, 2020

Existing Problem

- Road accidents kills 1.35 million/ year and injures 50 million/year
- Causes impact on the traffic
- Numbers can be minimized in many cases

Project Goal

- Build a model by Machine Learning
- Suggest any recommendation to minimize the impact of accident

Data Collection

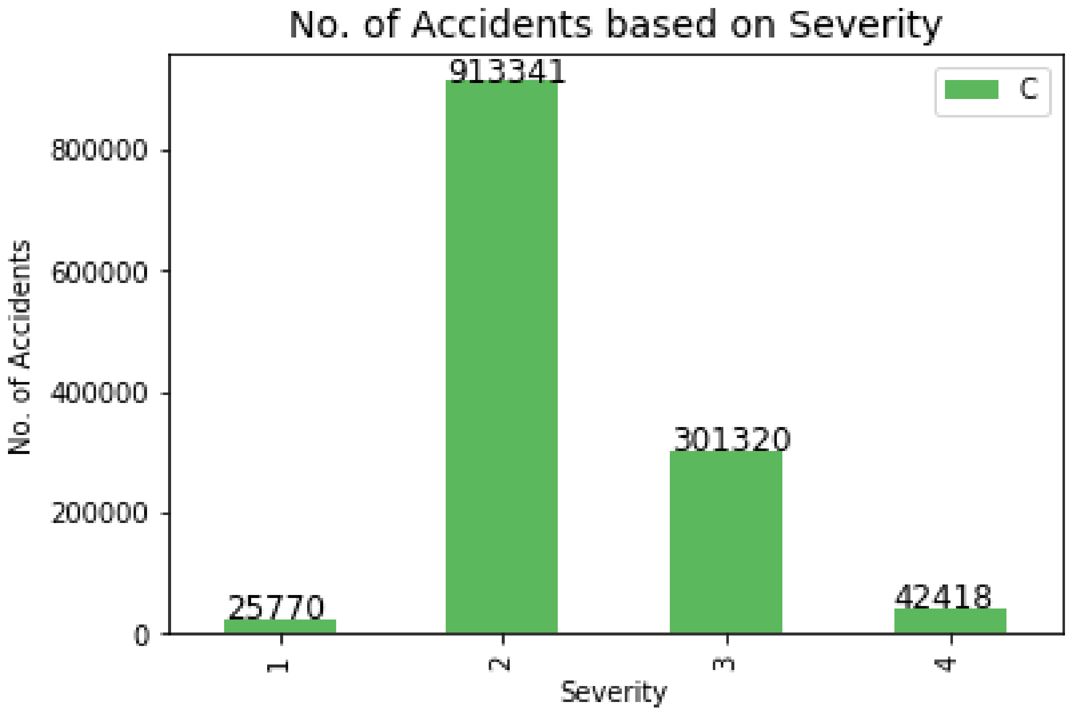
- ❖ Historical data of accidents from Kaggle
- ❖ Record from Feb 2016 to June 2020 covering 49 states of US
- ❖ 3.5 million accident records
- ❖ 46 columns covering many features; location, weather, traffic signals

Data Cleaning

- ❖ Target variable is Severity in terms of impact on the traffic
- ❖ Features related to time, weather and traffic objects
- ❖ Removing columns with NaN values
- ❖ Removing unnecessary columns

Data Visualization and Analysis

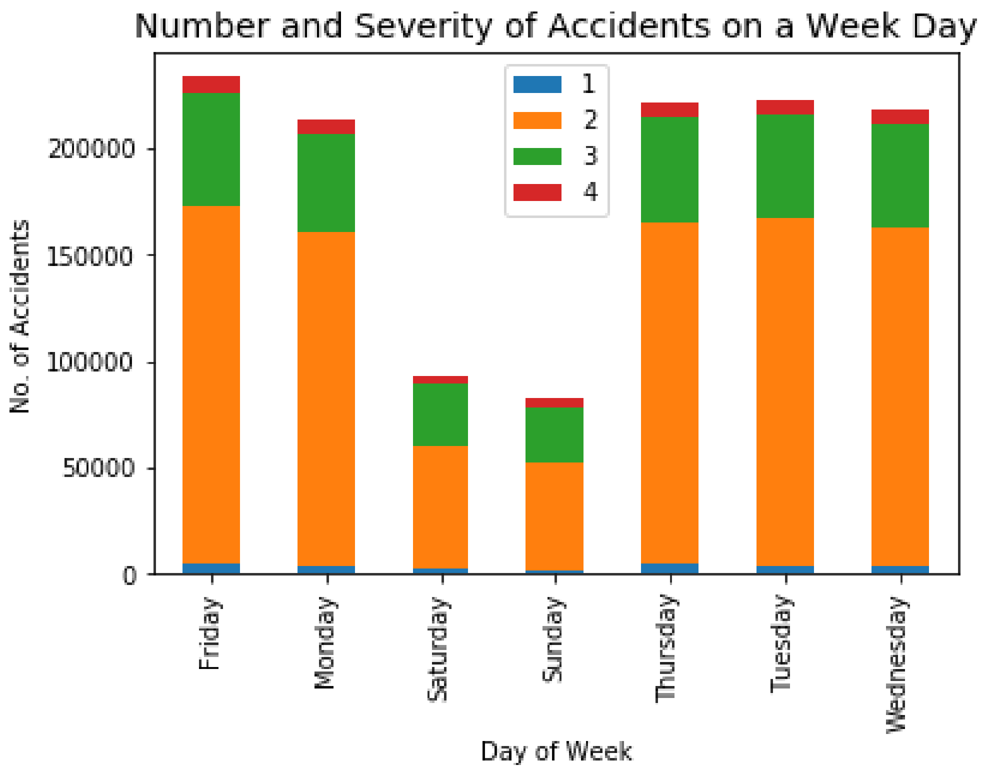
How many accidents of each category?

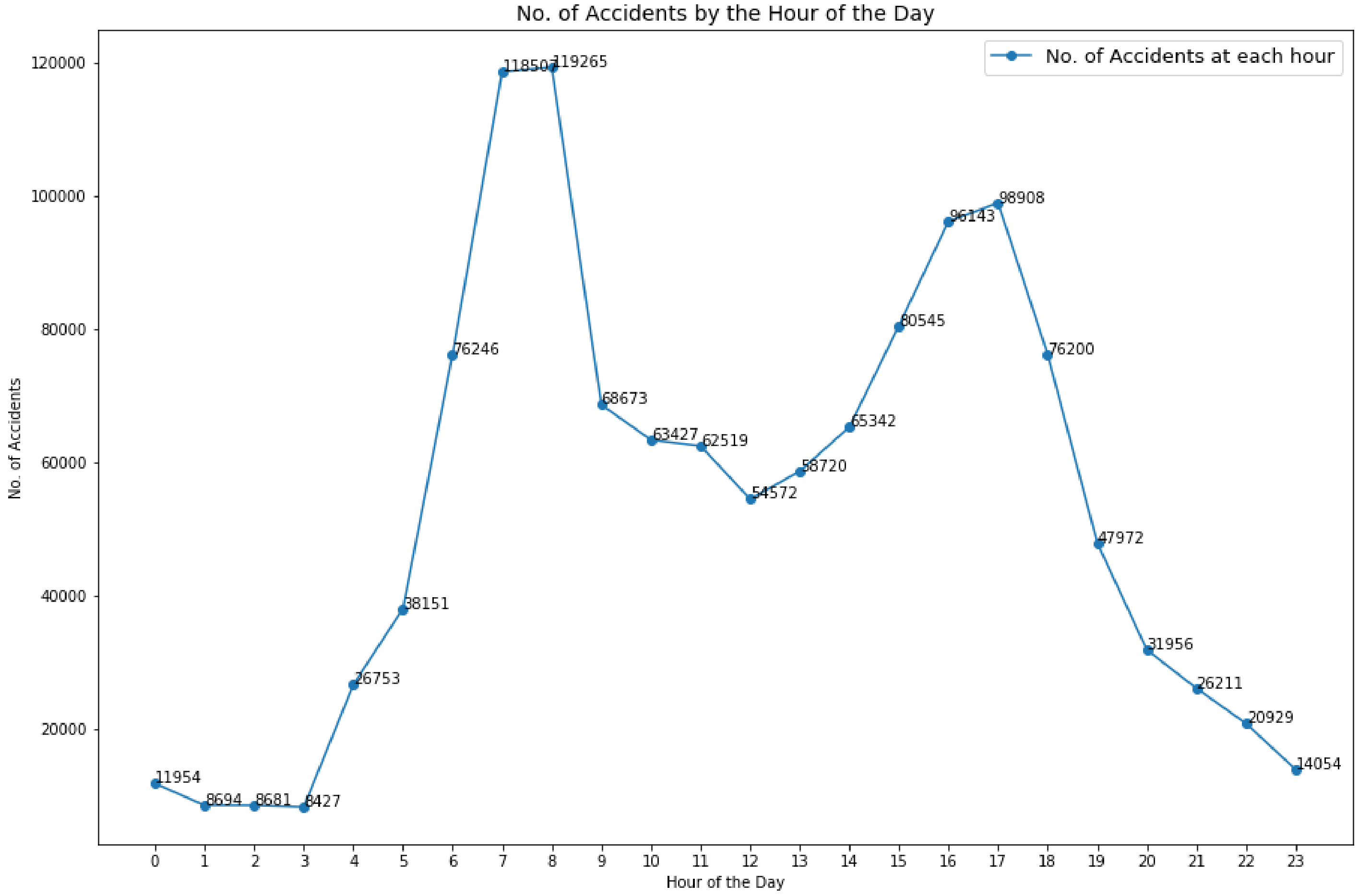


Majority of accidents were of severity level 2.

Data Visualization and Analysis

How a particular day and time affects?

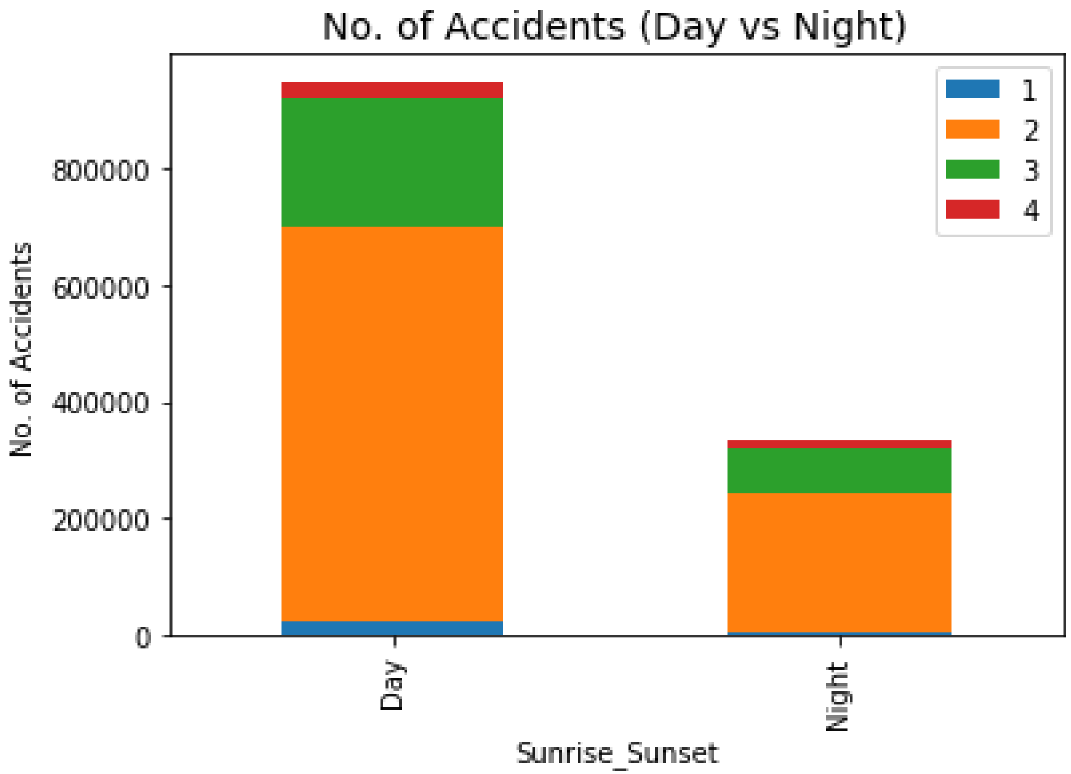




More accidents occurred during rush hours.

Data Visualization and Analysis

What is the affect of day and Night?

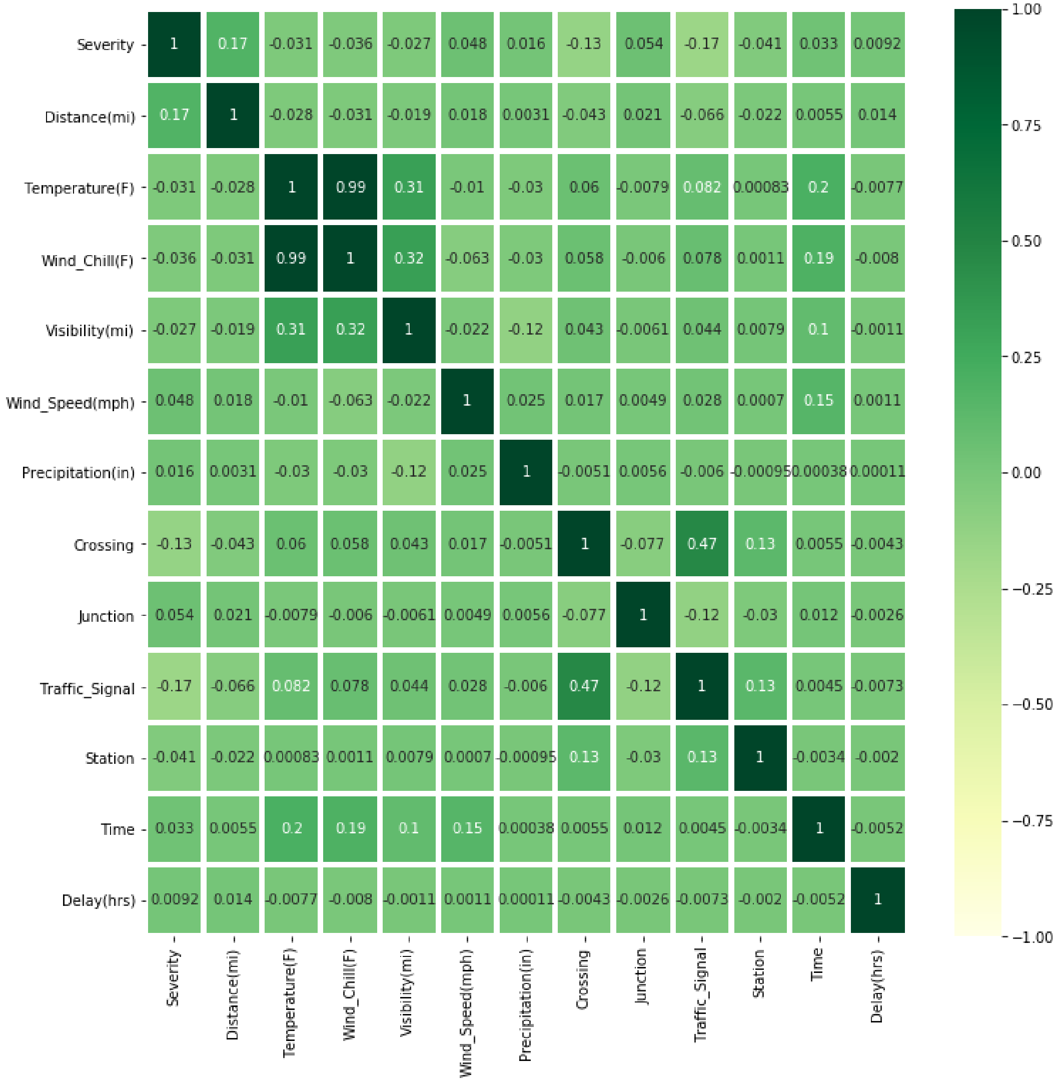


More accidents occurred during day

Data Visualization and Analysis

Correlation between different features and Severity

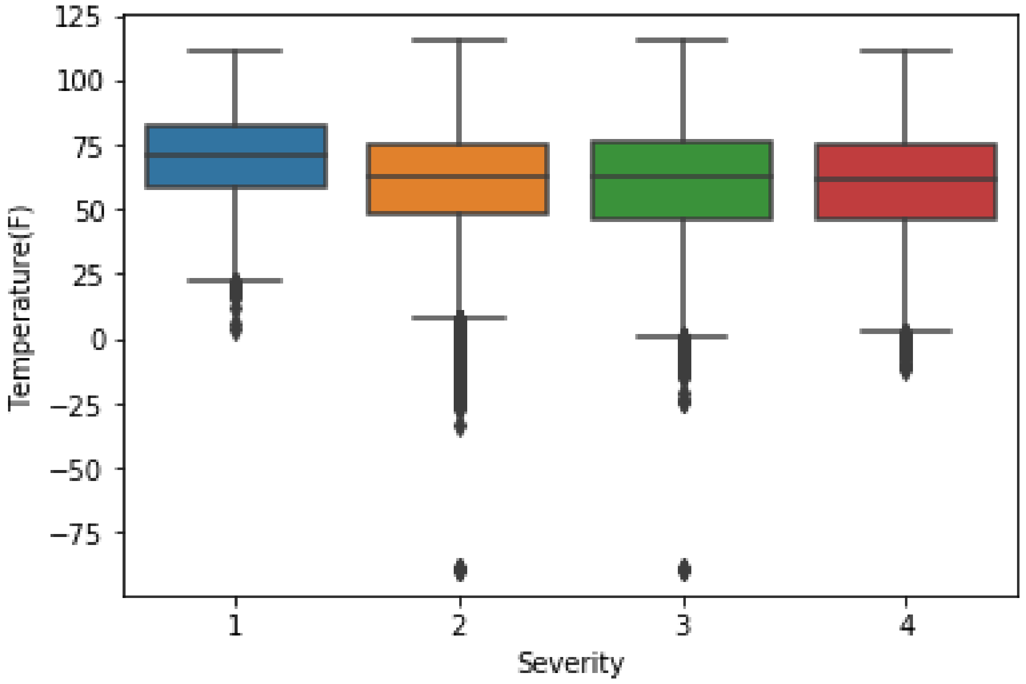
There is some degree of correlation.

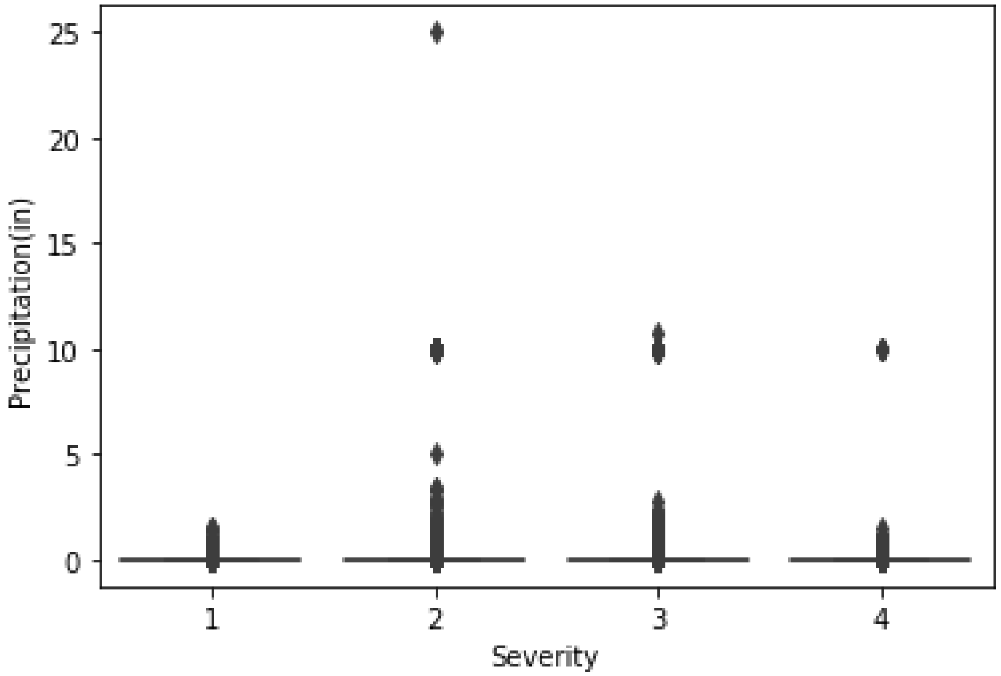


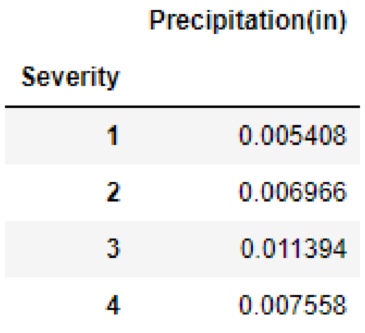
	Severity	Distance(mi)	Temperature(F)	Wind_Chill(F)	Visibility(mi)	Wind_Speed(mph)	Precipitation(in)	Crossing	Junction	Traffic_Signal	Station	Time	Delay(hrs)
Severity	1	0.17	-0.031	-0.036	-0.027	0.048	0.016	-0.13	0.054	-0.17	-0.041	0.033	0.0092
Distance(mi)	0.17	1	-0.028	-0.031	-0.019	0.018	0.0031	-0.043	0.021	-0.066	-0.022	0.0055	0.014
Temperature(F)	-0.031	-0.028	1	0.99	0.31	-0.01	-0.03	0.06	-0.0079	0.082	0.00083	0.2	-0.0077
Wind_Chill(F)	-0.036	-0.031	0.99	1	0.32	-0.063	-0.03	0.058	-0.006	0.078	0.0011	0.19	-0.008
Visibility(mi)	-0.027	-0.019	0.31	0.32	1	-0.022	-0.12	0.043	-0.0061	0.044	0.0079	0.1	-0.0011
Wind_Speed(mph)	0.048	0.018	-0.01	-0.063	-0.022	1	0.025	0.017	0.0049	0.028	0.0007	0.15	0.0011
Precipitation(in)	0.016	0.0031	-0.03	-0.03	-0.12	0.025	1	-0.0051	0.0056	-0.006	-0.00095	0.00038	0.00011
Crossing	-0.13	-0.043	0.06	0.058	0.043	0.017	-0.0051	1	-0.077	0.47	0.13	0.0055	-0.0043
Junction	0.054	0.021	-0.0079	-0.006	-0.0061	0.0049	0.0056	-0.077	1	-0.12	-0.03	0.012	-0.0026
Traffic_Signal	-0.17	-0.066	0.082	0.078	0.044	0.028	-0.006	0.47	-0.12	1	0.13	0.0045	-0.0073
Station	-0.041	-0.022	0.00083	0.0011	0.0079	0.0007	-0.00095	0.13	-0.03	0.13	1	-0.0034	-0.002
Time	0.033	0.0055	0.2	0.19	0.1	0.15	0.00038	0.0055	0.012	0.0045	-0.0034	1	-0.0052
Delay(hrs)	0.0092	0.014	-0.0077	-0.008	-0.0011	0.0011	0.00011	-0.0043	-0.0026	-0.0073	-0.002	-0.0052	1

Data Visualization and Analysis

How does weather condition impact?



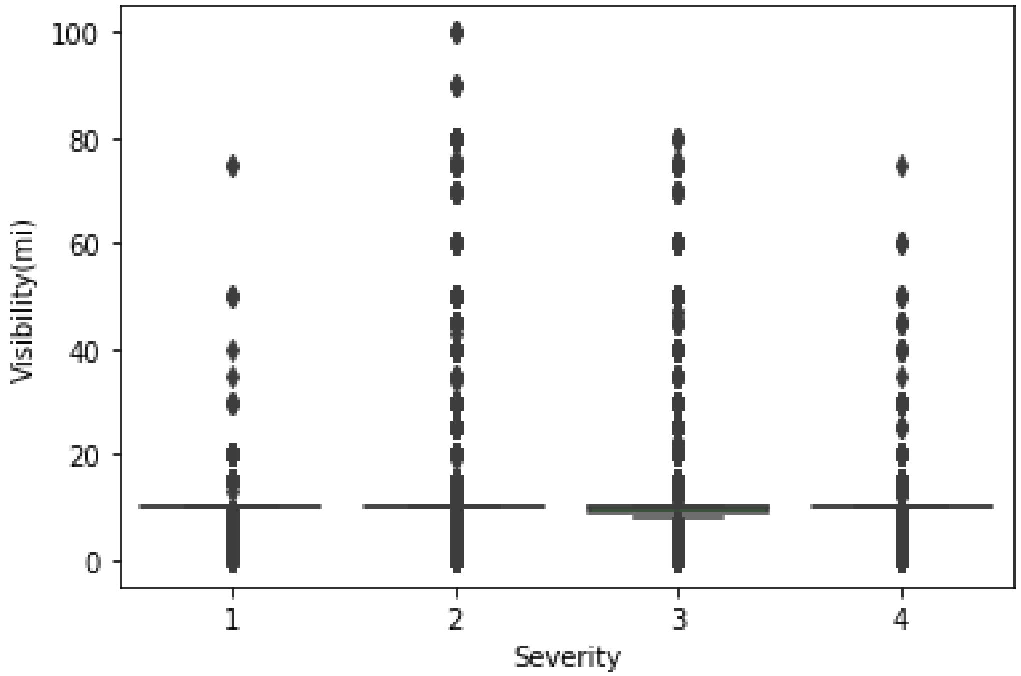


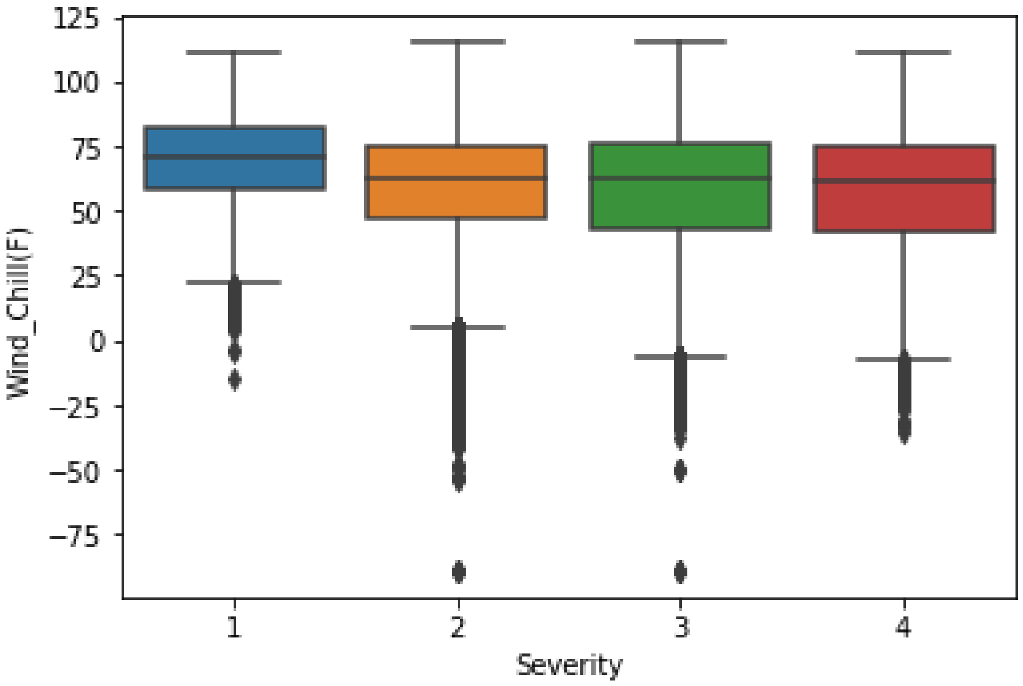


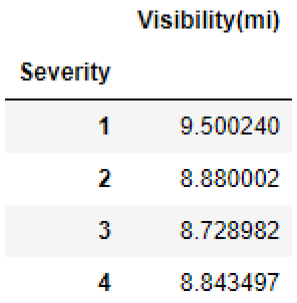
	Precipitation(in)
Severity	
1	0.005408
2	0.006966
3	0.011394
4	0.007558

Data Visualization and Analysis

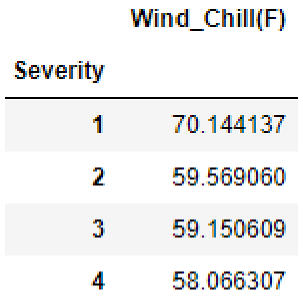
How does weather condition impact?







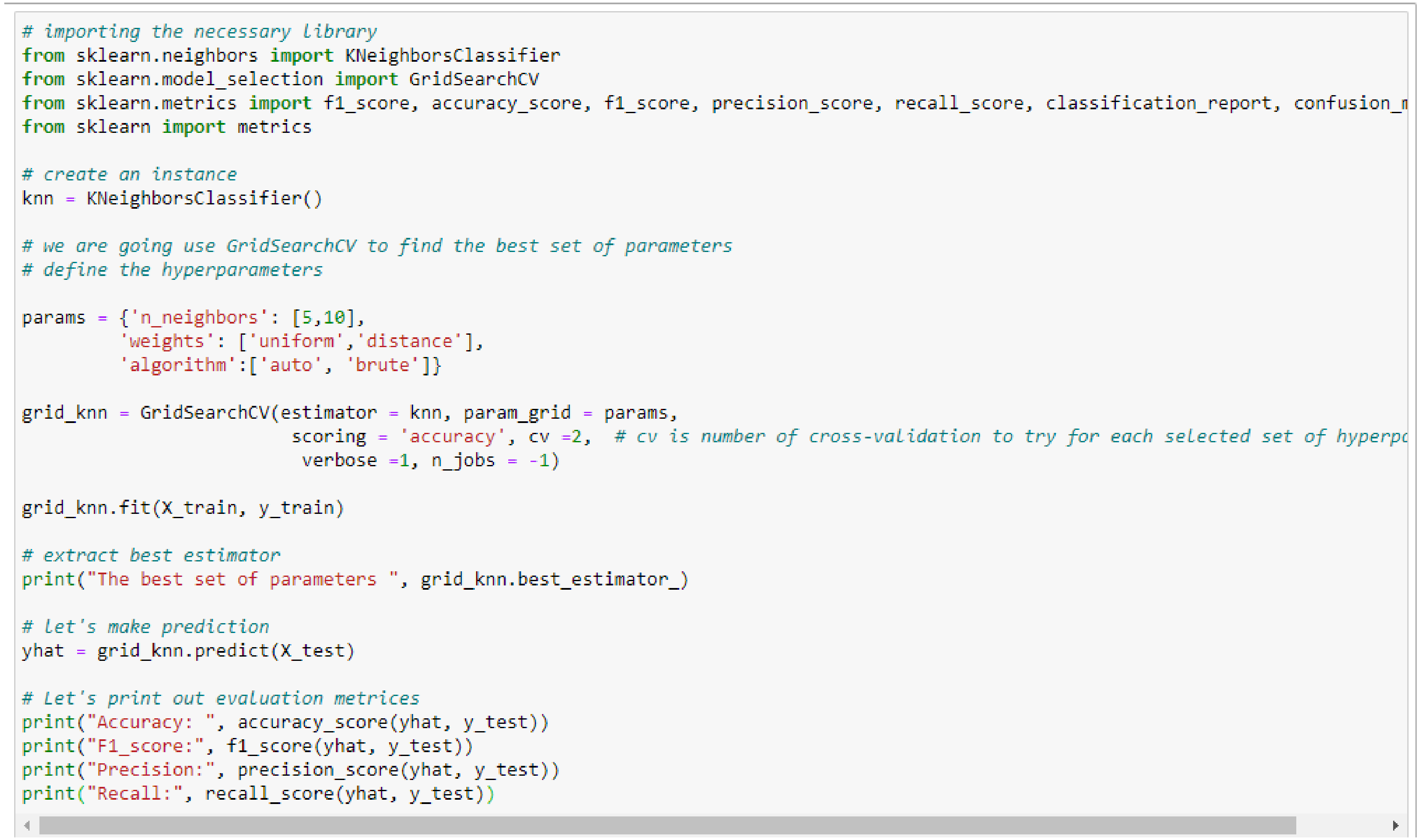
	Visibility(mi)
Severity	
1	9.500240
2	8.880002
3	8.728982
4	8.843497



	Wind_Chill(F)
Severity	
1	70.144137
2	59.569060
3	59.150609
4	58.066307

Building a Model

K Nearest algorithm is used.



```
# importing the necessary library
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import f1_score, accuracy_score, f1_score, precision_score, recall_score, classification_report, confusion_m
from sklearn import metrics

# create an instance
knn = KNeighborsClassifier()

# we are going use GridSearchCV to find the best set of parameters
# define the hyperparameters

params = {'n_neighbors': [5,10],
         'weights': ['uniform','distance'],
         'algorithm':['auto', 'brute']}

grid_knn = GridSearchCV(estimator = knn, param_grid = params,
                         scoring = 'accuracy', cv =2,  # cv is number of cross-validation to try for each selected set of hyperpa
                          verbose =1, n_jobs = -1)

grid_knn.fit(X_train, y_train)

# extract best estimator
print("The best set of parameters ", grid_knn.best_estimator_)

# let's make prediction
yhat = grid_knn.predict(X_test)

# Let's print out evaluation metrices
print("Accuracy: ", accuracy_score(yhat, y_test))
print("F1_score:", f1_score(yhat, y_test))
print("Precision:", precision_score(yhat, y_test))
print("Recall:", recall_score(yhat, y_test))
```

Tried to optimize the parameters but the computer couldn’t perform the task.

Result

- More accidents occurred during the day, typically at rush hour

- Outside temperature, precipitation, visibility, and wind chill impacts the seriousness of the accident.

- More accidents occurred in the proximity of traffic objects.

Recommendation

- More stringent rule should be implemented at junction or at the traffic lights.

- The drivers should be extra cautious when the weather is not ideal.

- If possible, the rush should be avoided.

Future Work

❖ Extend the model to cover other factors such as fatalities, and features like driver’s sex, speed, vehicle type.