IBM Datascience Certificate Program - Capstone Project

Khalid Sayyed

09/13/2020

Executive Summary

1. Business Problem:

- In the United State of America, more than 38,000 people die every year in crashes on roadways.
  The U.S. traffic fatality rate is 12.4 deaths per 100,000 inhabitants.

- In this study, we will be analyzing US accidents severity data in order to predict how dangerous driving conditions are. The target audience for this study will be general public/citizens of United States of America who are driving on roads.

- The study will analyze set of conditions that lead to high severity accidents.

- **Based on a given set of conditions on a particular day at a particular place, the algorithm will warn drivers of potential dangerous conditions.**

Executive Summary

Data:

- We will be using the US Accidents Severity data available on Kaggle.com.

- The data was collected between February 2016 to March 2019, using several data providers, including two APIs which provide streaming traffic event data. These APIs broadcast traffic events 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.

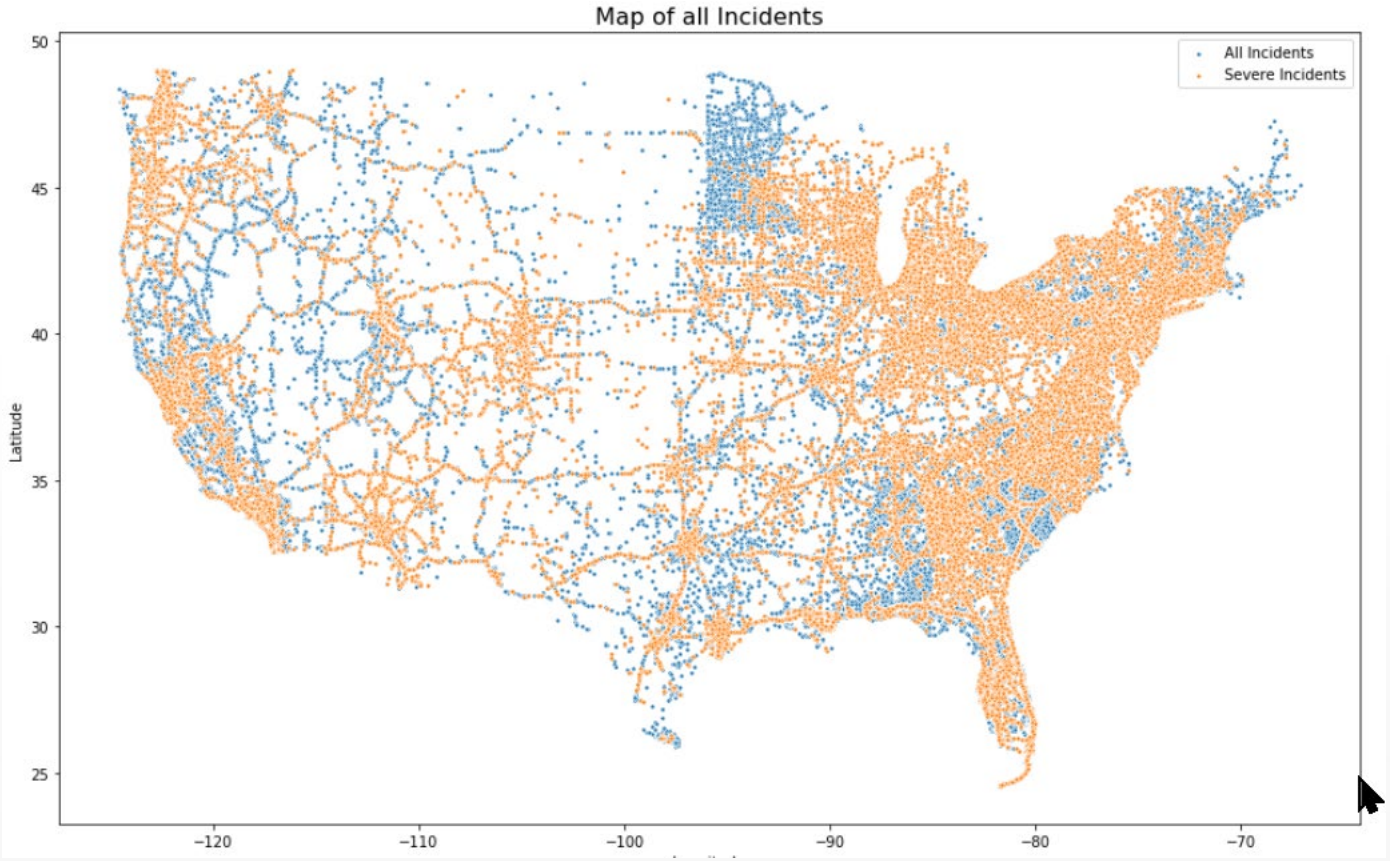
- The dataset contains 2,243,939(2.24 million) rows and 49 features. \

- The dataset contains geo-location features such as City, County, State, Street, Point of Interest features such as Crossing, Bump, Junction, Railway and weather features such as Wind Speed, Visibility, Humidity, Pressure etc.

Executive Summary

Methodology and Results:

Data Exploration and Manipulation



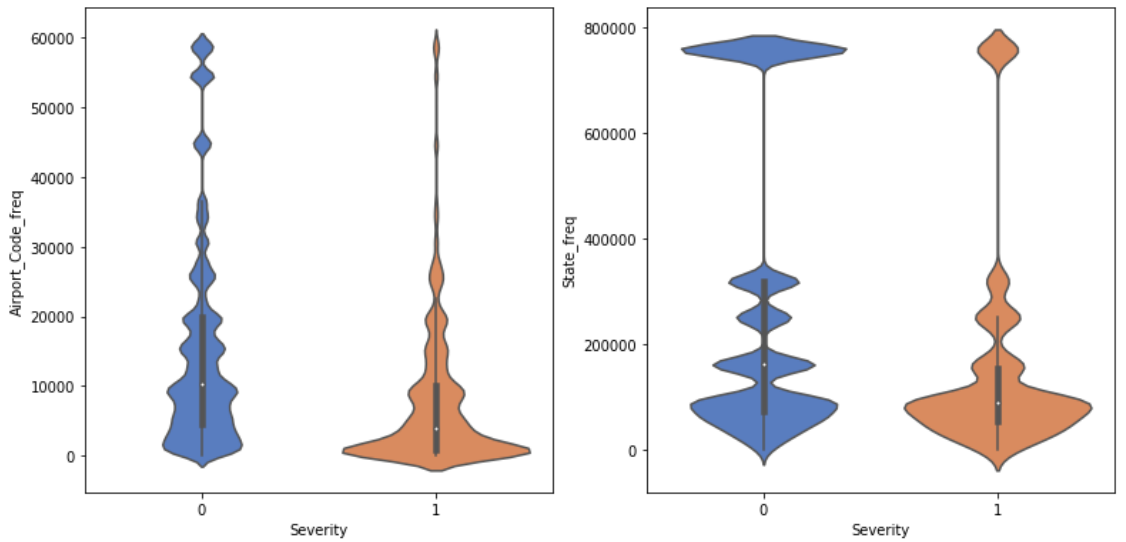
- From the map of accidents across latitudes and longitudes, it can be seen that majority of the accidents occur in east and west coast.
- This makes sense because of dense population in these regions of the US.
- Map (Especially at central USA) shows higher accidents along some lines, indicating interstate highways and cluster of higher accidents around major cities.

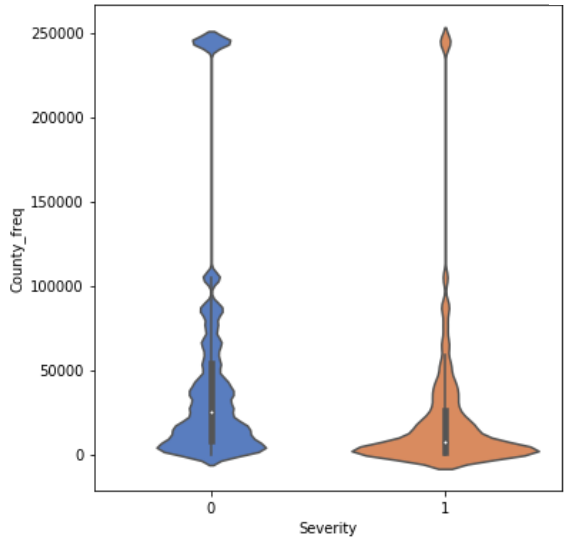
Executive Summary

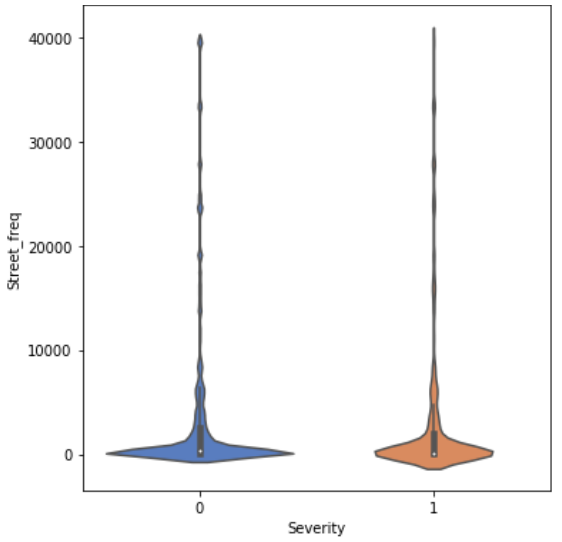
Methodology and Results:

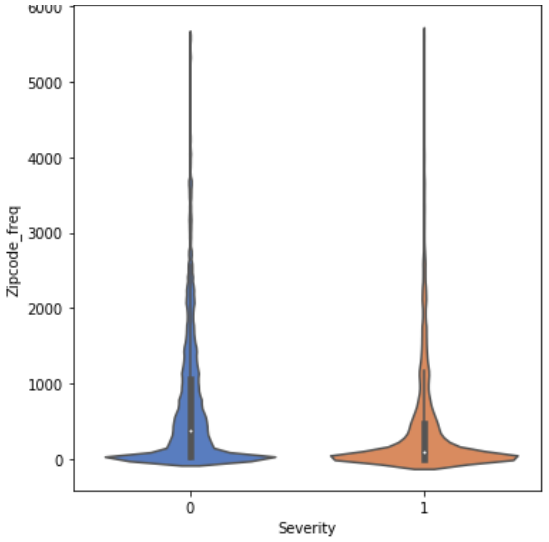
Data Exploration and Manipulation

- From violin plots of frequency of States, Counties, Streets, Zip Codes and Airport Codes, it can be seen that accidents are higher at some of these places than others.







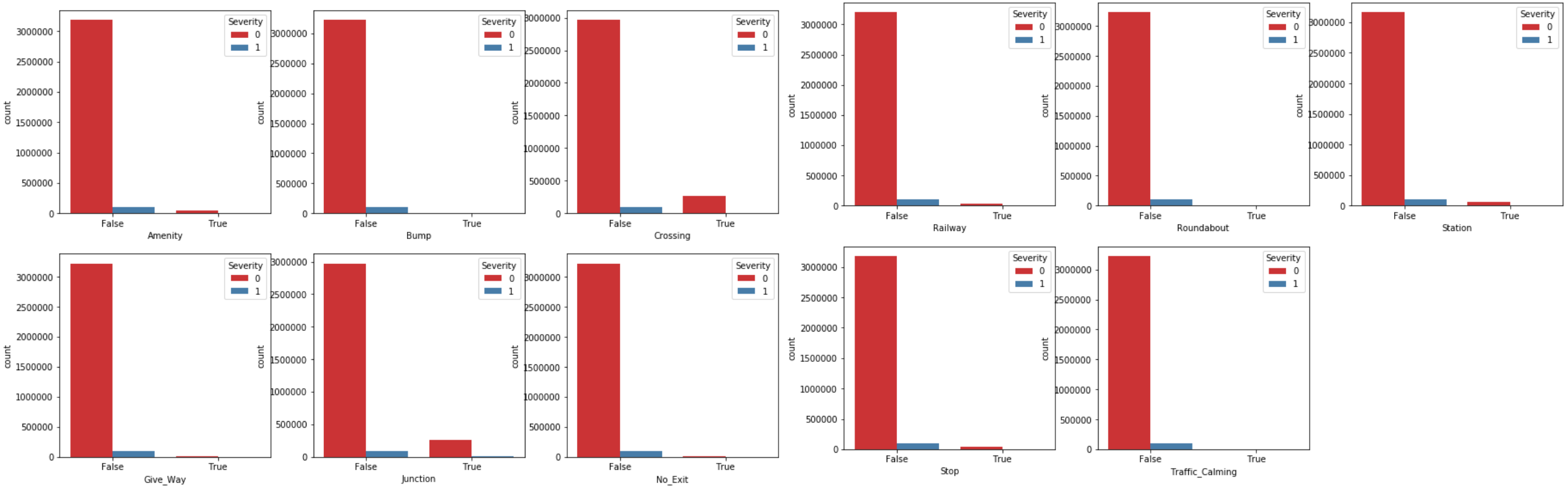


Executive Summary

Methodology and Results:

Data Exploration and Manipulation

- From bar charts of points of interest data, it can been seen the accidents are higher and sever at certain points of interest than others.

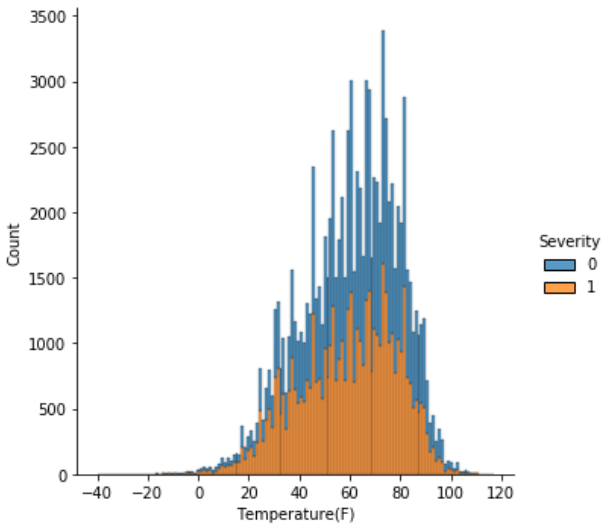


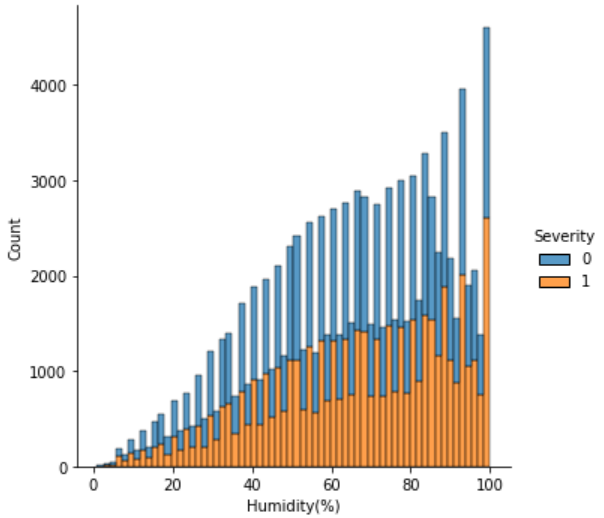
Executive Summary

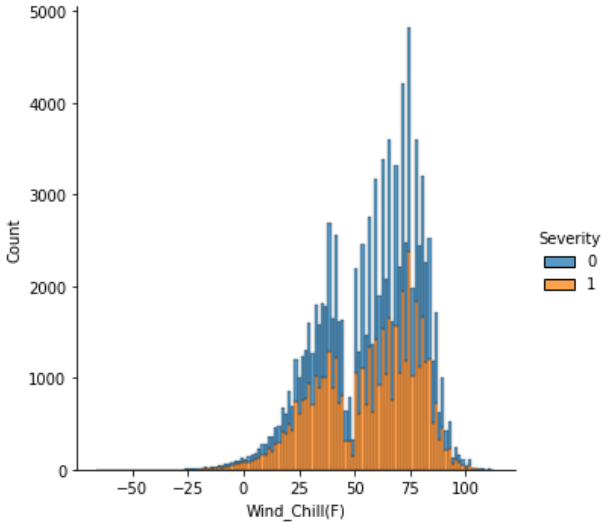
Methodology and Results:

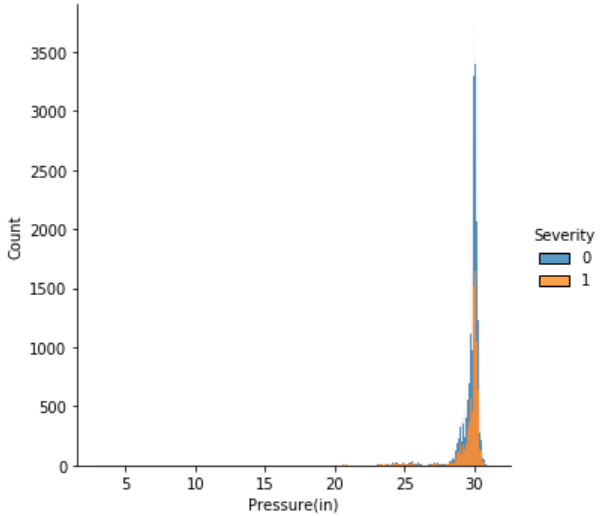
Data Exploration and Manipulation

- Weather conditions distribution plots show that accidents are higher and sever at some weather conditions such as during high humidity (probably during rain).







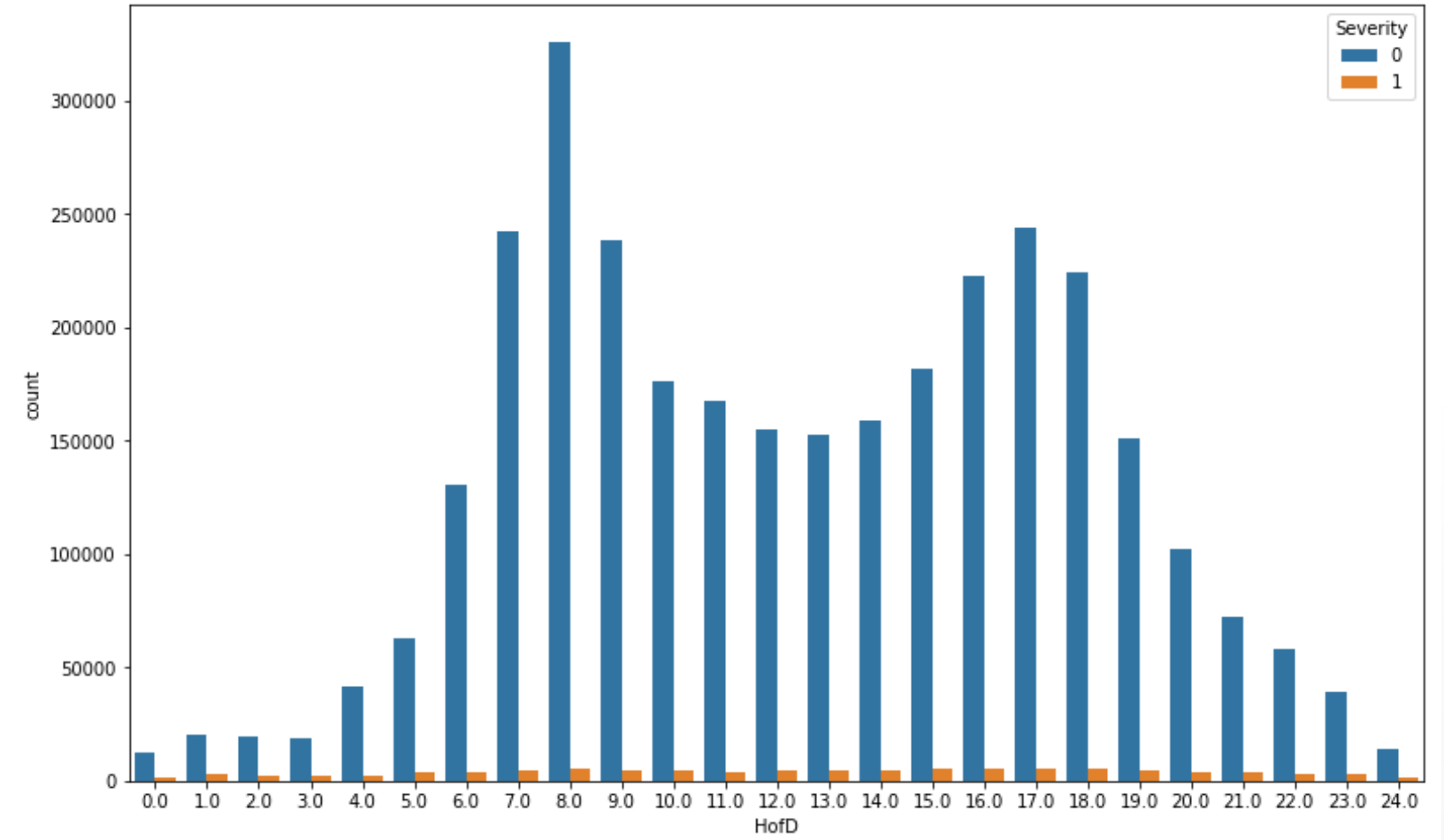


Executive Summary

Methodology and Results:

Data Exploration and Manipulation

- Hour of the day vs accidents plot shows higher accidents during morning and evening busy hours.



Executive Summary

Methodology and Results:

Machine Learning

- Finally, machine learning decision trees shows an accuracy of 92% and 85% on train and test data to predict severe accidents.

- **Hence we can conclude that the dataset can be used in predicting accidents severity and warning drivers of potential driving hazards.**

Fit decision Tree with Grid Search CV

[45]:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV
DT_grid = { 'min_samples_split': [5,10, 20, 30, 40],
            'max_features': [None, 'log2', 'sqrt']}
CV_DT = GridSearchCV(DecisionTreeClassifier(random_state=42), DT_grid, verbose=1, cv=3)
CV_DT.fit(X_train, y_train)

print('Best Parameters: ', CV_DT.best_params_)


from sklearn import tree
# Training step, on X_train with y_train
tree_clf = tree.DecisionTreeClassifier(min_samples_split = 40)
tree_clf = tree_clf.fit(X_train,y_train)

tree_accuracy_train = tree_clf.score(X_train, y_train)
print("Train Accuracy:", (tree_accuracy_train*100))
tree_accuracy_test = tree_clf.score(X_test,y_test)
print("Test Accuracy: ", (tree_accuracy_test*100))

prediction = tree_clf.predict(X_test)
```

```
Fitting 3 folds for each of 15 candidates, totalling 45 fits
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done  45 out of  45 | elapsed:   21.6s finished
Best Parameters:  {'max_features': None, 'min_samples_split': 40}
Train Accuracy: 92.1359375
Test Accuracy:  85.91250000000001
```