Capstone Project - Car accident severity

Introduction

For the final capstone project in the IBM course, we need to analyze the "severity" of the accident in terms of casualties, traffic delays, property damage, or any other types of accidents. The data is collected bu the Seattle SPOT Traffic Management Department and provided by Coursera via the link (https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv). The data set is updated weekly, from 2004 to present. It contains information such as severity code, address type, lication, collision type, weather, road conditions, speeding, etc.
We can begin to solve this complex problem by studying the data of past accidents, and gain insight into the factors that affect the severity of the accident, and then use machine learning technology to create a model based on its initial similarity to predict the severity of future the historical record of the accident. Moreover, there are two main beneficiaries of this model: 1. city planner, who can use the model to inform their road planning and traffic strategies; 2.emergency service providers, who can use the model to forecast the severity of the accident based on the information provided when the accident is reported, so that resources can be optimally allocated throughout the city.

Data

Our predictor or target variable will be "SEVERITYCODE", because it is used to measure the severity of accidents from 0 to 3. The attributes used to measure the severity of the accident are "WEATHER", "ROADCODE", "LIGHTCODE".
A code that corresponds to the severity of the collision:

- 3 - fatality
- 2b - serious injury
- 2 - injury
- 1 - prop damage
- 0 - unknown

Other important variables include:

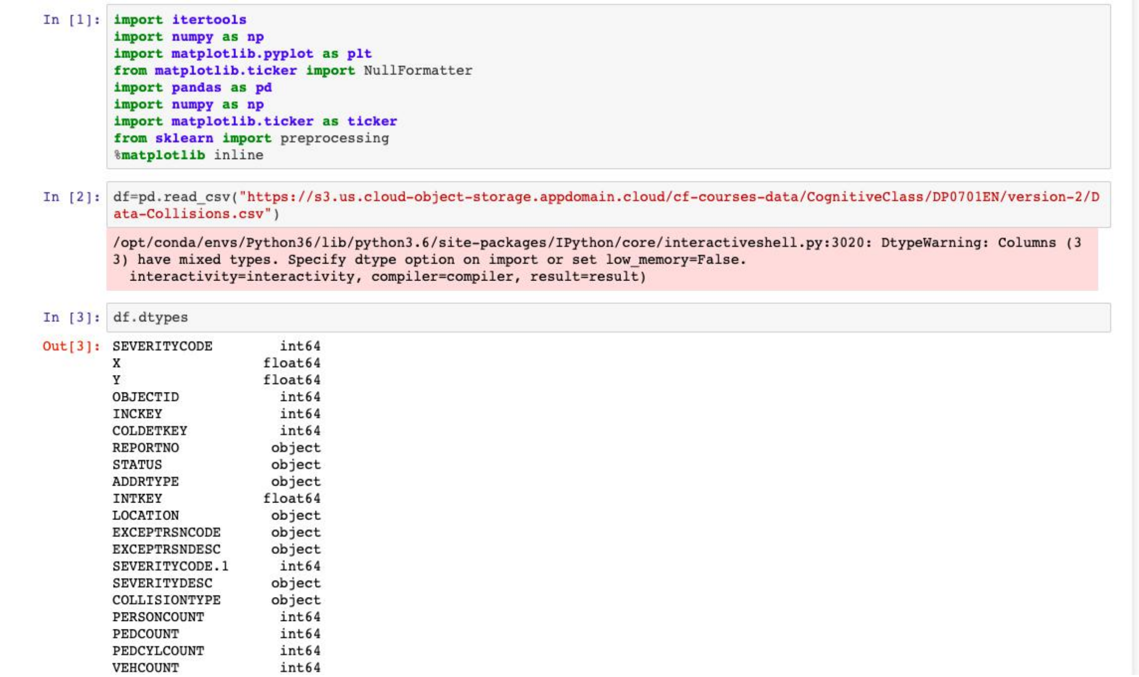
- LOCATION: Description of the general location of the collision.
- SEVERITYDESC: A detailed description of the severity of the collision.
- INCDATE: The date of the incident.
- INCDTTM: The data and time of the incident.
- JUNCTIONTYPE: Category of junction at which collision took place.
- INATTENTIONIND: Whether or not collision was due to inattention.
- UNDERINFL: Whether or not a driver involved was under the influence of drugs or alcohol.
- WEATHER: A description of the weather conditions during the time of the

collision.

- ROADCOND: The condition of the road during the collision.
- LIGHTCOND: The light conditions during the collision.
- PEDROWNOTGRNT: Whether or not the pedestrian right of way was not granted.
- SDOTCOLNUM: A number given to the collision by SDOT.
- SPEEDING: Whether or nor speeding was a factor in the collision.
- SEGLANEKEY: A key for the lane segment in which the collision occurred.
- CROSSWALKKEY: A key for the crosswalk at which the collision occurred.
- HITPARKEDCAR: Whether or not the collision involved hitting a parked car.

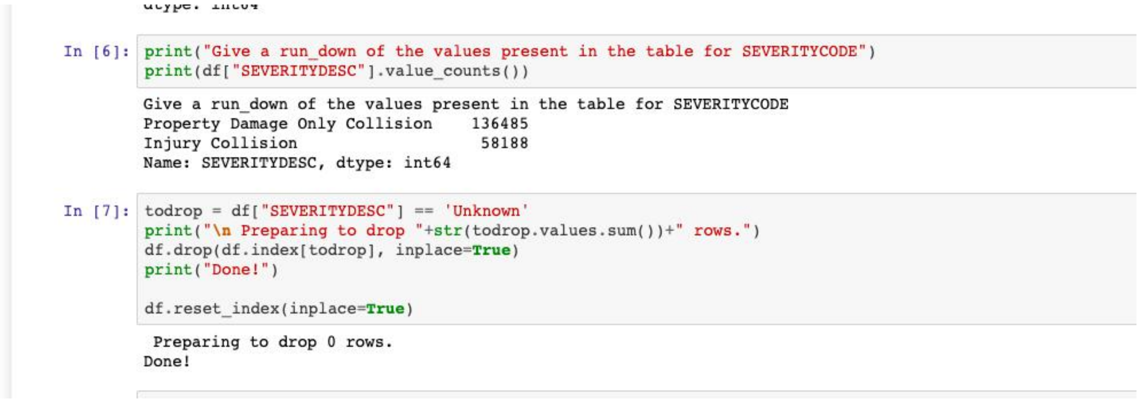
Methodology

We used Jupyter Notebook to do the data analysis. First, we imported the libraries, that are two libraries that we ended up using were Pandas and Numpy. And then, we read the provided .csv and listed the datatypes.



```
In [1]: import itertools
        import numpy as np
        import matplotlib.pyplot as plt
        from matplotlib.ticker import NullFormatter
        import pandas as pd
        import numpy as np
        import matplotlib.ticker as ticker
        from sklearn import preprocessing
        %matplotlib inline

In [2]: df=pd.read_csv("https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/D
        ata-Collisions.csv")

        /opt/conda/envs/Python36/lib/python3.6/site-packages/IPython/core/interactiveshell.py:3020: DtypeWarning: Columns (3
        3) have mixed types. Specify dtype option on import or set low_memory=False.
          interactivity=interactivity, compiler=compiler, result=result)

In [3]: df.dtypes

Out[3]: SEVERITYCODE          int64
        X                   float64
        Y                   float64
        OBJECTID              int64
        INCKEY                int64
        COLDETKEY             int64
        REPORTNO             object
        STATUS               object
        ADDRTYPE             object
        INTKEY              float64
        LOCATION             object
        EXCEPTRSNCODE        object
        EXCEPTRSNDESC        object
        SEVERITYCODE.1        int64
        SEVERITYDESC         object
        COLLISIONTYPE        object
        PERSONCOUNT           int64
        PEDCOUNT              int64
        PEDCYLCOUNT           int64
        VEHCOUNT              int64
```

Next, we need to addressed a lot of issues such as: rows which are missing information about the target variable; columns containing useless data; presence of features with categorical values.



```
In [6]: print("Give a run_down of the values present in the table for SEVERITYCODE")
        print(df["SEVERITYDESC"].value_counts())

        Give a run_down of the values present in the table for SEVERITYCODE
        Property Damage Only Collision    136485
        Injury Collision                   58188
        Name: SEVERITYDESC, dtype: int64

In [7]: todrop = df["SEVERITYDESC"] == 'Unknown'
        print("\n Preparing to drop "+str(todrop.values.sum())+" rows.")
        df.drop(df.index[todrop], inplace=True)
        print("Done!")

        df.reset_index(inplace=True)

         Preparing to drop 0 rows.
        Done!
```

The traffic accident records on the Seattle Open Data Portal include the latitude and longitude (X, Y, respectively) of each accident that occurred within the city council area. Let's create a map in Folium to see where it happened.

In [27]:

```
!pip install folium
import pandas as pd
import folium
```

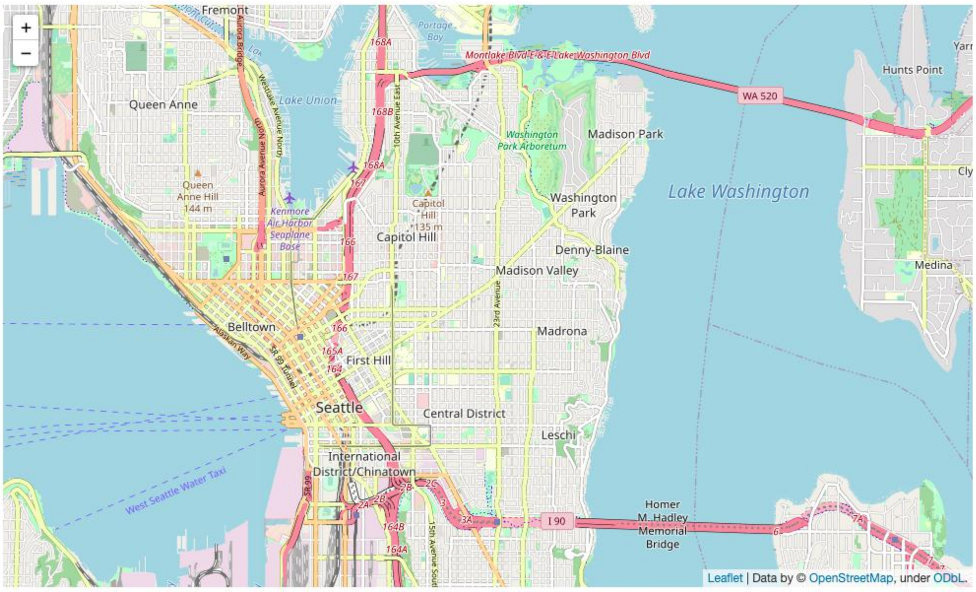
```
Requirement already satisfied: folium in /opt/conda/envs/Python36/lib/python3.6/site-packages (0.11.0)
Requirement already satisfied: branca>=0.3.0 in /opt/conda/envs/Python36/lib/python3.6/site-packages (from folium)
(0.4.1)
Requirement already satisfied: requests in /opt/conda/envs/Python36/lib/python3.6/site-packages (from folium) (2.21.
0)
Requirement already satisfied: numpy in /opt/conda/envs/Python36/lib/python3.6/site-packages (from folium) (1.15.4)
Requirement already satisfied: jinja2>=2.9 in /opt/conda/envs/Python36/lib/python3.6/site-packages (from folium) (2.1
0)
Requirement already satisfied: idna<2.9,>=2.5 in /opt/conda/envs/Python36/lib/python3.6/site-packages (from requests-
>folium) (2.8)
Requirement already satisfied: urllib3<1.25,>=1.21.1 in /opt/conda/envs/Python36/lib/python3.6/site-packages (from re
quests->folium) (1.24.1)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in /opt/conda/envs/Python36/lib/python3.6/site-packages (from re
quests->folium) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/envs/Python36/lib/python3.6/site-packages (from reque
sts->folium) (2020.6.20)
Requirement already satisfied: MarkupSafe>=0.23 in /opt/conda/envs/Python36/lib/python3.6/site-packages (from jinja2>
=2.9->folium) (1.1.0)
```

In [28]:

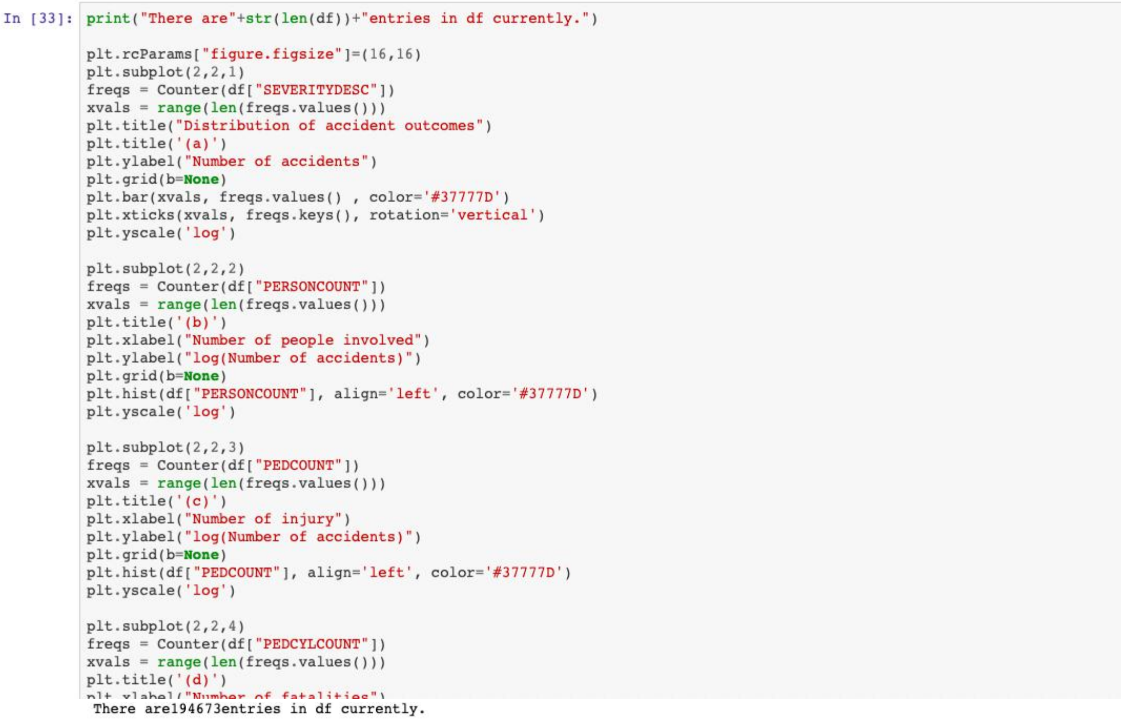
```
lon_med = data["X"].mean()
lat_med = data["Y"].mean()

print(lat_med)
print(lon_med)

seattle_map = folium.Map(location=[lat_med, lon_med], zoom_start=11)
seattle_map
```

```
47.619542517688615
-122.33051843904114
```

Out[28]:

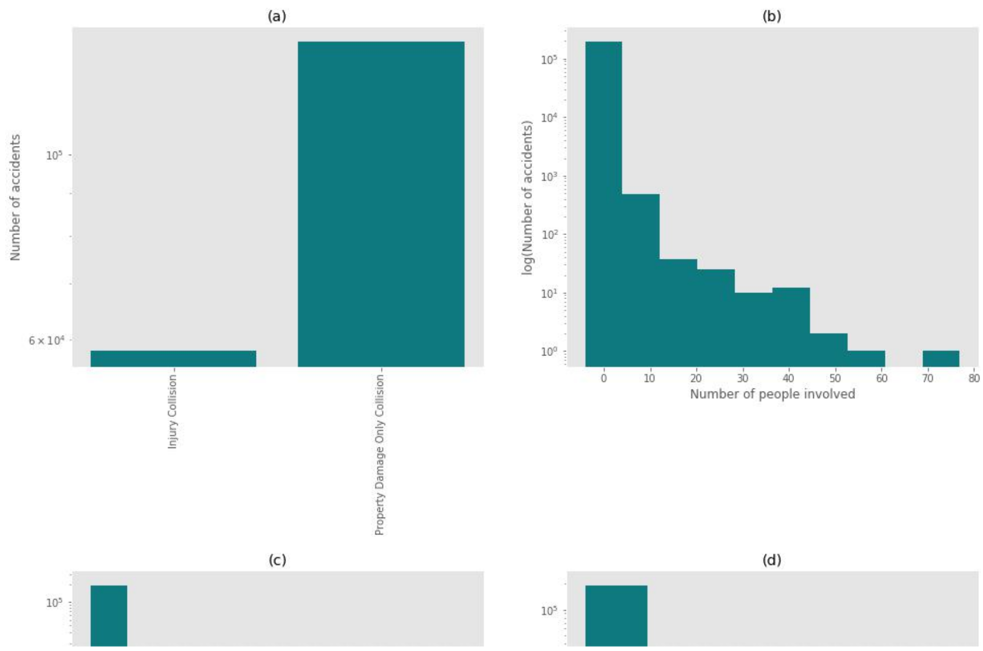


And in this study, I focused more on the graphical data and the value count for different categories.



```
In [33]: print("There are"+str(len(df))+"entries in df currently.")

         plt.rcParams["figure.figsize"]=(16,16)
         plt.subplot(2,2,1)
         freqs = Counter(df["SEVERITYDESC"])
         xvals = range(len(freqs.values()))
         plt.title("Distribution of accident outcomes")
         plt.title('(a)')
         plt.ylabel("Number of accidents")
         plt.grid(b=None)
         plt.bar(xvals, freqs.values() , color='#37777D')
         plt.xticks(xvals, freqs.keys(), rotation='vertical')
         plt.yscale('log')

         plt.subplot(2,2,2)
         freqs = Counter(df["PERSONCOUNT"])
         xvals = range(len(freqs.values()))
         plt.title('(b)')
         plt.xlabel("Number of people involved")
         plt.ylabel("log(Number of accidents)")
         plt.grid(b=None)
         plt.hist(df["PERSONCOUNT"], align='left', color='#37777D')
         plt.yscale('log')

         plt.subplot(2,2,3)
         freqs = Counter(df["PEDCOUNT"])
         xvals = range(len(freqs.values()))
         plt.title('(c)')
         plt.xlabel("Number of injury")
         plt.ylabel("log(Number of accidents)")
         plt.grid(b=None)
         plt.hist(df["PEDCOUNT"], align='left', color='#37777D')
         plt.yscale('log')

         plt.subplot(2,2,4)
         freqs = Counter(df["PEDCYLCOUNT"])
         xvals = range(len(freqs.values()))
         plt.title('(d)')
         plt.xlabel("Number of fatalities")
```

```
There are194673entries in df currently.
```



Otherwise, we also can develop other models like:

1.Logistic Regression Model: because of the data only provides two severity code , the model will only predict one of those two classes. So that, the model converts the severity of the accident into a binary variable (no injury is 0, for severe injury accident is 1). We can use logistic regression technology to try to classify the consequences of the accident.

2.Decision Tree model: The model provides us with a layout of all possible outcomes, so we can fully analyze the consequences of the decision. According to this model, the tree most important characteristics used to determine the likelihood of an accident with serious consequences are: 1) the number of pedestrians involved; 2) the number of vehicles involved, 3) whether there are one or more the driver was speeding.

3. The KNN model: The model will help us predict the severity code of the result by finding the most similar data point within k distance. But if too Neighbors are

included in the classification, this model may completely lose all diagnostic capabilities.

Conclusion

The project and analysis are helpful to the Seattle Transportation Department. Based on historical data from weather conditions pointing to certain categories, we can conclude that specific weather conditions will affect whether the trip will cause property damage or injury. Therefore, the traffic management department can try to improve safety instructions or other factors that can reduce accidents.

In addition, by predicting the severity of an accident based on weather, date, location, and road conditions, the model can allow emergency service call handlers to prioritize resources that have collided, which may be prioritized to help them make decisions. .

Finally, there are more accidents in some places during dark times. For those places, adding lights may be a good way to reduce collisions.

References

https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv

https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Metadata.pdf