Predicting the Severity type of Collision

Lingyu Ni

1. Business Understanding

- **Background**
  The car collision take place everyday around world. How severe could the collision be, ist the first question we would ask. Therefore, it is advantageous for analysis to accurately predict whether the collision leads to people injured or just proper damaged.

- **Business Problem**
  As we know, the severity of the collision is definitely influenced by some variables, for example, the pedestrian on the road, the address types, etc.

- **Interest**
  Given those Information, we could build a model and predict the severity of the collision, and response with appropriate reaction.
  In this case we set SEVERITYCODE as dependent value, and choose other multiple independent variables to build a classification model.

2. Data understanding

- **Data resource**
  I choose data provided by causer course, url: "https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv", totally 38 columns.

- **Data cleaning and preparing**

  Identification of the columns with no meaningful value
  At first look at all 38 columns, we could find that the feature like “INPORTNO” or “OBJECTIF" has no meaningful utility. Some columns have duplicated information with different data form, e.g. value with type “object” just as the description of another column with numerical value. How to deal with them? Ignore or just delete them

  Dealing with missing data
  By using df.isnull().sum() we could find some of the columns are lack of huge amount of information. There are 2 types of missing information:

- Feature "SPEEDING", "INATTENTIONIND", "PEDROWNOTGRNT", are only filled with value "Y" or "N" if the situation is confirmed, others are just empty. This kind of value can not simply regarded as missing value, it do contains value.
  Solution: fill the column value with dummy value "1" or “0"
- Feature “INTKEY","EXCEPTRSNCODE","SDOTCOLNUM" missed over 100,000 value that make no sense to take this value into account.

  Data formatting

- The data type would also influence the analysis of data—use df.dtypes to confirm. Some with data type of “object" as actually can work as numerical data, make sure that transfer the data into numerical data by using df[‘’].int() or pd.to_numeric().
  The purpose of that is to get as possible as much of value that can be applied into correlation analysis.
- The collision information were typed in from different people, or combined from different databank, thus the different data format is unavoidable. For example, there are 4 kinds of data in column “UNDERINFL”—“Y”,”N”,”1","0", where “Y” and “1" mean the same thing, that the driver are under influence of drug or alcohol.
  Solution: convert the all the value of this feature into dummy value. To notice: the datatype

should all be int64, or there would be different type of data.

```
[19]: df["UNDERINFL"].value_counts()
```

```
[19]: N    100274
      0     80394
      Y      5126
      1      3995
      Name: UNDERINFL, dtype: int64
```

Attribute filled with 4 kind value: "Y","N","0","1", replace "Y" with 1, replace "N" with 0

- As for Features like "Weather condition", “Drive condition”, “Address condition” etc. Firstly use value_counts() command to identify how much different genre belong to this feature. Than we found there are different expressions of same kind of genre, we need to combine it into same genre by replacing a standard value.

```
[31]: df1["WEATHER"].value_counts()
```

```
[31]: Clear                       108828
      Raining                      31980
      Overcast                     27099
      Unknown                      13846
      Snowing                        888
      Other                          765
      Fog/Smog/Smoke                 553
      Sleet/Hail/Freezing Rain       112
      Blowing Sand/Dirt               49
      Severe Crosswind                24
      Partly Cloudy                    5
      Name: WEATHER, dtype: int64
```

```
[36]: df1["WEATHER"]=df1["WEATHER"].replace(["Unknown"],"Other")
      df1["WEATHER"]=df1["WEATHER"].replace(["Sleet/Hail/Freezing Rain"],"Raining")
      df1["WEATHER"]=df1["WEATHER"].replace(["Partly Cloudy"],"Overcast")
```

- If we want to analyze the influence of time upon severity of collision, i.e. if the weekday has higher potential to cause severe collision, or, if some time period of a day has higher risk of

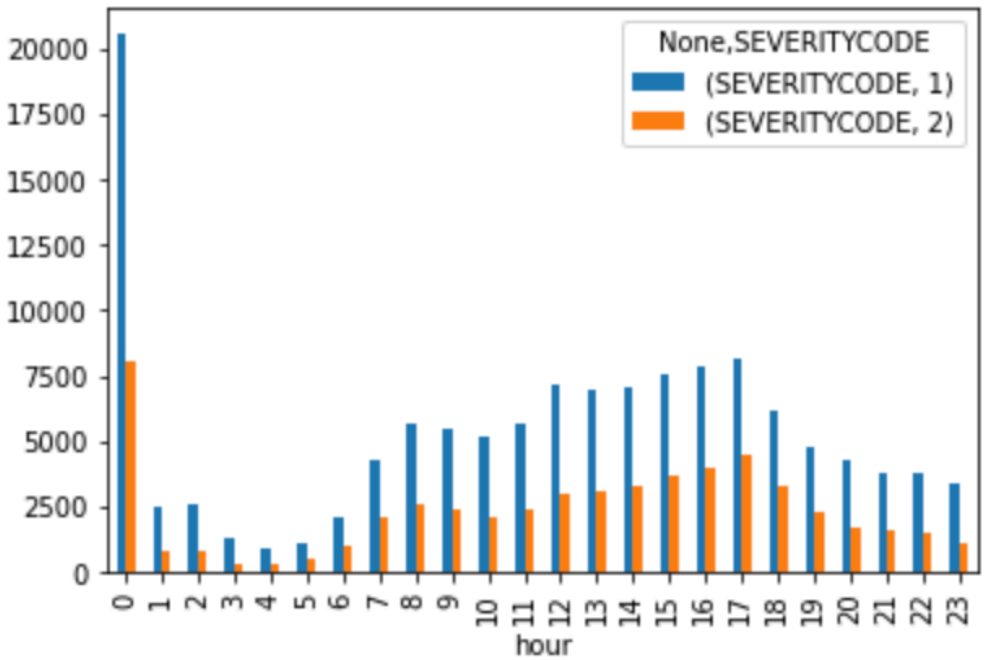

Figure 1 : the count of collision in each time period with comparison between Severitycode 1 and 2

severe collision like "morning peak “.
Solution: convert the value of feature into “datetime" type, and use “dt.weekofday” and “dt.hour” to extract expected information, then append the dataset with this columns.
From statistical visualization, we could find: t’s abnormal that the collision happens at midnight has a number that extremely high. After comparing with initial dataset, we could find that some of the value of "INCDTTM" does not include hour information--> so we should exclude the influence of it.
The good news is, we could find a time related trend of collision cases: in time period between around 17, there is relative larger quantity of collision. Bad news is, it’s not significant change of relationship between severity1 and 2. It need later analysis.

- **Data exploration and extraction**

  For numerical value, I first use df.corr() to identify the features that have relative higher correlation with SEVERITYCODE. Here I filter with threshold 0.1, and extract 7 numerical value to build Feature dateset: "PERSONCOUNT","PEDCOUND", “PEDCYLCOUNT","SDOT_COLCODE","PEDROWNOTGRNT","SEGLANEKEY","CROSSWALKKEY"

  Using statistical analysis to visualize the condition of object data and location data

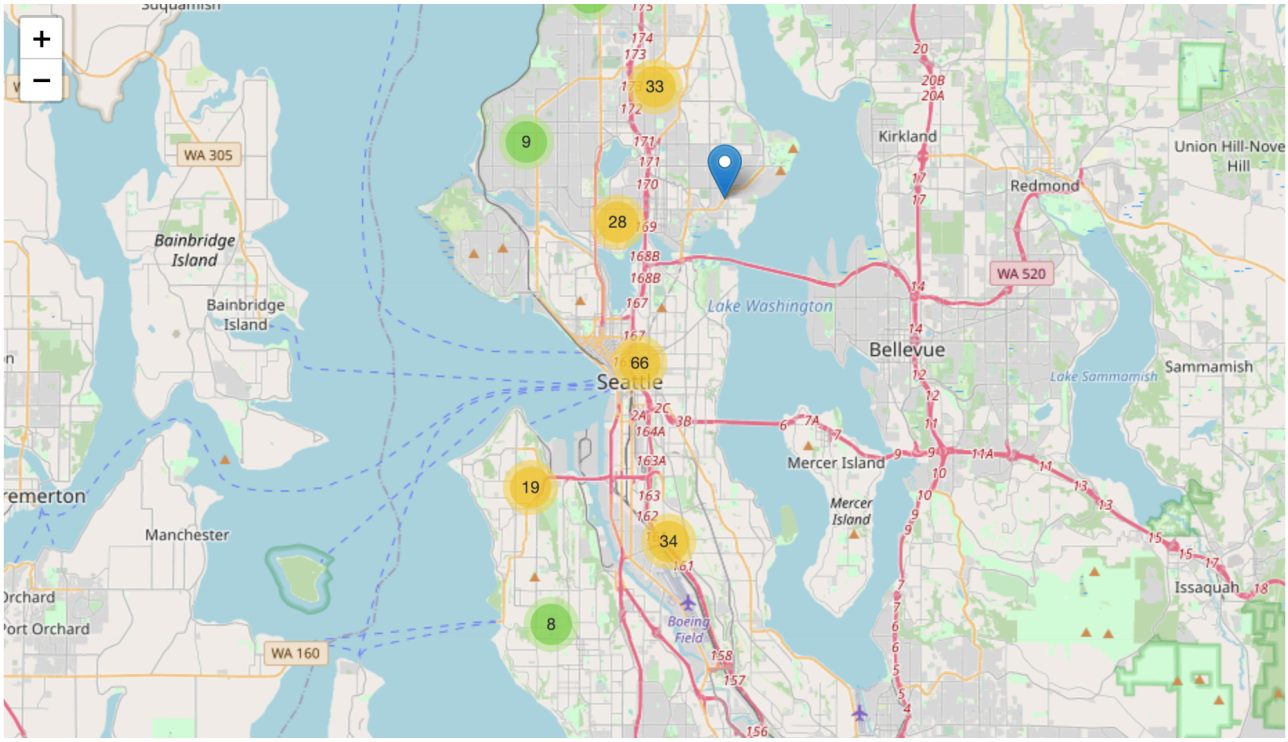

Figure 2: map of collision

- As Figure 2 shows, cars have higher risk of collision when driven around downtown area, where has more intersection and number of pedestrian. Along the main road (red line in the map), there is higher frequency of collision.

- As figure 3 shows below, most of collision happen in block. In block most of cases are type 1 severity, whereas in intersection area, the possibility of severity type 2 is higher.
  Conclusion: the probability of collision severity 1 and 2 could be significantly influenced by address dype.

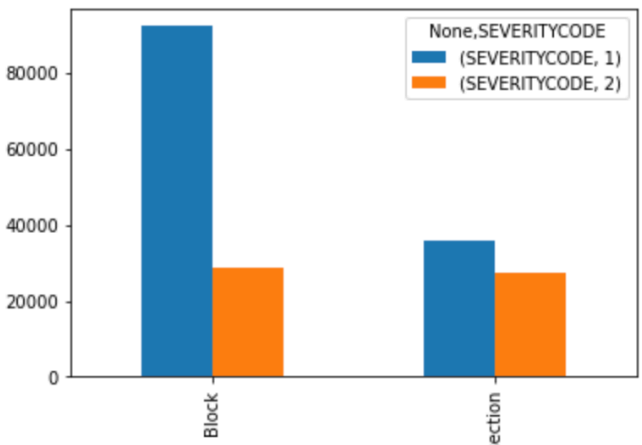

Figure 3: relationship between Address type and severity

- As for “weather condition”, “road condition”, “light condition” , it seems that there is no big relationship with severity of collision. Most cases happened in a relative good condition, e.g. dry road condition rather than wet condition, day light or dark with light on.

To conclusion I select following columns as feature dataset:

"PERSONCOUNT","PEDCOUNT","PEDCYLCOUNT", “SDOT_COLCODE”,"PEDROWNOTGRNT", “SEGLANEKEY”,”CROSSWALKKEY”,"ADDRTYPE"
where convert “ADDRTYPE" as dummy values.

3. Predictive Modeling

I use classification for modeling, where i use approach KNN.
first step: find the best k form modeling:

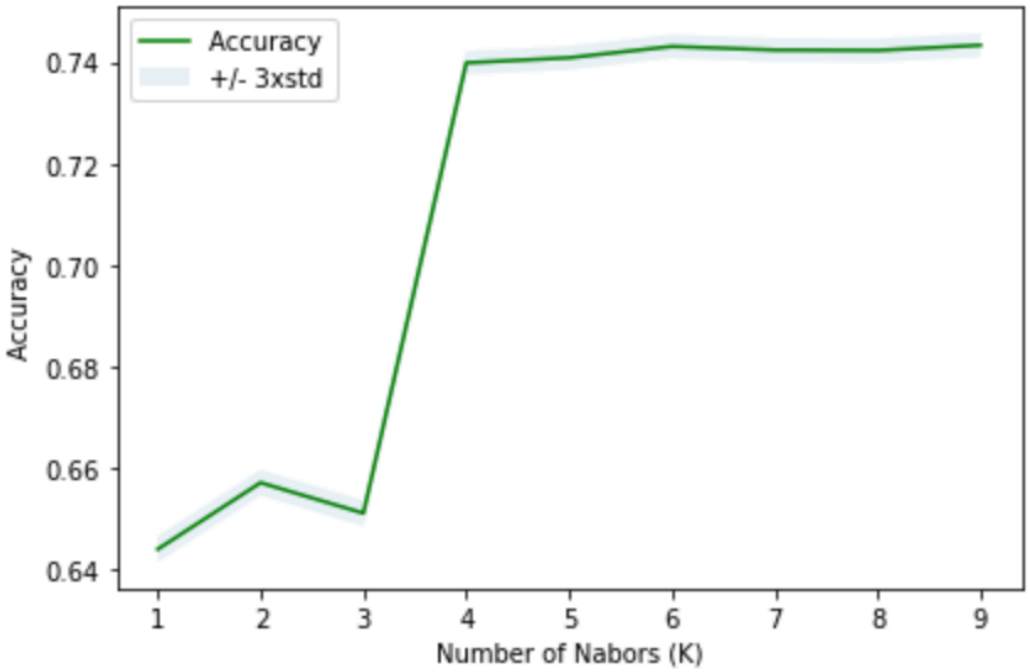

Figure 4: accuracy of different K Nabors.

When k>3, the accuracy improve significantly up to around 0.74 and relative stable--> here: select k=4 (figure 4)

4. Model test and evaluation

KNN Jaccard index: 0.74
KNN F1-score: 0.72