Car Accident Severity Code

<submission, week 3>

Yonghoon, Park

Index

Index of picture

I. Introduction

1. A description of the problem

1-1. Background

since the invention of the automobile road and highways become one of the largest life takers as mortality rates are higher than ever in the past 10 years. Each year around 20-30 million people get into a road accident. In which around 10% of those lose their lives. Road accidents are a serious shame for our society and still we are not in a state to reduce it. Most of the innocent casualties are of the pedestrians, cyclists and the bikers and between the age of 20-35 years, the future of any country and the solo earners of a family. Naturally, a question to be asked is what is the cause of such high percentage of car accident.

1-2. Problem Description

We have to gather old accident record and its severity for place with other information like location of accident, number of people involved, number of pedestrians, number of vehicles time and date of accident, way of accident, road condition, lighting and weather at place of accident and create a machine learning model with this data so that lots if we pass the following details, the model can predict the severity to us.

Predicting the data will help us to get cautious in advance and then it will be the person's choice to whether they should proceed on the same road with precaution or do a detour to a safer road.

In this project we will try to predict injury collision and Property Damage Only collision, this report are aim to stakeholders interested preventing and reducing injury collisions and Property Damage Only collision.

II. Data

The data that will be used in the model building will be the collision data from Seattle. It includes all collisions provided by SPD and recorded by traffic records. The data has been recorded from 2004 until present, updated weekly. Below shows the first five rows of the data table.

```
#import necessary programs
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
```

<Picture 2-1. header(code)>

	SEVERITYCODE	X	Y	OBJECTID	INCKEY	COLDETKEY	REPORTNO	STATUS	ADDRTYPE	INTK
0	2	-122.323148	47.703140	1	1307	1307	3502005	Matched	Intersection	3747
1	1	-122.347294	47.647172	2	52200	52200	2607959	Matched	Block	NaN
2	1	-122.334540	47.607871	3	26700	26700	1482393	Matched	Block	NaN
3	1	-122.334803	47.604803	4	1144	1144	3503937	Matched	Block	NaN
4	2	-122.306426	47.545739	5	17700	17700	1807429	Matched	Intersection	3438

5 rows × 38 columns

<Picture 2-2. Severity code>

There are 37 attributes in the data. Among them, severity code (*SEVERITYCODE*), which categorizes the severity of the accident from 0 (unknown) to 3 (fatal), will be our target value. As for attributes that will be used to predict the target value, collision address type (*ADDRTYPE*), weather condition (*WEATHER*) and road condition (*ROADCOND*) will be used.

```
In [4]: df=collision_data[['SEVERITYCODE', 'ADDRTYPE', 'WEATHER', 'ROADCOND']]
        df.head()
```

Out[4]:

	SEVERITYCODE	ADDRTYPE	WEATHER	ROADCOND
0	2	Intersection	Overcast	Wet
1	1	Block	Raining	Wet
2	1	Block	Overcast	Dry
3	1	Block	Clear	Dry
4	2	Intersection	Raining	Wet

<Picture 2-3. Collision relationship(code)>

shows the table with only the target and attributes stated above.

III. Methods

3-1. Data pre-processing

This section will pre-process the data before model building. Each attributes and target will be processed. The data type is shown below.

```
In [5]: df.dtypes

Out[5]: SEVERITYCODE     int64
        ADDRTYPE        object
        WEATHER         object
        ROADCOND        object
        dtype: object
```

<Picture 3-1. Data pre-processing>

3-2. Severity code

The unique values of the data are shown below.

```
In [6]:  df['SEVERITYCODE'].unique()

Out[6]:  array([2, 1])
```

<Picture 3-2. Severity code>

The severity code target has values of 1 and 2. From the metadata, 1 indicates property damage and 2 indicates injury.

3-3. Address type

Remove non values, convert categorical variables into quantitative variables.

In [7]:
```
df['ADDRTYPE'].unique()
```

Out[7]:
```
array(['Intersection', 'Block', 'Alley', nan], dtype=object)
```

In [8]:
```
df.dropna()
df=pd.concat([df, pd.get_dummies(df['ADDRTYPE'])], axis=1)
df.drop(['ADDRTYPE'], axis=1, inplace=True)
df.head()
```

Out[8]:

	SEVERITYCODE	WEATHER	ROADCOND	Alley	Block	Intersection
0	2	Overcast	Wet	0	0	1
1	1	Raining	Wet	0	1	0
2	1	Overcast	Dry	0	1	0
3	1	Clear	Dry	0	1	0
4	2	Raining	Wet	0	0	1

<Picture 3-3. Address type>

3-4. Weather

Remove non values, convert categorical variables into quantitative variables.

```
In [9]:  df['WEATHER'].unique()

Out[9]:  array(['Overcast', 'Raining', 'Clear', nan, 'Unknown', 'Other', 'Snowing',
                'Fog/Smog/Smoke', 'Sleet/Hail/Freezing Rain', 'Blowing Sand/Dirt',
                'Severe Crosswind', 'Partly Cloudy'], dtype=object)

In [10]: df.dropna(axis=0, inplace=True)
         indexname = df[df['WEATHER']=='Unknown'].index
         df.drop(indexname, inplace=True)
         indexname2 = df[df['WEATHER']=='Other'].index
         df.drop(indexname2, inplace=True)
         df['WEATHER'].unique()

Out[10]: array(['Overcast', 'Raining', 'Clear', 'Snowing', 'Fog/Smog/Smoke',
                'Sleet/Hail/Freezing Rain', 'Blowing Sand/Dirt',
                'Severe Crosswind', 'Partly Cloudy'], dtype=object)

In [11]: df=pd.concat([df, pd.get_dummies(df['WEATHER'])], axis=1)
         df.drop(['WEATHER'], axis=1, inplace=True)
         df.head()

Out[11]:
```

	SEVERITYCODE	ROADCOND	Alley	Block	Intersection	Blowing Sand/Dirt	Clear	Fog/Smog/Smoke	Overcast	Partly Cloudy	Raini
0	2	Wet	0	0	1	0	0	0	1	0	0
1	1	Wet	0	1	0	0	0	0	0	0	1
2	1	Dry	0	1	0	0	0	0	1	0	0
3	1	Dry	0	1	0	0	1	0	0	0	0
4	2	Wet	0	0	1	0	0	0	0	0	1

<Picture 3-4. Weather>

3-5. Road condition

Remove non values, convert categorical variables into quantitative variables.

In [12]:
```
df['ROADCOND'].unique()
```
Out[12]:
```
array(['Wet', 'Dry', 'Unknown', 'Snow/Slush', 'Ice', 'Other',
       'Sand/Mud/Dirt', 'Standing Water', 'Oil'], dtype=object)
```
In [13]:
```
indexname = df[df['ROADCOND']== 'Unknown'].index
df.drop(indexname, inplace=True)
indexname2 = df[df['ROADCOND']== 'Other'].index
df.drop(indexname2, inplace = True)
df['ROADCOND'].unique()
```
Out[13]:
```
array(['Wet', 'Dry', 'Snow/Slush', 'Ice', 'Sand/Mud/Dirt',
       'Standing Water', 'Oil'], dtype=object)
```
In [14]:
```
df=pd.concat([df, pd.get_dummies(df['ROADCOND'])],axis=1)
df.drop(['ROADCOND'], axis=1, inplace=True)
df.head()
```
Out[14]:

	SEVERITYCODE	Alley	Block	Intersection	Blowing Sand/Dirt	Clear	Fog/Smog/Smoke	Overcast	Partly Cloudy	Raining	Severe Crosswin
0	2	0	0	1	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	0	1	0
2	1	0	1	0	0	0	0	1	0	0	0
3	1	0	1	0	0	1	0	0	0	0	0
4	2	0	0	1	0	0	0	0	0	1	0

<Picture 3-5. Road condition>

3-6. Feature selection : x and y data

Set the attribute data(feature data) as X, and set the target data(severity code) as y.

In [15]:
```
x = pd.DataFrame(df)
x.drop(['SEVERITYCODE'], axis=1, inplace=True)
x[0:5]
```

Out[15]:

	Alley	Block	Intersection	Blowing Sand/Dirt	Clear	Fog/Smog/Smoke	Overcast	Partly Cloudy	Raining	Severe Crosswind	Sleet/Hail/Freez Rain
0	0	0	1	0	0	0	1	0	0	0	0
1	0	1	0	0	0	0	0	0	1	0	0
2	0	1	0	0	0	0	1	0	0	0	0
3	0	1	0	0	1	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	1	0	0

In [16]:
```
y = df['SEVERITYCODE'].values
y[0:5]
```

Out[16]: `array([2, 1, 1, 1, 2])`

<Picture 3-6. Feature selection x and y data>

3-7. Normalize data

Collecting Normal data are shown below.

```
In [18]: x= preprocessing.StandardScaler().fit(x).transform(x)
         x[0:5]

/opt/conda/envs/Python36/lib/python3.6/site-packages/sklearn/preprocessing/data.py:645: DataConversionWarning: Data
with input dtype uint8 were all converted to float64 by StandardScaler.
  return self.partial_fit(X, y)
/opt/conda/envs/Python36/lib/python3.6/site-packages/ipykernel/__main__.py:1: DataConversionWarning: Data with inpu
t dtype uint8 were all converted to float64 by StandardScaler.
  if __name__ == '__main__':

Out[18]: array([[-0.0598465 , -1.3187756 ,  1.34155615, -0.01735958, -1.33456065,
                 -0.05705211,  2.30601482, -0.00538224, -0.48704473, -0.01203576,
                 -0.02525264, -0.07231742, -1.58365616, -0.08056091, -0.01864761,
                 -0.01940933, -0.07247915, -0.02502187,  1.6358207 ],
                [-0.0598465 ,  0.75827912, -0.74540301, -0.01735958, -1.33456065,
                 -0.05705211, -0.43364856, -0.00538224,  2.05319951, -0.01203576,
                 -0.02525264, -0.07231742, -1.58365616, -0.08056091, -0.01864761,
                 -0.01940933, -0.07247915, -0.02502187,  1.6358207 ],
                [-0.0598465 ,  0.75827912, -0.74540301, -0.01735958, -1.33456065,
                 -0.05705211,  2.30601482, -0.00538224, -0.48704473, -0.01203576,
                 -0.02525264, -0.07231742,  0.6314502 , -0.08056091, -0.01864761,
                 -0.01940933, -0.07247915, -0.02502187, -0.61131394],
                [-0.0598465 ,  0.75827912, -0.74540301, -0.01735958,  0.74931027,
                 -0.05705211, -0.43364856, -0.00538224, -0.48704473, -0.01203576,
                 -0.02525264, -0.07231742,  0.6314502 , -0.08056091, -0.01864761,
                 -0.01940933, -0.07247915, -0.02502187, -0.61131394],
                [-0.0598465 , -1.3187756 ,  1.34155615, -0.01735958, -1.33456065,
                 -0.05705211, -0.43364856, -0.00538224,  2.05319951, -0.01203576,
                 -0.02525264, -0.07231742, -1.58365616, -0.08056091, -0.01864761,
                 -0.01940933, -0.07247915, -0.02502187,  1.6358207 ]])
```

<Picture 3-7. Normalize data>

IV. Conclusion

This report, show that method to prevent car accidents. Firstly, the climate and road conditions have lots of influence on car accident. Secondly, making up the dependent and independent variables, the purpose of collecting the data, dependent variables such as car accidents and independent variables such as climate and road conditions were set and so on. Third, we have established a range of coordinates to implement the methodology through big data.

As a result, it is recognized that the cause of car accidents are highly correlated with climate and road conditions. This requires a matrix of accurate statistical data on climate and data on road conditions in each region. In the future, it will be implemented as a system that can reduce car accidents by presenting results through raw data and preannouncement through machining learning.