Capstone Project - Car accident severity

Ruilin Wang

August 24, 2020

1. Introduction: Business Problem

Say you are driving to another city for work or to visit some friends. It is rainy and windy, and on the way, you come across a terrible traffic jam on the other side of the highway. Long lines of cars barely moving. As you keep driving, police car start appearing from afar shutting down the highway. Oh, it is an accident and there's a helicopter transporting the ones involved in the crash to the nearest hospital. They must be in critical condition for all of this to be happening. Now, wouldn't it be great if there is something in place that could warn you, given the weather and the road conditions about the possibility of you getting into a car accident and how severe it would be, so that you would drive more carefully or even change your travel if you are able to.

2. Data benefit

By looking at variables such as :

- road considion
- light condition
- weather
- speend

- Lane

we can build a model to see if the variables abovementioned will affect severity of traffic accident.

This model can classify the severity of the accident which would provide the driver with the **'worst-case scenario'**, rather than a probabilistic estimate of an accident occuring. This will help in inducing an appropriate level of cautiousness in the driver!

Features used are speeding and road condition for this model.

2.1 Data Cleaning

Data Cleaning

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report
from sklearn.tree import DecisionTreeClassifier
from sklearn import svm
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
from sklearn import preprocessing
from sklearn.linear_model import LogisticRegression
```

In [2]:
```
# Lets Download the Dataset
!wget -O Data-Collisions https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-d
```

```
--2020-08-24 17:32:22--  https://s3.us.cloud-object-storage.appdomain.cloud/cf-courses-data/CognitiveClass/DP0701EN/version-2/Data-Collisions.csv
Resolving s3.us.cloud-object-storage.appdomain.cloud (s3.us.cloud-object-storage.appdomain.cloud)... 67.228.254.196
Connecting to s3.us.cloud-object-storage.appdomain.cloud (s3.us.cloud-object-storage.appdomain.cloud)|67.228.254.196|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 73917638 (70M) [text/csv]
Saving to: 'Data-Collisions'

Data-Collisions     100%[===================>]  70.49M  27.3KB/s    in 36m 22s

2020-08-24 18:08:47 (33.1 KB/s) - 'Data-Collisions' saved [73917638/73917638]
```

In [3]:
```
# Load Data From CSV File
df = pd.read_csv('Data-Collisions')
df.head()
```

```
/opt/anaconda3/lib/python3.7/site-packages/IPython/core/interactiveshell.py:3072: DtypeWarning: Columns (33) have mixed types.Specify dtype option on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

Out[3]:

	SEVERITYCODE	X	Y	OBJECTID	INCKEY	COLDETKEY	REPORTNO	STATU
0	2	-122.323148	47.703140	1	1307	1307	3502005	Matche
1	1	-122.347294	47.647172	2	52200	52200	2607959	Matche
2	1	-122.334540	47.607871	3	26700	26700	1482393	Matche
3	1	-122.334803	47.604803	4	1144	1144	3503937	Matche
4	2	-122.306426	47.545739	5	17700	17700	1807429	Matche

5 rows × 38 columns

In [4]:
```
df.dtypes
```

Out[4]:
```
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
INCDATE            object
INCDTTM            object
JUNCTIONTYPE       object
SDOT_COLCODE        int64
SDOT_COLDESC       object
INATTENTIONIND     object
UNDERINFL          object
WEATHER            object
ROADCOND           object
LIGHTCOND          object
PEDROWNOTGRNT      object
SDOTCOLNUM        float64
SPEEDING           object
ST_COLCODE         object
ST_COLDESC         object
SEGLANEKEY          int64
CROSSWALKKEY        int64
HITPARKEDCAR       object
dtype: object
```

2.2 Assumptions for missing data

1. Drivers are not speeding
2. Road condition is unknow

In [5]:
```
# Replacing 'nan' with 'N' for speeding
df['SPEEDING'] = df['SPEEDING'].fillna('N')

# Replacing 'nan' witih 'Unknown' fpr road condition
df['ROADCOND'] = df['ROADCOND'].fillna('Unknown')
```

In [6]:
```
# To get the Dimensions of the Data
df.shape
```

Out[6]: (194673, 38)

2.3 Assumptions for data analysis

We assume that all road condition are desired only for Good and bad

In [7]:
```
# Replacing ROADCOND values:
df['ROADCOND'].replace(to_replace=['Wet','Dry','Unknown','Snow/Slush','Ice','Other','Sand/Mu
```

In [8]:
```
# Changing the data into numerical values...
df["SPEEDING"].replace(to_replace=['N', 'Y'], value=[0,1], inplace=True)
df['ROADCOND'].replace(to_replace=['Good','Bad'],value=[0,1],inplace=True)
# Defining dataset
test_condition = df[['SPEEDING','ROADCOND']]
test_condition.head()
```

Out[8]:

	SPEEDING	ROADCOND
0	0	1
1	0	1
2	0	0
3	0	0
4	0	1

2.4 Data Analysis

The proportion of L2 severity is higher when the driver speed:

In [9]:
```
speed_analysis = df.groupby(['SPEEDING'])['SEVERITYCODE'].value_counts(normalize=True)
speed_analysis
```

Out[9]:

SPEEDING	SEVERITYCODE	
0	1	0.705099
	2	0.294901
1	1	0.621665
	2	0.378335

Name: SEVERITYCODE, dtype: float64

The proportion of L2 severity is higher when the road condition is bad:

In [10]:
```
road_analysis = df.groupby(['ROADCOND'])['SEVERITYCODE'].value_counts(normalize=True)
road_analysis
```

Out[10]:

ROADCOND	SEVERITYCODE	
0	1	0.710389
	2	0.289611
1	1	0.674176
	2	0.325824

Name: SEVERITYCODE, dtype: float64

This means that these features do have an effect on the severity of accidents when it happens..

3. Methodology

Metrics:

In [11]:
```
x = test_condition
y = df['SEVERITYCODE'].values.astype(str)
x = preprocessing.StandardScaler().fit(x).transform(x)
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=1234)

# verifying set dimensions
print("Training set: ", x_train.shape, y_train.shape)
print("Testing set: ", x_test.shape, y_test.shape)
```

```
Training set:  (155738, 2) (155738,)
Testing set:  (38935, 2) (38935,)
```

Method I

In [12]:
```
KNN_model = KNeighborsClassifier(n_neighbors = 4).fit(x_train, y_train)
predicted = KNN_model.predict(x_test)
KNN_f1 = f1_score(y_test, predicted, average='weighted')
KNN_acc = accuracy_score(y_test, predicted)
```

Method II

In [13]:
```
Tree_model = DecisionTreeClassifier(criterion="entropy", max_depth = 4)
Tree_model.fit(x_train, y_train)
predicted = Tree_model.predict(x_test)
Tree_f1 = f1_score(y_test, predicted, average='weighted')
Tree_acc = accuracy_score(y_test, predicted)
```

Method III

In [14]:
```
LR_model = LogisticRegression(C=0.01, solver='liblinear').fit(x_train, y_train)
predicted = LR_model.predict(x_test)

LR_f1 = f1_score(y_test, predicted, average='weighted')
LR_acc = accuracy_score(y_test, predicted)
```

4. Results

In [15]:
```
table = {
    "Algorithm": ["KNN", "Decision Tree", "LogisticRegression"],
    "F1-score": [KNN_f1, Tree_f1, LR_f1],
    "Accuracy": [KNN_acc, Tree_acc, LR_acc]
}

table = pd.DataFrame(table)
table
```

Out[15]:

	Algorithm	F1-score	Accuracy
0	KNN	0.591378	0.696751
1	Decision Tree	0.576051	0.699679
2	LogisticRegression	0.576051	0.699679

I will choose LR model.

In [16]:
```
table = {
    "Intercept": LR_model.intercept_,
    "Coef:SPEEDING ": LR_model.coef_[:,0],
    "Coef:ROADCOND ": LR_model.coef_[:,1],
}

table = pd.DataFrame(table)
table
```

Out[16]:

	Intercept	Coef:SPEEDING	Coef:ROADCOND
0	-0.853729	0.067702	0.068295

As the coefficients are positive, I conclude that the 2 conditions (speeding and road conditions) have an effect of increasing accident severity.

5. Conclusions

This model provide empirical evidence against speeding and road conditions