Predicting the Severity of Car accidents

Mohammad Abdullah Tahir

September 18, 2018

Contents

1.Introduction

1.1 Background:

Seattle, also known as the Emerald city, is Washington State's largest city, with home to a large tech industry with Microsoft and Amazon headquartered in its metropolitan area. As of 2020, it has a total metro area population of 3.4 million. The total number of personal vehicles in Seattle in the year 2016 hit a new high of nearly 444,000 vehicles. In one South Lake Union census tract, the car population has more than doubled since 2010. The increase in car ownership rates can lead to higher numbers of accidents on the road because of a simple probability. Worldwide, approximately 1.35 million people die in road crashes each year, on average 3,700 people lose their lives every day on the roads and an additional 4.4 million suffer non-fatal injuries, often resulting in long-term disabilities.

1.2 Problem

As we all know that thousands of people lost their lives or become disable every year due to some accidents. According to USA National high authority safety administration, every year car accident costs up to $850 billion. It's a huge amount and the rate the death in car accidents is also very high i.e. 3800 people per year. In addition to this 4.4 million people are seriously injured. The project aims to predict how the severity of accidents can be based on various factors like the condition of the road, condition of light of that area at the time of the accident, and many other factors.

1.3 Stakeholders

If the severity of car accidents reduces then this must be very beneficial for the Seattle Public Development Authority because we have already seen that car accidents are very expensive also. If this cost reduces then Public development authority will use that expense in the development of the city and launch many other development plans.

2 Data

2.1Understanding the data:

There are lots of problems in the dataset which we are going to use in this problem. There are total 194672 rows/observations and 37 columns/features

namely Severity Code, X(X co-ordinate of location), Y(Y co-ordinate of location), LIGHTCOND and many other you can check it [here](#).

```
df.columns

Index(['SEVERITYCODE', 'X', 'Y', 'OBJECTID', 'INCKEY', 'COLDETKEY', 'REPORTNO',
       'STATUS', 'ADDRTYPE', 'INTKEY', 'LOCATION', 'EXCEPTRSNCODE',
       'EXCEPTRSNDESC', 'SEVERITYCODE.1', 'SEVERITYDESC', 'COLLISIONTYPE',
       'PERSONCOUNT', 'PEDCOUNT', 'PEDCYLCOUNT', 'VEHCOUNT', 'INCDATE',
       'INCDTTM', 'JUNCTIONTYPE', 'SDOT_COLCODE', 'SDOT_COLDESC',
       'INATTENTIONIND', 'UNDERINFL', 'WEATHER', 'ROADCOND', 'LIGHTCOND',
       'PEDROWNOTGRNT', 'SDOTCOLNUM', 'SPEEDING', 'ST_COLCODE', 'ST_COLDESC',
       'SEGLANEKEY', 'CROSSWALKKEY', 'HITPARKEDCAR'],
      dtype='object')

df.shape

(194673, 38)
```

This dataset has very high variation for our machine learning problem which to categories the severity of accidents for which we are going to use machine learning algorithms like "Logistic Regression", "Support Vector Machine", "Decision Tree" or any other. As in all of these algorithms use numerical values as an intake to predict the result.So we have to clean the data.

2.2 Cleaning the data:

First we try to find how any values are empty in all columns.We can do that like in I do below:

```
missing_value =  df.isnull()
for column in missing_value.columns:
    print(missing_value[column].value_counts())

False    194673
Name: SEVERITYCODE, dtype: int64
False    189339
True       5334
Name: X, dtype: int64
False    189339
True       5334
Name: Y, dtype: int64
False    194673
Name: OBJECTID, dtype: int64
False    194673
Name: INCKEY, dtype: int64
False    194673
Name: COLDETKEY, dtype: int64
False    194673
Name: REPORTNO, dtype: int64
False    194673
Name: STATUS, dtype: int64
False    192747
True       1926
Name: ADDRTYPE, dtype: int64
True     129603
False     65070
Name: INTKEY, dtype: int64
False    191996
True       2677
Name: LOCATION, dtype: int64
True     109862
False     84811
Name: EXCEPTRSNCODE, dtype: int64
True     189035
False      5638
```

There are many values which are empty so we try get the values for that or try to replace the empty values with average of the all values in that column and we also try to fill all the empty with the most occurring value for the columns like "WEATHER", "LIGHTCOND" and for columns like "UNERINFL" we convert all "No" to 0 and " YES" to 1.

All the above change are given below in codes:

```python
df1['ADDRTYPE'].fillna(value =df1['ADDRTYPE'].value_counts().idxmax(),inplace  = True)

df1['JUNCTIONTYPE'].fillna(value =df1['JUNCTIONTYPE'].value_counts().idxmax(),inplace  = True)

df1['UNDERINFL'].replace({'N':0,'Y':1}, inplace = True)

df1['WEATHER'].fillna(value =df1['WEATHER'].value_counts().idxmax(),inplace  = True)

df1['ROADCOND'].fillna(value =df1['ROADCOND'].value_counts().idxmax(),inplace  = True)

df1['UNDERINFL'].fillna(value = 0, inplace = True)
```

Next we drop some columns like "Reportno", "Lcation" because these columns are not for our use.

```python
df1 = df.drop(columns = ['LIGHTCOND','LOCATION','INCDATE','INCDTTM','SDOT_COLCODE','SDOT_COLDESC',
                         'SEVERITYDESC',
                         'SDOTCOLNUM','ST_COLCODE'
                        ,'ST_COLDESC','EXCEPTRSNCODE','EXCEPTRSNDESC','STATUS','CROSSWALKKEY',
                         'COLLISIONTYPE','COLDETKEY','REPORTNO'])
```
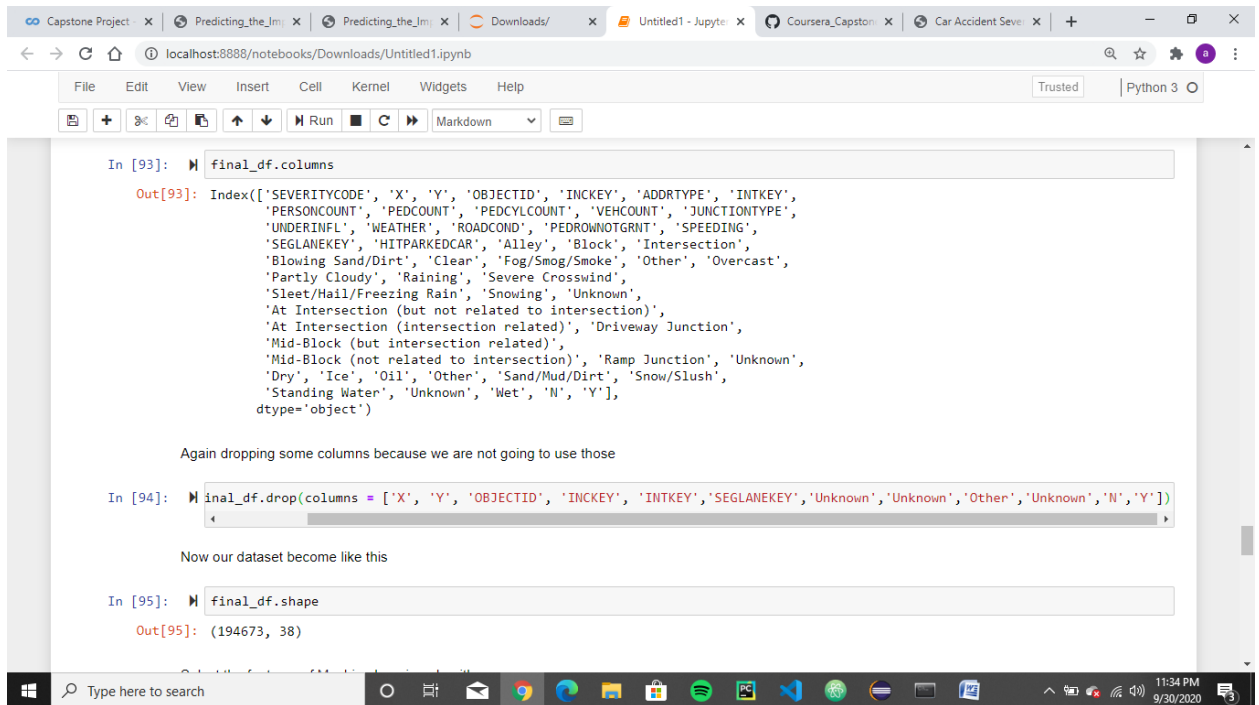
Now we just want to convert the columns like "ADDRTYPE" because column like these are categorical data with not numerical values. So we have convert all the columns like this one to get numerical values.
We can do that like given below:

```python
ADDRTYPE = pd.get_dummies(df1['ADDRTYPE'])

WEATHER = pd.get_dummies(df1['WEATHER'])

JUNCTIONTYPE = pd.get_dummies(df1['JUNCTIONTYPE'])

ROADCOND = pd.get_dummies(df1['ROADCOND'])

ROADCOND = pd.get_dummies(df1['ROADCOND'])

HITPARKEDCAR  = pd.get_dummies(df1['HITPARKEDCAR'])
```

Now, we get our final dataframe which we are going to use in this problem Which look like this:

2.3 Feature Selection

Now select the features for Machine learning models

3. Methodology
3.1 Data Collection

The dataset used for this project is based on car accidents which have taken place within the city of Seattle, Washington from the year 2004 to 2020. This data is regarding car accidents the

severity of each car accidents along with the time and conditions under which each accident occurred. The data set used for this project can be found [here](#).

3.2 EDA

Considering that the feature set and the target variable are categorical variables with the likes of weather, road condition and light condition being an above level 2 categorical variables whose values are limited and usually based on a particular finite group whose correlation might depict a different image then what it actually is. Generally, considering the effect of these variables in car accidents are important hence these variables were selected. A few pictorial depictions of the dataset were made in order to better understand the data.

```
sns.countplot(y='COLLISIONTYPE', data = df)
```
```
<matplotlib.axes._subplots.AxesSubplot at 0x2283fa60f88>
```
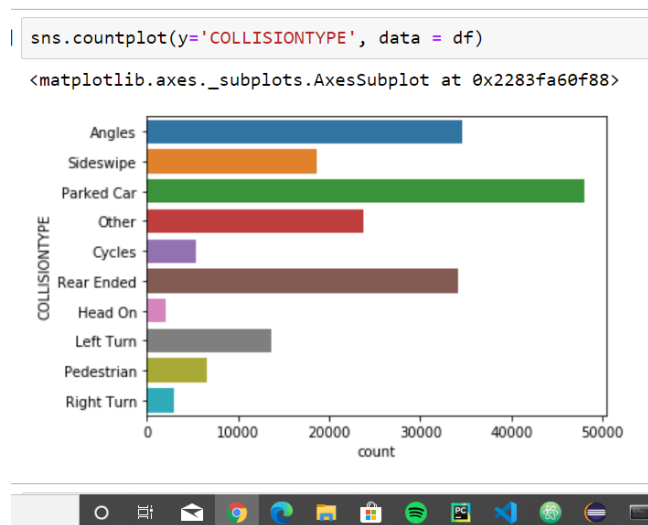

The above figure tell us that most of the cars in accident are hit the parked cars

```
sns.countplot(y = 'SEVERITYCODE', data =df)
```
```
<matplotlib.axes._subplots.AxesSubplot at 0x228532a4e88>
```
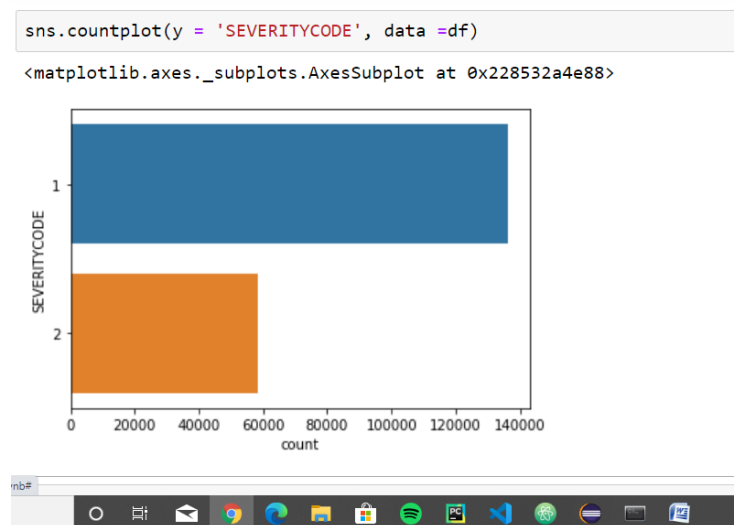
The above figure illustrates, after data cleaning has taken place, the distribution of the target variables between Physical Injury and Property Damage Only. As it can be seen that the dataset is supervised but an unbalanced dataset where the distribution of the target variable is in almost 1:2 ratio in favor of property damage. It is very important to have a balanced dataset when using machine learning algorithms.

3.3 Machine learning model selection

The machine learning models used are Logistic Regression, and Decision Tree Analysis. Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable. The Decision Tree Analysis breaks down a data set into smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. The reason why Decision Tree Analysis, and Logistic Regression were chosen is because the Support Vector Machine (SVM) model is inaccurate for large data sets, while this data set has more than 190,000 rows filled with data. Furthermore, SVM works best with dataset filled with text and images.

4. Result

Decision Tree

Decision Tree Classifier from the scikit-learn library was used to run the Decision Tree Classification model on the Car Accident Severity data. The criterion chosen for the classifier was 'entropy' and the max depth was '5'. The balanced data was used to predict and fit the Decision Tree Classifier.

```
from sklearn.tree import DecisionTreeClassifier
```

```
= DecisionTreeClassifier(criterion='entropy',splitter='random',max_depth=5)
```

```
tree.fit(X_train,y_train)
```

```
]: DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=5,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False,
            random_state=None, splitter='random')
```

Test the train model

```
prediction = tree.predict(X_test)
```

4.1 Classification Report

```
print(classification_report(y_test,prediction))
```

	precision	recall	f1-score	support
1	0.75	0.99	0.85	45306
2	0.88	0.19	0.31	18937
accuracy			0.75	64243
macro avg	0.81	0.59	0.58	64243
weighted avg	0.79	0.75	0.69	64243

Logistic Regression

Logistic Regression from the scikit-learn library was used to run the Logistic Regression model on the Car Accident Severity data. The balanced data was used to predict and fit the model.

Import LogisticRegression and make instance of it and then train the model

```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression(solver = 'liblinear')

lr.fit(X_train, y_train)
```
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='liblinear', tol=0.0001, verbose=0,
                   warm_start=False)
```

Test the trained model

```
pred =  lr.predict(X_test)
```

we get aroung 75% accuracy for the trained logistic classifier

4.2 Classification Report

```
In [202]: print(classification_report(y_test,pred))
```

	precision	recall	f1-score	support
1	0.75	0.98	0.85	45306
2	0.81	0.22	0.35	18937
accuracy			0.76	64243
macro avg	0.78	0.60	0.60	64243
weighted avg	0.77	0.76	0.70	64243

5.Conclusion

When comparing all the models by their f1-scores, Precision and Recall, we can have a clearer picture in terms of the accuracy of the two models individually as a whole and how well they perform Car Accident Severity – Seattle, Washington 14 for each output of the target variable. When comparing these scores, we can see that the f1-score is approximately equal at 0.85 When looking at the other two models, we can see that the Decision Tree has a more balanced precision and recall also for 0 and 1. Furthermore, the average f1-score of the two models are very close but for the Logistic Regression it is higher by 0.04. It can be concluded that the both the models can be used side by side for the best performance. Both the model given approximately same accuracy so we can you any of it according to our convenience.

6. Recommendation

After assessing the data and the output of the Machine Learning models, a few recommendations can be made for the stakeholders. The developmental body for Seattle city can assess how much of these accidents have occurred in a place where road or light conditions were not ideal for that specific area and could launch development projects for those areas where most severe accidents take place in order to minimize the effects of these two factors. Whereas, the car drivers could also use Car Accident Severity – Seattle, Washington 15 this data to assess when to take extra precautions on the road under the given circumstances of light condition, road condition and weather, in order to avoid a severe accident, if any