SEATTLE ACCIDENT COLISSION ANALYSIS

Coursera Capstone Project – Sep 2020

By: Adeel Naim Khan

INTRODUCTION / BUSINESS UNDERSTANDING

- Predicting an accident collision based on the data collected is always challenging aspect of the data science topology. In our daily routine commuting, it is always advisable to take the route which has less blockage or rush due to accidents. Our main purpose here is to build a model which can predict a model to assist the daily commuters avoiding these issues.
- There are many factors which causes the accidents and to predict the pattern using these factors can be very helpful to avoid such incidents. Imagine an accident occurring at a particular place or time many times and by analysis we find the relation of the occurrence of accident with factors which can be avoided or warned against.
- This will be highly beneficial for authorities to take necessary steps and implement a plan for drivers to avoid it from happening in future.

Map pic by Chris Daniels

DATA UNDERSTANDING

- Data Source: http://data-seattlecitygis.opendata.arcgis.com/datasets/5b5c745e0f1f48e7a53acec63a0022ab_0.csv?outSR={%22latestWkid%22:2926,%22wkid%22:2926}

- Total number of 40 variables available with 221389 data values.

- Main aim for the project is to predict the SEVERITY of accidents in Seattle, based on the data provided it was considered as dependent variable.

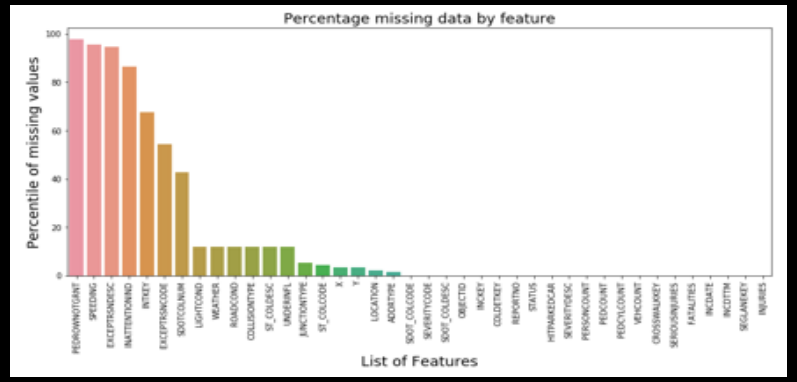
- Different attributes like Weather, Road conditions, Light conditions, physical coordinates, Types of junctions are considered as independent variables which helped to predict the Severity of the accident.

Out[20]:

	X	Y	OBJECTID	INCKEY	COLDETKEY	REPORTNO	STATUS	ADDRTYPE	INTKEY	LOCATION	...	RO
count	213918.000000	213918.000000	221389.00000	221389.000000	221389.000000	221389	221389	217677	71884.000000	216801	...	195
unique	NaN	NaN	NaN	NaN	NaN	221386	2	3	NaN	25198	...	9
top	NaN	NaN	NaN	NaN	NaN	1780512	Matched	Block	NaN	BATTERY ST TUNNEL NB BETWEEN ALASKAN WY VI NB ...	...	Dry
freq	NaN	NaN	NaN	NaN	NaN	2	195232	144917	NaN	298	...	128
mean	-122.330756	47.620199	110695.00000	144708.701914	144936.934541	NaN	NaN	NaN	37612.330964	NaN	...	Nal
std	0.030055	0.056043	63909.64371	89126.729589	89501.312920	NaN	NaN	NaN	51886.084219	NaN	...	Nal
min	-122.419091	47.495573	1.00000	1001.000000	1001.000000	NaN	NaN	NaN	23807.000000	NaN	...	Nal
25%	-122.349280	47.577151	55348.00000	71634.000000	71634.000000	NaN	NaN	NaN	28652.750000	NaN	...	Nal
50%	-122.330363	47.616053	110695.00000	127184.000000	127184.000000	NaN	NaN	NaN	29973.000000	NaN	...	Nal
75%	-122.311998	47.664290	166042.00000	209783.000000	210003.000000	NaN	NaN	NaN	33984.000000	NaN	...	Nal
max	-122.238949	47.734142	221389.00000	333843.000000	335343.000000	NaN	NaN	NaN	757580.000000	NaN	...	Nal

DATA PREPARATION / ANALYSIS

- Many columns have high number of missing values.
- Columns with missing percentile greater than 40 were immediately removed



- Other columns which had less than 15% of missing data were either removed (based on their no direct relationship with Severity Code) or their missing values were replaced with the Top value of the attribute.

We will drop the columns like Location, ObjectID, Report no, Status and different keys because they have NO IMPACT on our target variable SEVERITYCODE.

```
In [6]: DataSetCol.drop(['LOCATION','OBJECTID','INCKEY','COLDETKEY','REPORTNO','STATUS','ST_COLCODE','ST_COLDESC', 'SEGLANEKEY', 'CROSSWALKKEY','SDOT_COLCODE','SDOT_COLDESC'],axis=1,inplace=True)
```

- Variable SEVERITYCODE was Normalized by replacing "**Unknown**" value category with "**Property Damage only Collision category**"

```
In [7]: DataSetCol['SEVERITYCODE'].replace("0",np.nan, inplace = True)
        DataSetCol['SEVERITYDESC'].replace("Unknown",np.nan, inplace = True)

In [8]: DataSetCol[['SEVERITYCODE','SEVERITYDESC']].isnull().sum()

Out[8]: SEVERITYCODE    21595
        SEVERITYDESC    21595
        dtype: int64

In [8]: Severity_unknown=DataSetCol[['SEVERITYDESC']].isnull().sum().sort_values(ascending=False)
        percent_unknown = (DataSetCol[['SEVERITYDESC']].isnull().sum()/DataSetCol[['SEVERITYDESC']].isnull().count()*100).sort_values(ascending=False)
        unknown_data_perc = pd.concat([Severity_unknown, percent_unknown], axis=1, keys=['Severity Unknown', 'Percent Unknown'])
        unknown_data_perc.head()
```

Out[8]:

	Severity Unknown	Percent Unknown
SEVERITYDESC	21616	9.757815

```
In [17]: DataSetCol['SEVERITYCODE'].describe(include='all')

Out[17]: count      199794
         unique          4
         top             1
         freq       137596
         Name: SEVERITYCODE, dtype: object
```

```
In [9]: DataSetCol['SEVERITYCODE'].replace(np.nan,"1", inplace = True)
        DataSetCol['SEVERITYDESC'].replace(np.nan,"Property Damage Only Collision", inplace = True)

In [19]: DataSetCol[['SEVERITYCODE','SEVERITYDESC']].isnull().sum()

Out[19]: SEVERITYCODE    0
         SEVERITYDESC    0
         dtype: int64

In [20]: DataSetCol['SEVERITYCODE'].value_counts()

Out[20]: 1     159191
         2      58747
         2b      3102
         3        349
         Name: SEVERITYCODE, dtype: int64

In [21]: DataSetCol.groupby(['SEVERITYCODE'])['SEVERITYDESC'].value_counts()

Out[21]: SEVERITYCODE  SEVERITYDESC
         1             Property Damage Only Collision    159191
         2             Injury Collision                   58747
         2b            Serious Injury Collision            3102
         3             Fatality Collision                   349
         Name: SEVERITYDESC, dtype: int64
```

MODEL TECHNIQUES

- Following 3 Models are utilized

➢ **KNN**

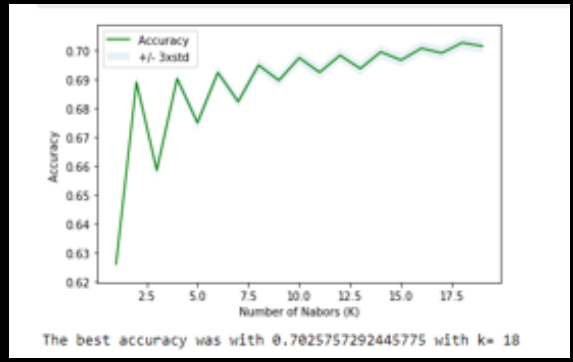
- ➢ The K-Nearest Neighbours algorithm is a classification algorithm that takes a bunch of labelled points and uses them to learn how to label other points.
- ➢ **A method of classifying cases, based on the similarity of other cases.** Cases that are near each other are said to be neighbours.

➢ **Decision Tree**

- ➢ The basic intuition behind a decision tree is to map out all possible decision paths in the form of a tree.
- ➢ Decision trees are about testing an attribute and branching the cases based on the result of the test.

➢ **Logistic Regression**

- ➢ It is a classification algorithm for categorical variables
- ➢ It is analogous to linear regression but tries to predict a categorical or discrete target field instead of numeric one.



PREDICTION RESULT

- Accuracy of the Model is attained through calculating the following Metrics

- **F1- Score**

- **Jaccard Score**

- **Log Loss**

Algorithm	Jaccard	F1-score	LogLoss
KNN	0.7046	0.637	NA
Decision Tree	0.7126	0.5993	NA
Logistic Regression	0.7133	0.6394	0.5566

CONCLUSION

- From the results of their accuracy models, we can see that all of them experienced accuracy of more than 70% which is a good indication based on the huge data we received.
- From Evaluation point of the view, we can see the Jaccard set Accuracy for all of them is also greater than 70%, which is also a good sign and proves the confidence on our working model. F1 Score for KNN is around 63% whereas for Decision Tree and Logistic Regression it is around 59%. We can consider this as acceptable.
- This model can help predict the severity of the accidents based on factors considered as Road condition, Light conditions, Weather conditions, Junction types and physical location of the accident area.