IBM DATA SCIENCE COURSE offered on COURSERA

CAPSTONE PROJECT FOR DATA SCIENCE:

PREDICTING THE SEVERITY OF A ROAD COLLISION USING HISTORICAL DATA FOR SEATTLE CITY

Author: RUDRA SHEKHAR

Date: 14th September 2020

Place: INDIA

ABSTRACT

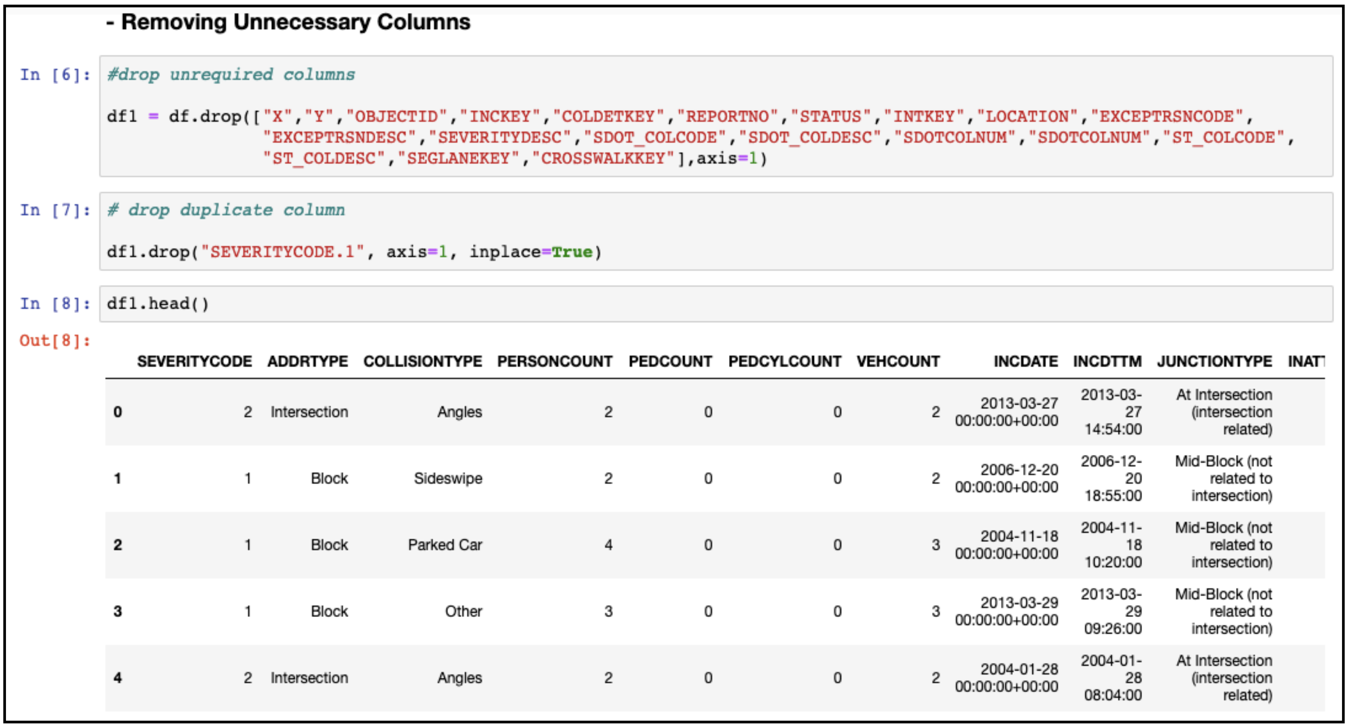
Accidents cause huge damage to life and property but these losses can be minimized by deploying proper strategies. Our aim was to study We analyzed historical accident data of Seattle using three different machine learning algorithms which are k-nearest neighbors, decision tree and logistic regression. We evaluated the accuracy of these models using Jaccard index, F1- score and Log loss as estimator. We find that decision tree can be deployed to predict the severity of road accident with accuracy of 0.59 that was calculated using Jaccard index.

INTRODUCTION

Road accidents are a common phenomenon in our daily lives. Accidents can occur due to external phenomenon which are outside the control of driver or due to factors associated with driving. External factors include weather conditions, road conditions or light conditions on the road. Besides it there are internal factors like over speeding, drug abuse, drink and drive, in attention which are associated with the driver. Studying severity of an accident based on these parameters give us meaningful insight about the situation in which an accident takes place. An accident also gives information about the location and the particular day of week on which accident is most probable. As a data analyst our aim is to study the severity of accident based on the above mentioned factors , after which, we can train a model for given set of conditions, where it can predict the severity of accident for future events. After knowing its severity and associated chances of fatality, we can minimize the damage caused both to human as well as the property. The model aims to address various stake holders which include individuals, government agency and the insurance industry. Individual can pay particular attention in situation where there are high chances of accident and mitigate the risk. Government can work in both direction, i.e., by framing laws where there is insincerity on part of citizens and on other side it can work on infrastructure by improvement road condition and street light facility. At last it an insurance provider can use it to improve the quality of its service delivery by knowing the location as it can provide nearest road side assistance and enhance its customer base by expanding its efficiency.

DATA

We have the data of road accident of Seattle city, which consists of 1,94,673 cases and have 37 attributes or columns for all these accidents. In our data, the target label is “SEVERITYCODE” where the code for severity of accident is 1 for property damage collision and 2 for Injury collision. We have 37 columns in our data set where many of the columns in our data set are not fit for analysis like how many pedestrians, bicycles, vehicles or individuals involved in accident. There are various columns which are purely for administrative records. We exclude these unnecessary data and then make a bar plot of possible attributes in our data set to see which attributes gives us insight about the road accident severity. Data, in its original form, has many attributes which are unnecessary. First we remove the unnecessary columns.



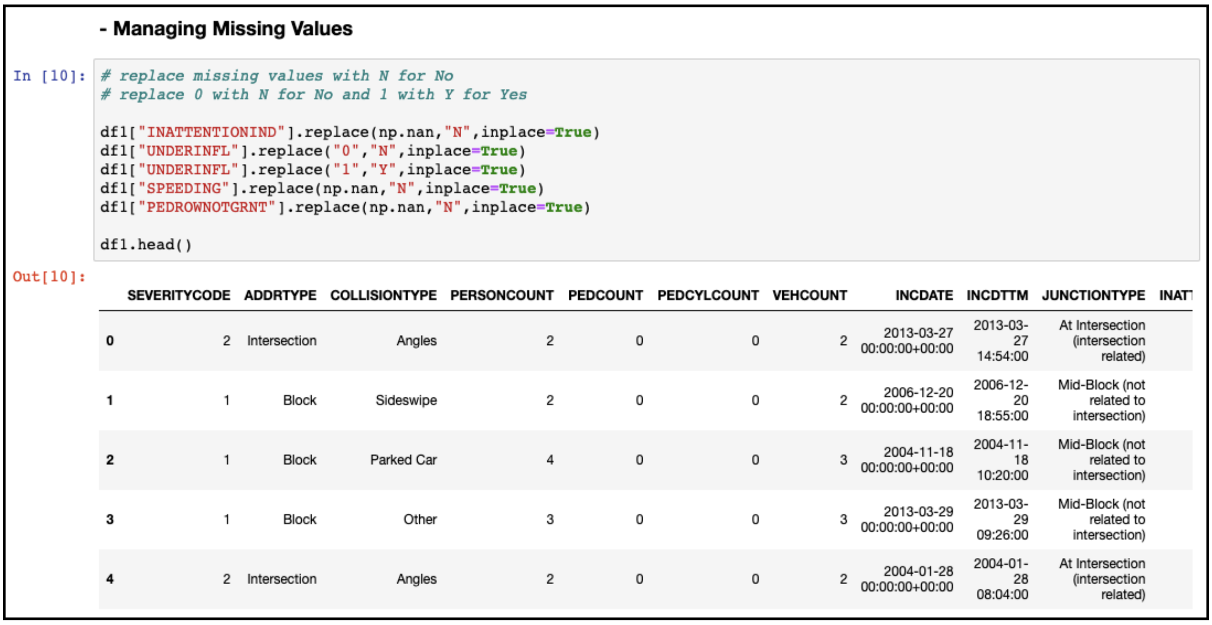
- **Removing Unnecessary Columns**

```
In [6]: #drop unrequired columns

        df1 = df.drop(["X","Y","OBJECTID","INCKEY","COLDETKEY","REPORTNO","STATUS","INTKEY","LOCATION","EXCEPTRSNCODE",
                       "EXCEPTRSNDESC","SEVERITYDESC","SDOT_COLCODE","SDOT_COLDESC","SDOTCOLNUM","SDOTCOLNUM","ST_COLCODE",
                       "ST_COLDESC","SEGLANEKEY","CROSSWALKKEY"],axis=1)

In [7]: # drop duplicate column

        df1.drop("SEVERITYCODE.1", axis=1, inplace=True)

In [8]: df1.head()

Out[8]:
```

	SEVERITYCODE	ADDRTYPE	COLLISIONTYPE	PERSONCOUNT	PEDCOUNT	PEDCYLCOUNT	VEHCOUNT	INCDATE	INCDTTM	JUNCTIONTYPE	INAT
0	2	Intersection	Angles	2	0	0	2	2013-03-27 00:00:00+00:00	2013-03-27 14:54:00	At Intersection (intersection related)	
1	1	Block	Sideswipe	2	0	0	2	2006-12-20 00:00:00+00:00	2006-12-20 18:55:00	Mid-Block (not related to intersection)	
2	1	Block	Parked Car	4	0	0	3	2004-11-18 00:00:00+00:00	2004-11-18 10:20:00	Mid-Block (not related to intersection)	
3	1	Block	Other	3	0	0	3	2013-03-29 00:00:00+00:00	2013-03-29 09:26:00	Mid-Block (not related to intersection)	
4	2	Intersection	Angles	2	0	0	2	2004-01-28 00:00:00+00:00	2004-01-28 08:04:00	At Intersection (intersection related)	

Figure 1

Original data had various missing values which must be replaced or removed. We either replaced them with mostly occurring variables of that column, or completed removed the corresponding rows from the data set.



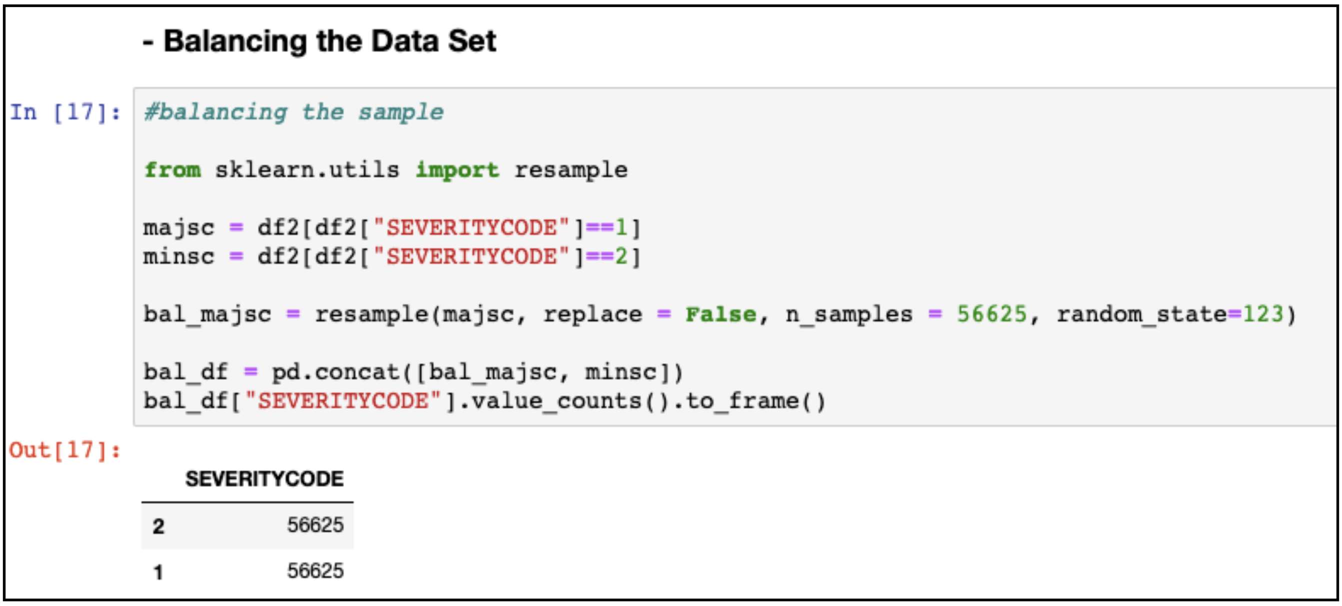
- **Managing Missing Values**

```
In [10]: # replace missing values with N for No
         # replace 0 with N for No and 1 with Y for Yes

         df1["INATTENTIONIND"].replace(np.nan,"N",inplace=True)
         df1["UNDERINFL"].replace("0","N",inplace=True)
         df1["UNDERINFL"].replace("1","Y",inplace=True)
         df1["SPEEDING"].replace(np.nan,"N",inplace=True)
         df1["PEDROWNOTGRNT"].replace(np.nan,"N",inplace=True)

         df1.head()

Out[10]:
```

	SEVERITYCODE	ADDRTYPE	COLLISIONTYPE	PERSONCOUNT	PEDCOUNT	PEDCYLCOUNT	VEHCOUNT	INCDATE	INCDTTM	JUNCTIONTYPE	INAT
0	2	Intersection	Angles	2	0	0	2	2013-03-27 00:00:00+00:00	2013-03-27 14:54:00	At Intersection (intersection related)	
1	1	Block	Sideswipe	2	0	0	2	2006-12-20 00:00:00+00:00	2006-12-20 18:55:00	Mid-Block (not related to intersection)	
2	1	Block	Parked Car	4	0	0	3	2004-11-18 00:00:00+00:00	2004-11-18 10:20:00	Mid-Block (not related to intersection)	
3	1	Block	Other	3	0	0	3	2013-03-29 00:00:00+00:00	2013-03-29 09:26:00	Mid-Block (not related to intersection)	
4	2	Intersection	Angles	2	0	0	2	2004-01-28 00:00:00+00:00	2004-01-28 08:04:00	At Intersection (intersection related)	

Figure 2

After plotting bar graphs we saw that data is highly unbalanced, where severity code in class 1 is almost three times the size of class 2. We balanced the data by down sampling it.



- Balancing the Data Set

```
In [17]: #balancing the sample

         from sklearn.utils import resample

         majsc = df2[df2["SEVERITYCODE"]==1]
         minsc = df2[df2["SEVERITYCODE"]==2]

         bal_majsc = resample(majsc, replace = False, n_samples = 56625, random_state=123)

         bal_df = pd.concat([bal_majsc, minsc])
         bal_df["SEVERITYCODE"].value_counts().to_frame()

Out[17]:
```

	SEVERITYCODE
2	56625
1	56625

Figure 3

We made a bar plot of attributes like speeding, under influence of alcohol, and inattention of driver to study the accidents due to errors on part of individual.

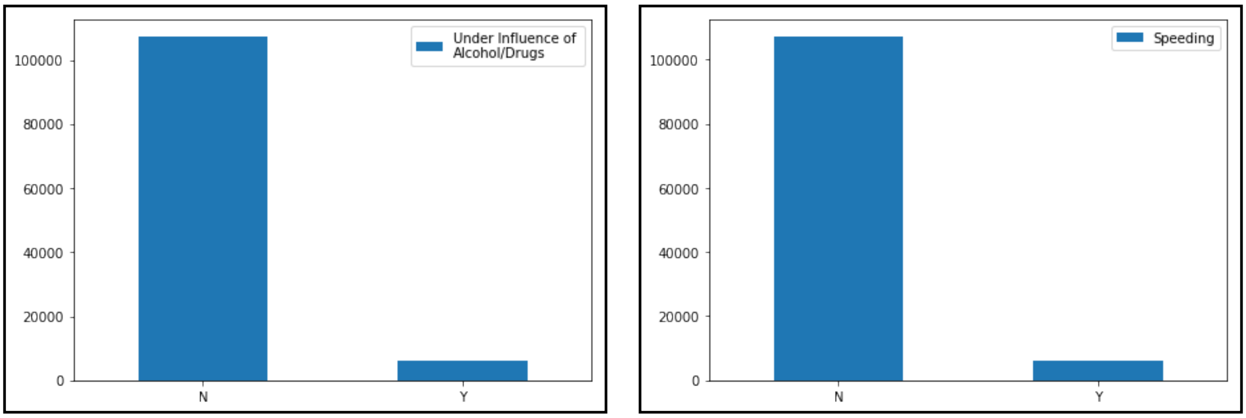


Figure 4

We also checked if the day of the week resembles with chances of accident and its severity. We noticed that working days have a higher frequency of collisions.

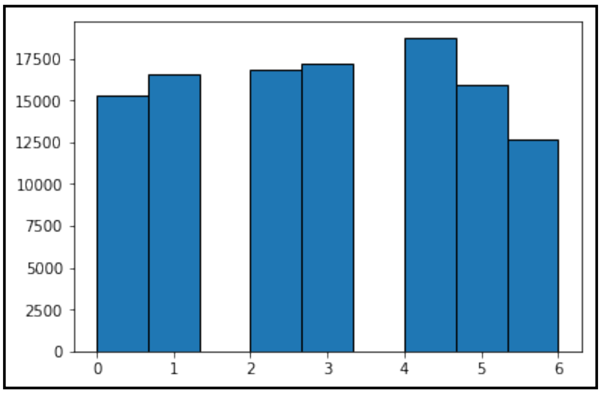
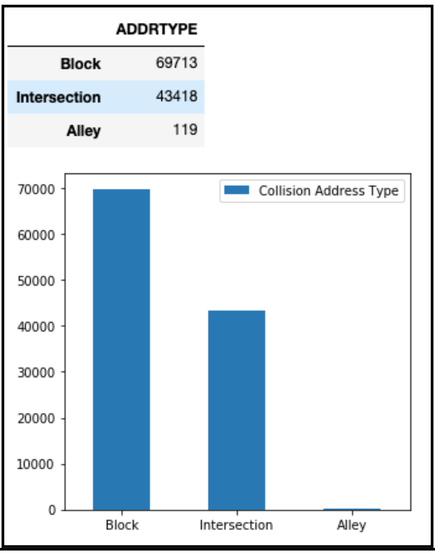
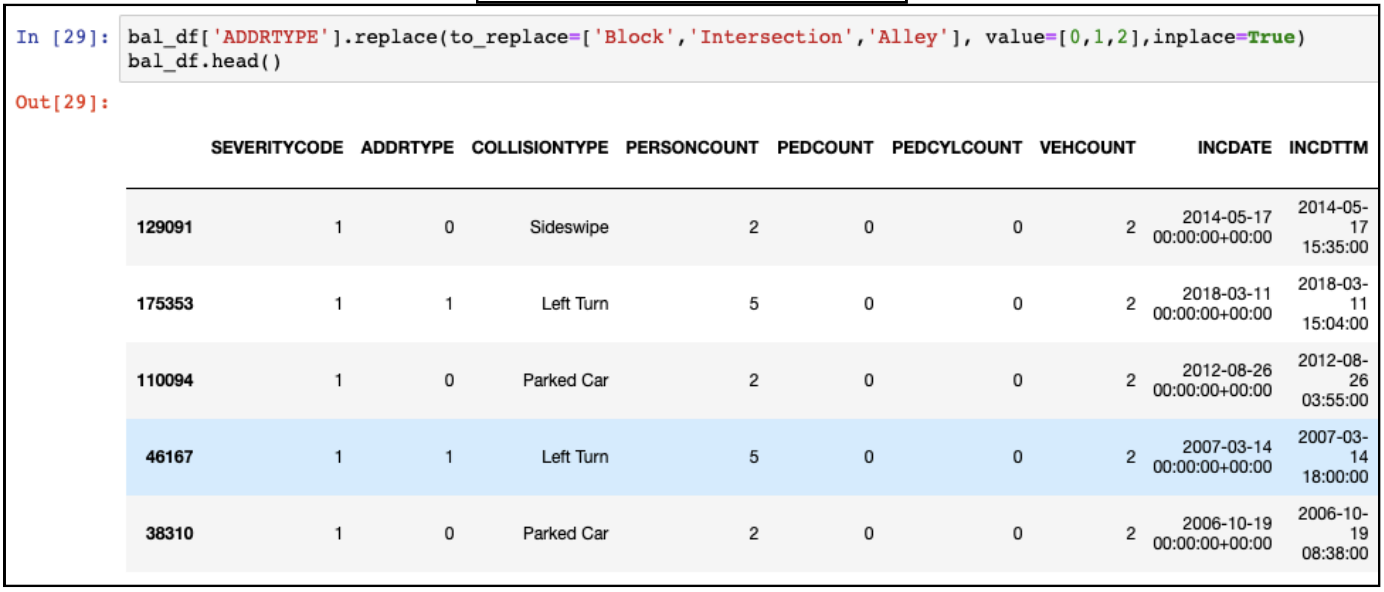


Figure 5

Attributes in our data set are mostly of string types so we used label encoding to make our data suitable for fitting with model. A sample of attribute "ADDRTYPE" is shown here.



	ADDRTYPE
Block	69713
Intersection	43418
Alley	119



```
In [29]: bal_df['ADDRTYPE'].replace(to_replace=['Block','Intersection','Alley'], value=[0,1,2],inplace=True)
         bal_df.head()
```

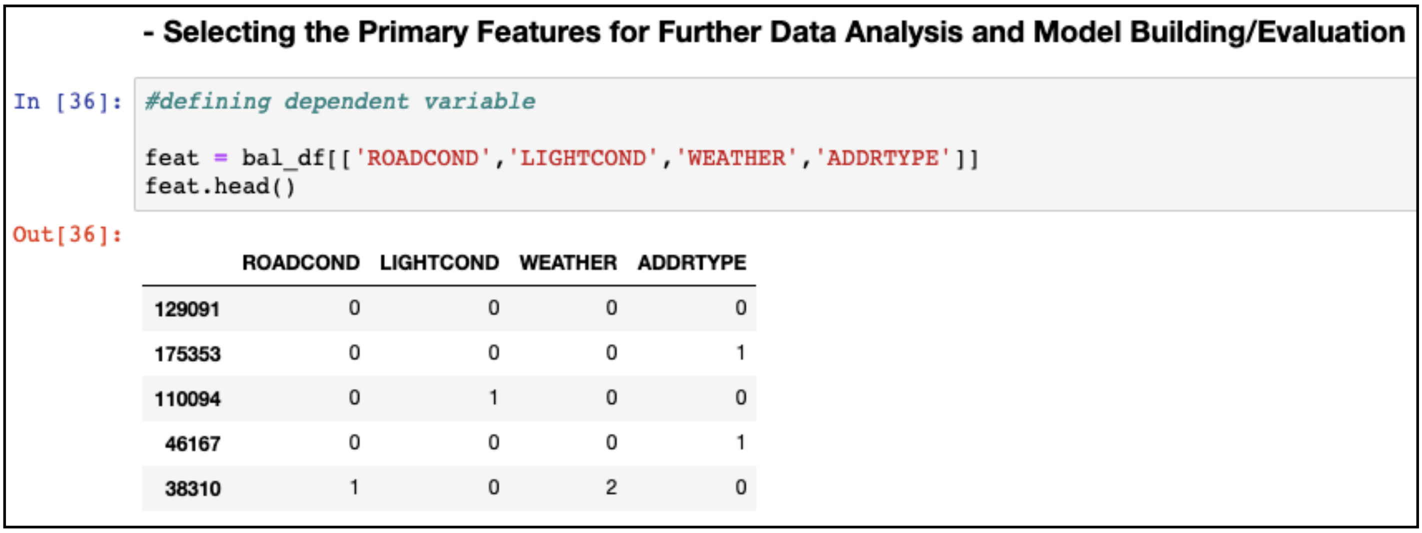
Out[29]:

	SEVERITYCODE	ADDRTYPE	COLLISIONTYPE	PERSONCOUNT	PEDCOUNT	PEDCYLCOUNT	VEHCOUNT	INCDATE	INCDTTM
129091	1	0	Sideswipe	2	0	0	2	2014-05-17 00:00:00+00:00	2014-05-17 15:35:00
175353	1	1	Left Turn	5	0	0	2	2018-03-11 00:00:00+00:00	2018-03-11 15:04:00
110094	1	0	Parked Car	2	0	0	2	2012-08-26 00:00:00+00:00	2012-08-26 03:55:00
46167	1	1	Left Turn	5	0	0	2	2007-03-14 00:00:00+00:00	2007-03-14 18:00:00
38310	1	0	Parked Car	2	0	0	2	2006-10-19 00:00:00+00:00	2006-10-19 08:38:00

Figure 6

After all this preprocessing we finally select our attributes for model fitting which have significant impact on severity of accident and which can be used for model fitting. The attributes are:

1. **WEATHER**: It reaves about weather conditions of the day
2. **ROADCOND**: It gives information about road conditions.
3. **LIGHTCOND**: It gives information about the kind of light available during accident.
4. **ADDRTYPE**: It reveals the location or address of accident out of Alley, Block and Intersection.



- Selecting the Primary Features for Further Data Analysis and Model Building/Evaluation

In [36]:

```
#defining dependent variable

feat = bal_df[['ROADCOND','LIGHTCOND','WEATHER','ADDRTYPE']]
feat.head()
```

Out[36]:

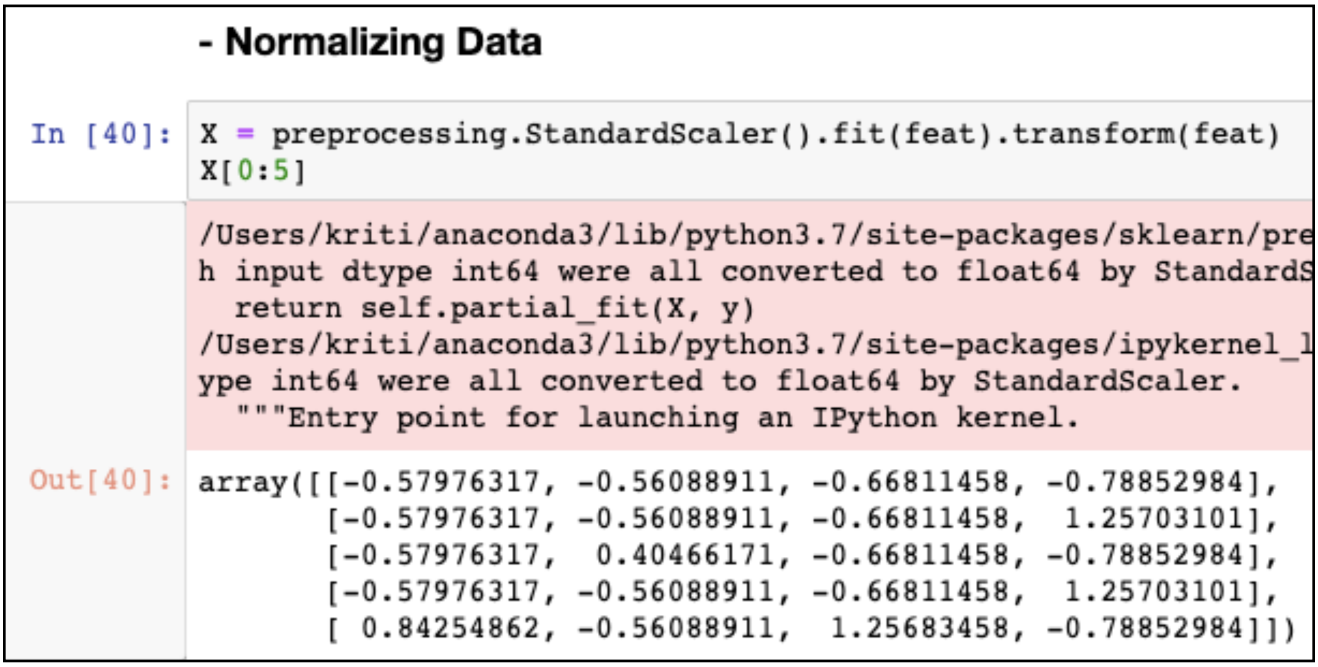
	ROADCOND	LIGHTCOND	WEATHER	ADDRTYPE
129091	0	0	0	0
175353	0	0	0	1
110094	0	1	0	0
46167	0	0	0	1
38310	1	0	2	0

Figure 7

We got a perfectly balanced dataset with our chosen attributes to determine the target.

METHODOLOGY

Our data is ready after preprocessing but before using it into machine learning algorithms, we normalize and transform our data set which makes our data suitable for training the model.

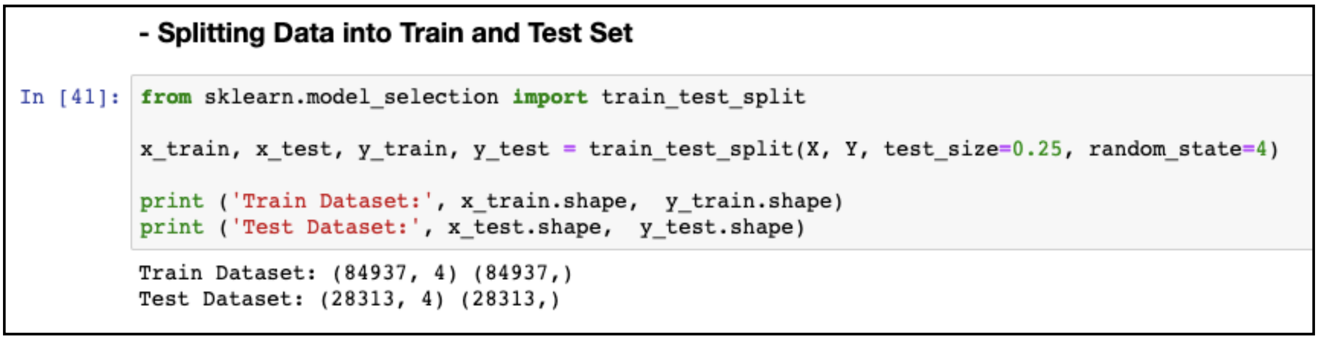


- Normalizing Data

```
In [40]: X = preprocessing.StandardScaler().fit(feat).transform(feat)
         X[0:5]

         /Users/kriti/anaconda3/lib/python3.7/site-packages/sklearn/pre
         h input dtype int64 were all converted to float64 by StandardS
           return self.partial_fit(X, y)
         /Users/kriti/anaconda3/lib/python3.7/site-packages/ipykernel_l
         ype int64 were all converted to float64 by StandardScaler.
           """Entry point for launching an IPython kernel.

Out[40]: array([[-0.57976317, -0.56088911, -0.66811458, -0.78852984],
                [-0.57976317, -0.56088911, -0.66811458,  1.25703101],
                [-0.57976317,  0.40466171, -0.66811458, -0.78852984],
                [-0.57976317, -0.56088911, -0.66811458,  1.25703101],
                [ 0.84254862, -0.56088911,  1.25683458, -0.78852984]])
```

Figure 8

After this, we divide the data into train set and test set using train_test-split from sklearn.model_selection. It randomly divides our data set into train and test set where train set will be used for training the model and test set will be used for testing the model.



- Splitting Data into Train and Test Set

```
In [41]: from sklearn.model_selection import train_test_split

         x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.25, random_state=4)

         print ('Train Dataset:', x_train.shape,  y_train.shape)
         print ('Test Dataset:', x_test.shape,  y_test.shape)

         Train Dataset: (84937, 4) (84937,)
         Test Dataset: (28313, 4) (28313,)
```

Figure 9

We will use following models in our analysis:

1. K-NEAREST NEIGHBOR (KNN)
2. DECISION TREE
3. LOGISTIC REGRESSION

KNN Algorithm

This algorithm helps us in finding the severity of an accident by taking into account k-nearest neighbors from the data set. we train our model on different values of k from 1 to 15 and checked its accuracy using Jaccard index and F1 score. We find that model gives best result for k= 12 with a high corresponding Jaccard score and F1 score. The Jaccard index value was high so we finally selected our model with k = 12 and used it on test data set. We evaluated the model by checking it's accuracy using Jaccard index and F1 score.

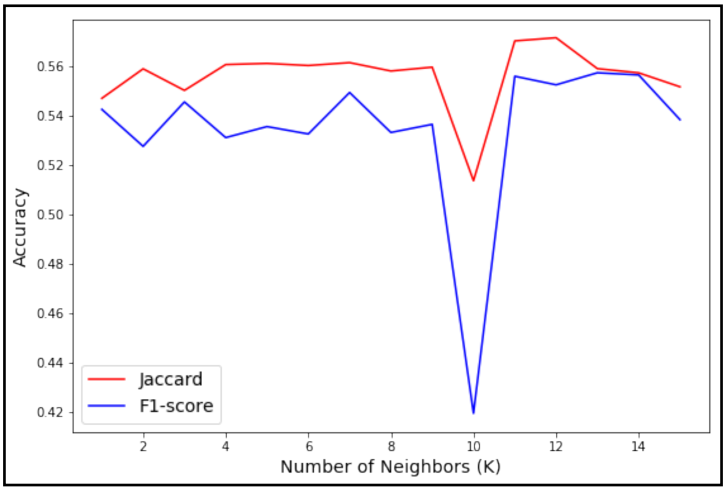


Figure 10

Decision Tree

A decision tree model gives us a layout of all possible outcomes and it evaluates the possible consequences in each situation. We can vary the depth of decision tree to check where it gives best result and we got the best outcome for depth = 5. We used it to train our model and predict the values of our test sample. To find its accuracy we used both Jaccard index and F1 score.

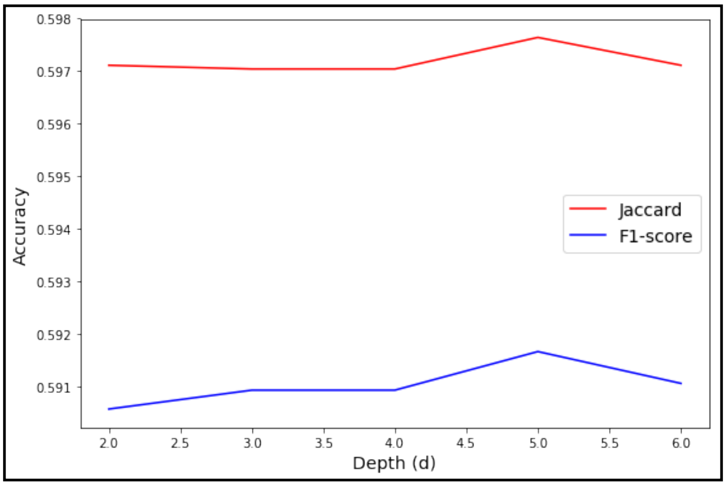


Figure 11

Logistic Regression

Logistic regression is perfect to work with when we have two possible outcomes and interestingly our data set predicts two outcome for our target label, making it binary. Thus we chose Logistic regression where we can vary the solver function and regularization value. After training our model and checking it's accuracy, we observed that it gives best result for Sag as solver and 0.001 as regularization value. Then using these values we trained our model and tested it using test data set. Besides Jaccard and F1 score we also used Log loss method to predict its accuracy.

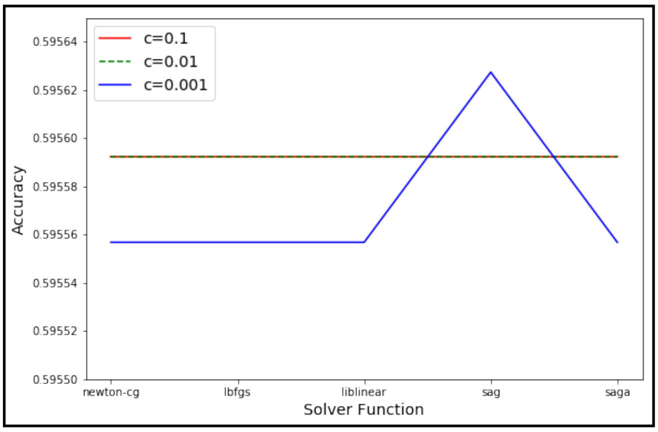
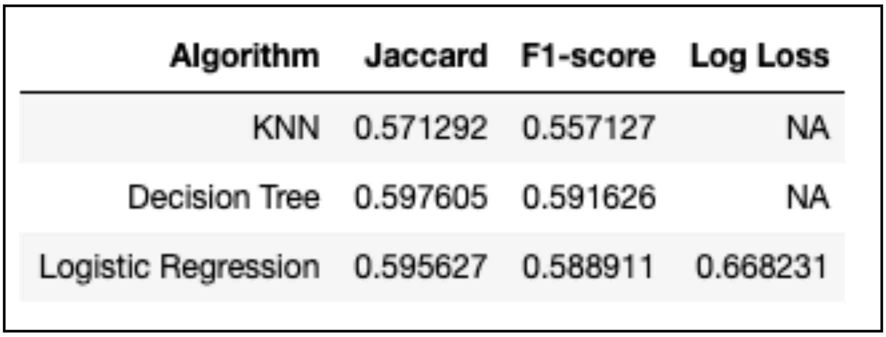


Figure 12

RESULTS AND EVALUATION

We evaluated all the three models which was trained using train data set and tested using test data set. We used Jaccard Index, F1 score and Log loss formula to evaluate the models and the results are displayed in tabular format.



Algorithm	Jaccard	F1-score	Log Loss
KNN	0.571292	0.557127	NA
Decision Tree	0.597605	0.591626	NA
Logistic Regression	0.595627	0.588911	0.668231

Figure 13

DISCUSSION

In the beginning, our data set was raw from which we removed various unwanted attributes. The data was unbalanced so we performed downsampling of our data using resample tool of sklearn. Finally we performed label encoding of our categorical data to make it fit for analysis. Once the preprocessing of Data was done, we normalized the data and divided it into train set and test and then fed it through three different classification machine learning algorithms: K-Nearest neighbor, Decision tree and Logistic Regression. We evaluated these models using Jaccard index, F1-score and Log loss. After all this it was observed that accidents due to insincerity on part of driver whether due to inattention or drink and drive are very minor and it can be controlled by stricter regulations. Contrary to our assumptions the amount of accidents on clear weather is high as compared to days when weather is not perfectly clear and a possible reason could be that people are more careful and attentive during unusual times while they tend to be careless during ordinary days. We also considers the days on which accident are more, and it was observed on bar plots that working days have more accident. Another parameter regulating accident was its address where Alley have significantly higher accidents and it can be controlled by stricter regulations. Using these observations and the models trained, these models can be deployed by various agency to control the loss to property and individual accidents.

CONCLUSION

We used historical data set of Seattle city to predict severity of road accident using KNN algorithm, decision tree and logistic regression using Jaccard index, F1-score and Log loss as estimator. After the analysis, we can say that **decision tree** algorithm can be used predict the severity of road collision for an unknown situation with accuracy of **0.59** using **Jaccard index** as accuracy estimator.