Predict the Severity of a Traffic Accident

Monica
LinkedIn | GitHub

9 October 2020

Contents

1 Introduction

1.1 Background

Every year car accidents cause hundreds of thousands of deaths worldwide. According to a research conducted by the World Health Organization (WHO) there were 1.35 million road traffic deaths globally in 2016, with millions more sustaining serious injuries and living with long-term adverse health consequences. Globally, road traffic crashes are a leading cause of death among young people, and the main cause of death among those aged 15–29 years. Road traffic injuries are currently estimated to be the eighth leading cause of death across all age groups globally and are predicted to become the seventh leading cause of death by 2030[1]. Leveraging the tools and all the information nowadays available, an extensive analysis to predict traffic accidents and its severity would make a difference to the death toll. Analysing a significant range of factors, including weather conditions, locality, type of road and lighting among others, an accurate prediction of the severity of the accidents can be performed. Thus, trends that commonly lead to severe traffic incidents can help identifying the highly severe accidents. This kind of information could be used by emergency services, to send the exact required staff and equipment to the place of the accident, leaving more resources available for accidents occurring simultaneously. Moreover, this severe accident situation can be warned to nearby hospitals which can have all the equipment ready for a severe intervention in advance. Consequently, road safety should be a prior interest for governments, local authorities and private companies investing in technologies that can help reduce accidents and improve overall driver safety.

1.2 Problem

Data that might contribute to determining the likeliness of a potential accident occurring might include information on previous accidents such as road conditions, weather conditions, exact time and place of the accident, type of vehicles involved in the accident, information on the users involved in the accident and off course the severity of the accident. This project aims to forecast the severity of accidents with previous information that could be given by a witness informing the emergency services.

1.3 Interest

Governments should be highly interested in accurate predictions of the severity of an accident, to reduce the time of arrival and to make a more efficient use of the resources, and thus save a significant amount of people each year. Others interested could be private companies investing in technologies aiming to improve road safeness.

2 Data

2.1 Data Source

The dataset is a part of an example dataset in IBM Data Science Professional Certificate. It can be downloaded from my GitHub repository https://github.com/monica110394/Coursera_Capstone. The name of the file is Data-Collisions.csv.

2.2 Variable Selection

After reading the data in pandas, analysis is performed to find the number of samples present in the dataset, also the number of attributes. The dataset has 194,673 records and 38 attributes including the target variable. The columns present in the dataset are:

```
'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'
```

Going through the description of every attribute, which can also be found at my GitHub repository with the file name Metadata.pdf, intuitionally it can be seen that many of the attributes will not contribute to the prediction. The attributes that have been ignored for this prediction are:

```
'X', 'Y', 'OBJECTID', 'INCKEY', 'COLDETKEY', 'REPORTNO, 'STATUS', 'INTKEY', 'LOCAT
ION', 'EXCEPTRSNCODE', 'EXCEPTRSNDESC', 'SEVERITYCODE.1',
'SEVERITYDESC', 'INCDATE', 'INCDTTM', 'SDOT_COLDESC', 'PEDROWNOTGRNT',
'SDOTCOLNUM', 'SPEEDING', 'ST_COLDESC', 'SEGLANEKEY', 'CROSSWALKKEY'
```

The attributes that are kept for the prediction are (17 variable including target variable):

```
'SEVERITYCODE', 'ADDRTYPE', 'COLLISIONTYPE', 'PERSONCOUNT', 'PEDCOUNT',
'PEDCYLCOUNT', 'VEHCOUNT', 'JUNCTIONTYPE', 'SDOT_COLCODE',
'INATTENTIONIND’, 'UNDERINFL', 'WEATHER', 'ROADCOND', 'LIGHTCOND',
'SPEEDING', 'ST_COLCODE', 'HITPARKEDCAR'
```

Target variable: **'SEVERITYCODE'**

2.3 Feature Engineering

After knowing about the dataset, feature engineering is performed such as:

- Null Values are detected
    - **SEVERITYCODE   0**
    - ADDRTYPE   1926
    - COLLISIONTYPE   4904
    - PERSONCOUNT   0
    - PEDCOUNT   0
    - PEDCYLCOUNT   0
    - VEHCOUNT   0
    - JUNCTIONTYPE   6329
    - SDOT_COLCODE   0
    - INATTENTIONIND   164868
    - UNDERINFL   4884
    - WEATHER   5081

- ROADCOND 5012
- LIGHTCOND 5170
- SPEEDING 185340
- ST_COLCODE 18
- HITPARKEDCAR 0

- Attributes that have most of the values, more than 70% values, as null, are dropped
    - INATTENTIONIND 164868
    - SPEEDING 185340

- Current number of rows, columns and attributes
    - Number of rows: 194673
    - Number of Columns 15
    - '**SEVERITYCODE**', 'ADDRTYPE', 'COLLISIONTYPE', 'PERSONCOUNT', 'PEDCOUNT', 'PEDCYLCOUNT', 'VEHCOUNT', 'JUNCTIONTYPE', 'SDOT_COLCODE', 'UNDERINFL', 'WEATHER', 'ROADCOND', 'LIGHTCOND', 'ST_COLCODE', 'HITPARKEDCAR'

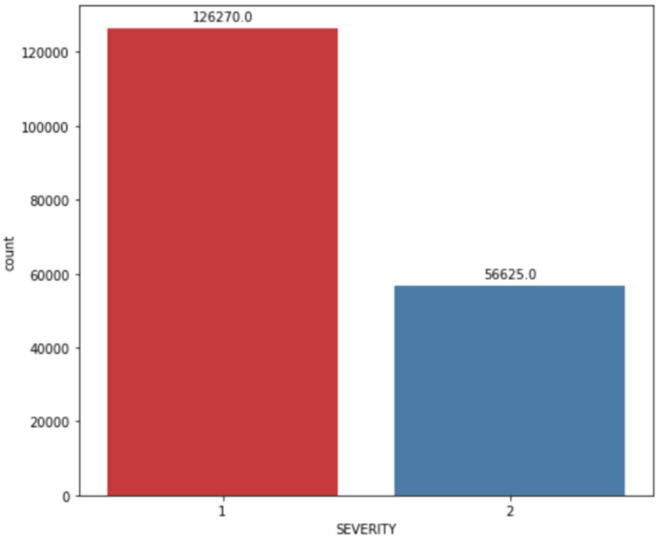
- Records or rows that have missing values are dropped as well.
    - Number of rows: 182895
    - Number of Columns 15

- Label Encoded all the categorical columns converting the ‘string’ labels into numbers and then changed their data types from ‘integer’ to ‘category’.
    - 'ADDRTYPE'
    - 'COLLISIONTYPE'
    - 'JUNCTIONTYPE'
    - 'SDOT_COLCODE'
    - 'UNDERINFL'
    - 'WEATHER'
    - 'ROADCOND'
    - 'LIGHTCOND'
    - 'ST_COLCODE'
    - 'HITPARKEDCAR'

- Numeric columns
    - 'PERSONCOUNT'
    - 'PEDCOUNT'
    - 'PEDCYLCOUNT'
    - 'VEHCOUNT'

- Renaming columns to reasonable names

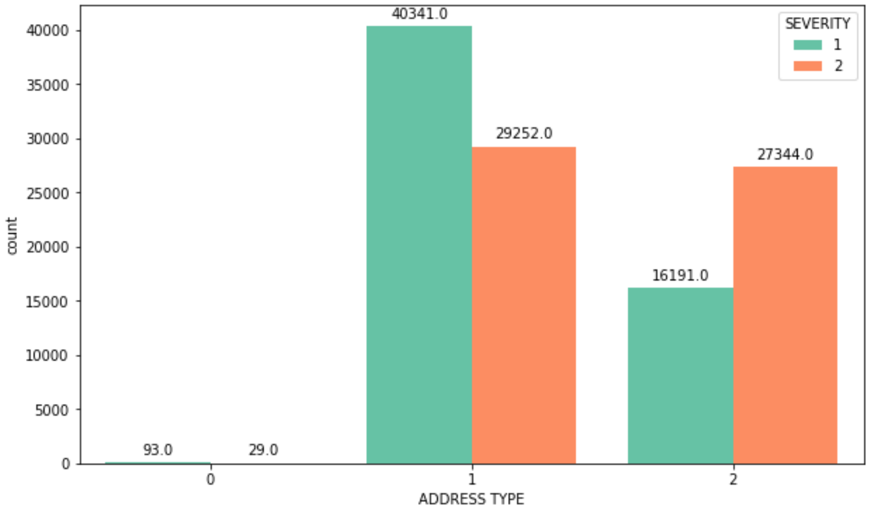
- On plotting a count plot of the target variable, '**SEVERITYCODE**', it is found that the data set is highly unbalanced, which can cause the prediction to be skewed. Hence, the dataset is balanced by down sampling the category whic h as greater number of samples, in this case, when '**SEVERITYCODE**' is 1.

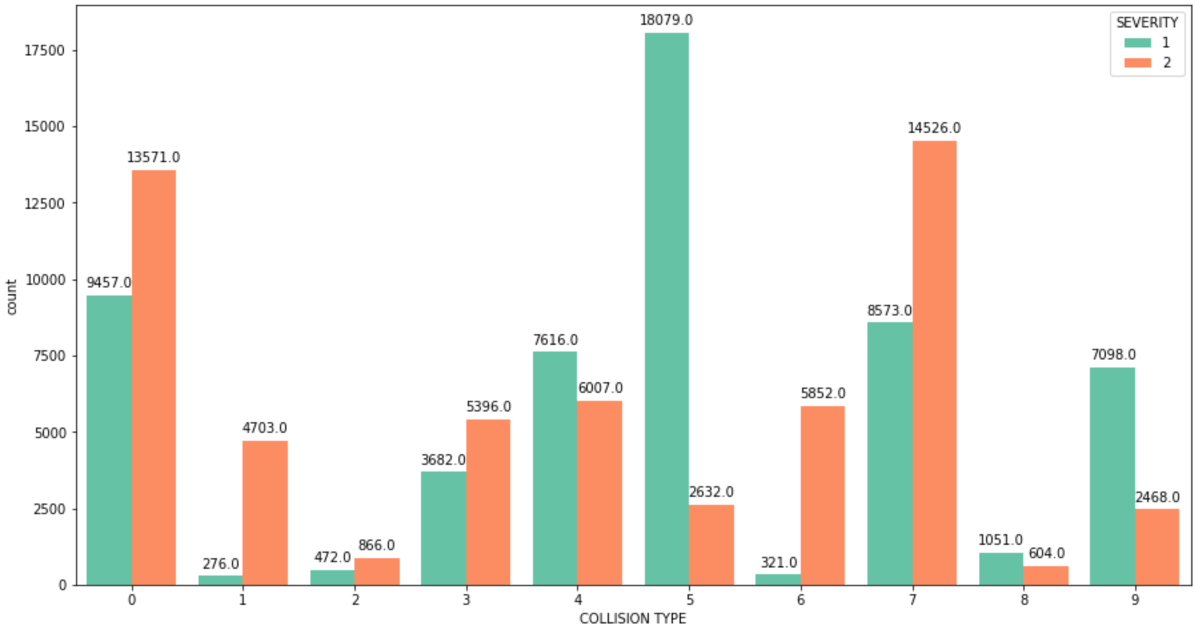


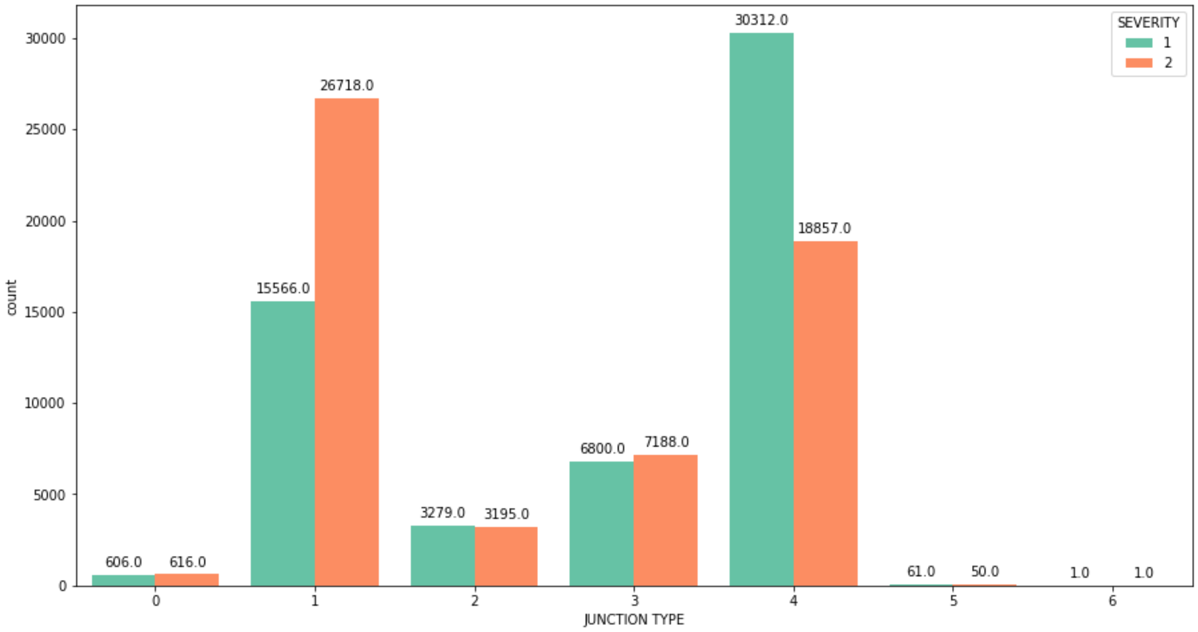
3 Exploratory Data Analysis

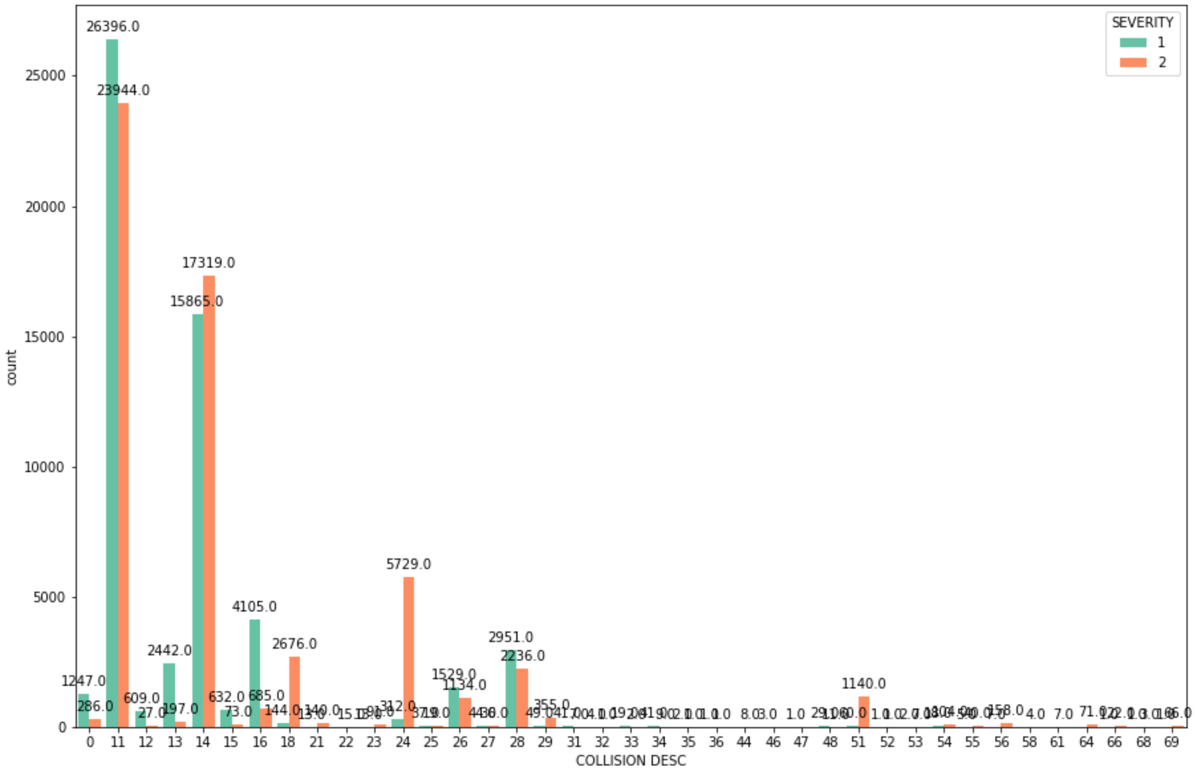
3.1 Categorical Variables

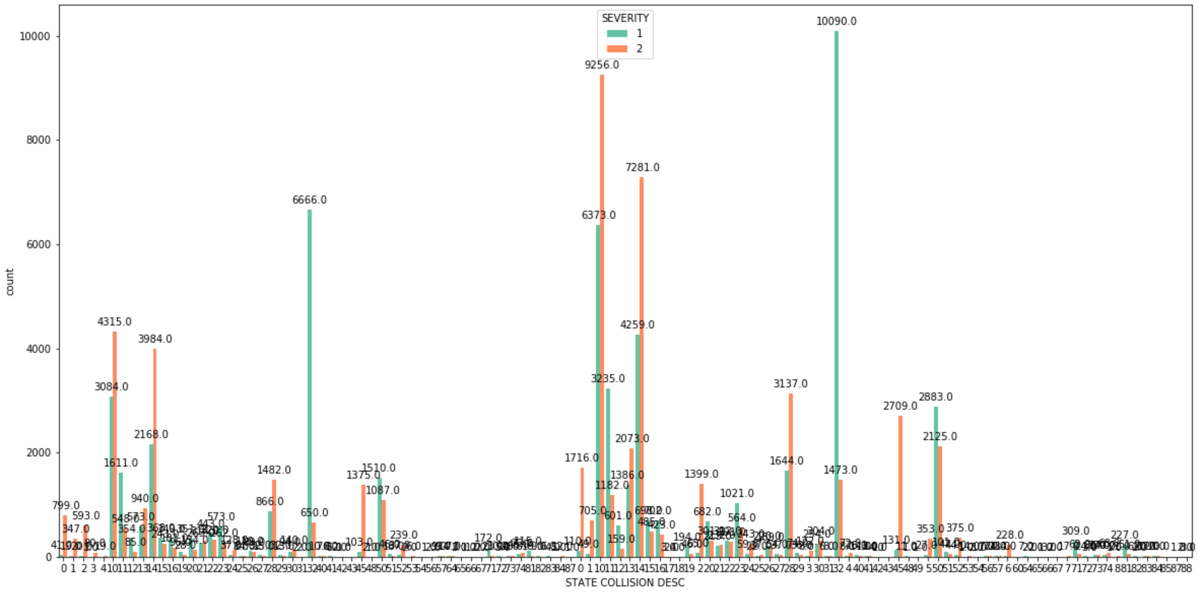
The best way to analyse categorical variables is using a count plot against the target variable `'SEVERITYCODE'`.

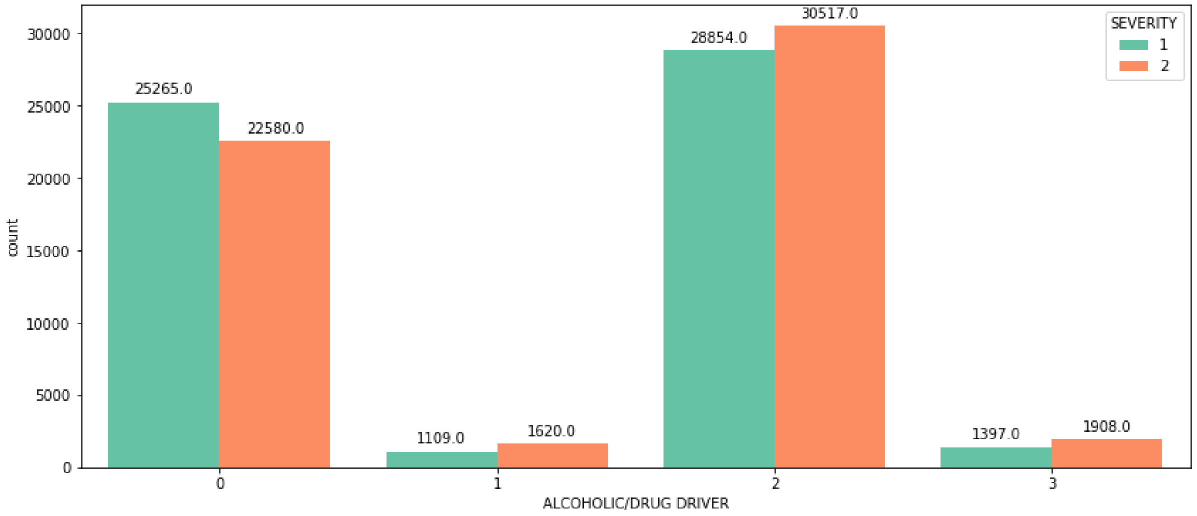


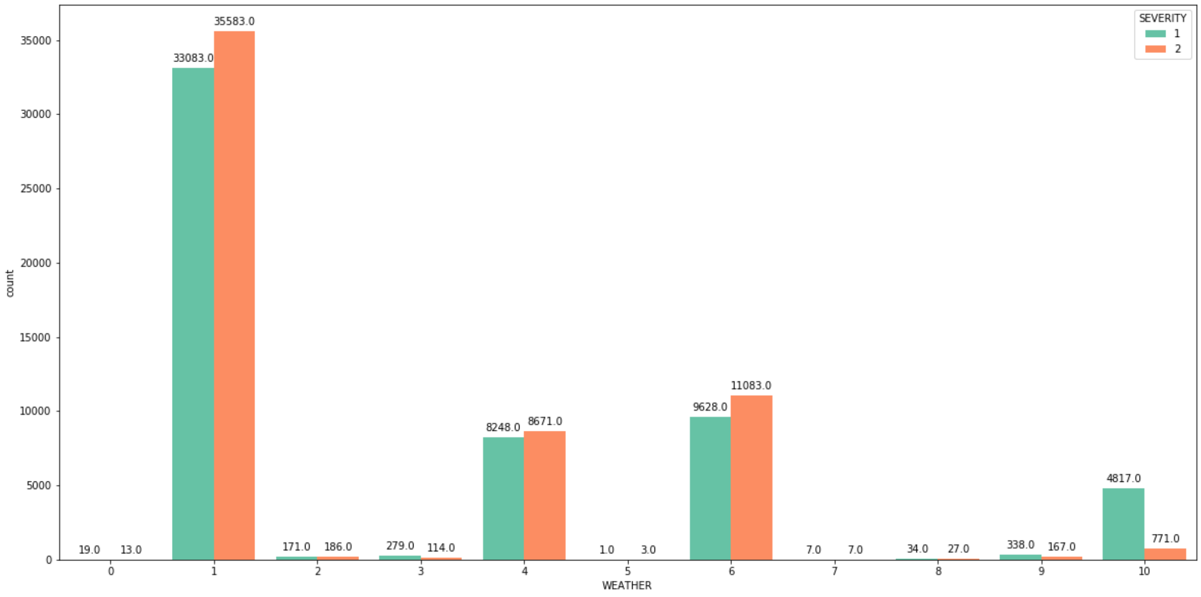


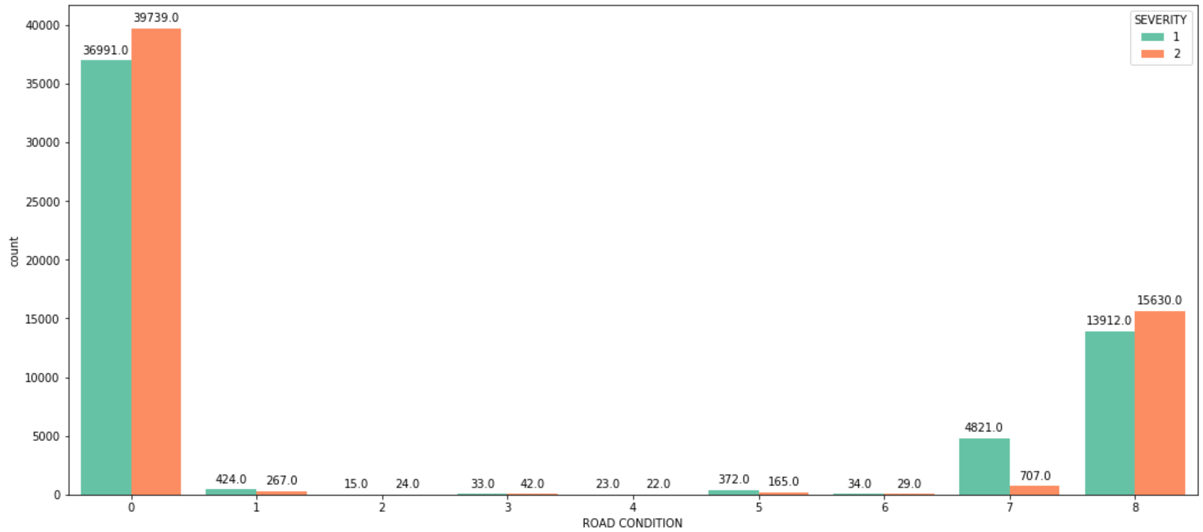

SEVERITY
1
2
count
30000
25000
20000
15000
10000
5000
0
606.0
616.0
15566.0
26718.0
3279.0
3195.0
6800.0
7188.0
30312.0
18857.0
61.0
50.0
1.0
1.0
0
1
2
3
4
5
6
JUNCTION TYPE

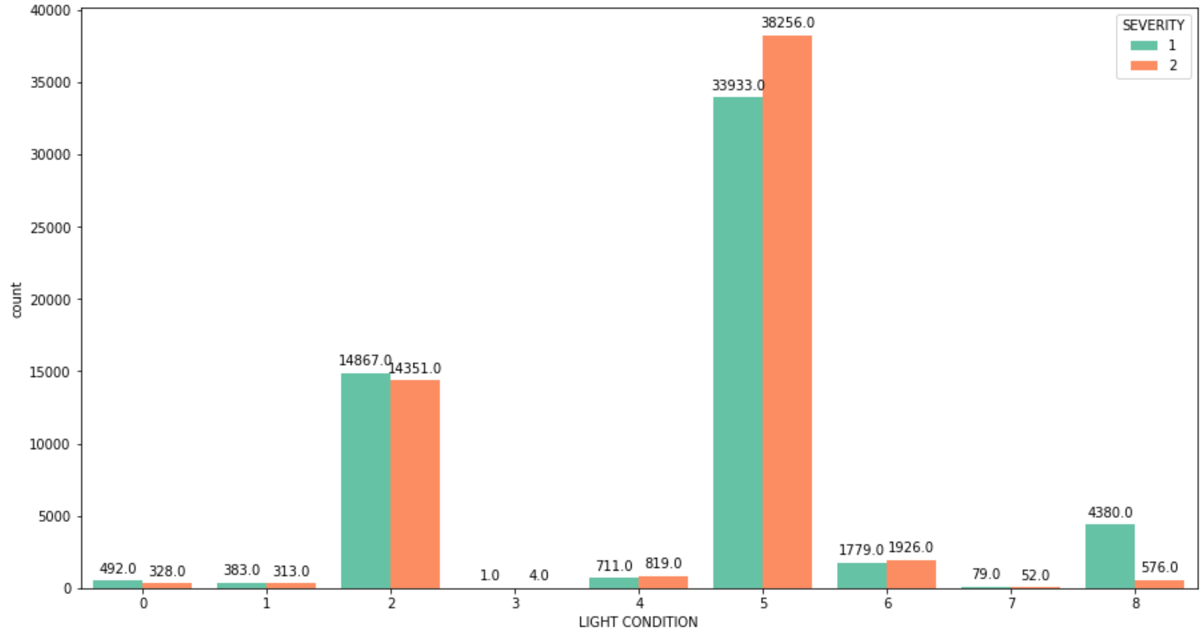

SEVERITY
1
2
count
25000
20000
15000
10000
5000
0
26396.0
23944.0
17319.0
15865.0
5729.0
4105.0
2951.0
2676.0
2442.0
2236.0
1529.0
1247.0
1140.0
1134.0
685.0
632.0
609.0
355.0
312.0
286.0
197.0
158.0
144.0
73.0
71.0
27.0
13.0
8.0
3.0
1.0
4.0
7.0
0
11
12
13
14
15
16
18
21
22
23
24
25
26
27
28
29
31
32
33
34
35
36
44
46
47
48
51
52
53
54
55
56
58
61
64
66
68
69
COLLISION DESC

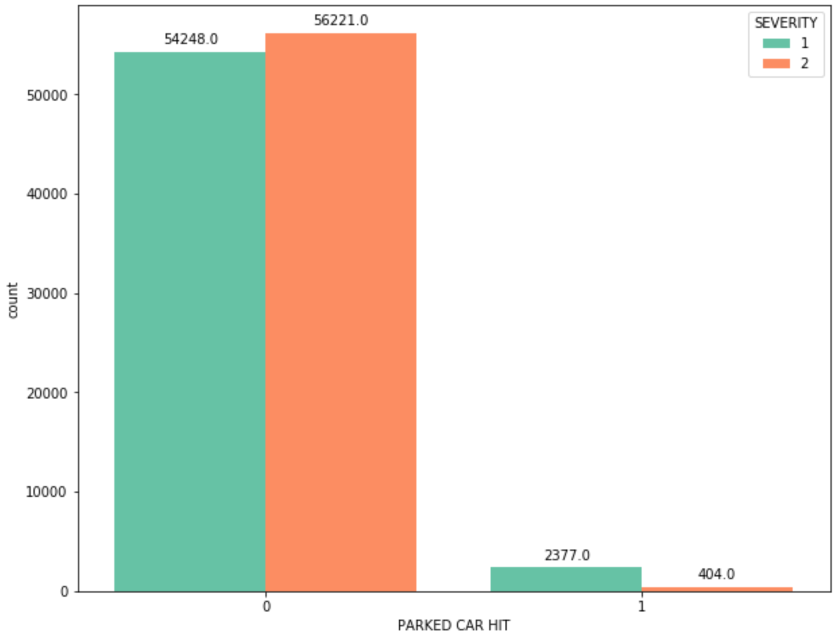

SEVERITY
1
2
count
0
2000
4000
6000
8000
10000
STATE COLLISION DESC
10090.0
9256.0
7281.0
6666.0
6373.0
4315.0
4259.0
3984.0
3235.0
3137.0
3084.0
2883.0
2709.0
2168.0
2125.0
2073.0
1716.0
1644.0
1611.0
1510.0
1482.0
1473.0
1399.0
1386.0
1375.0
1182.0
1087.0
1021.0
940.0
866.0
799.0
705.0
682.0
650.0
601.0
593.0
573.0
564.0
485.0
375.0
353.0
347.0
309.0
239.0
228.0
227.0
194.0
172.0
159.0
131.0
110.0
103.0
85.0


SEVERITY
1
2
count
0
5000
10000
15000
20000
25000
30000
25265.0
22580.0
1109.0
1620.0
28854.0
30517.0
1397.0
1908.0
0
1
2
3
ALCOHOLIC/DRUG DRIVER


SEVERITY
1
2
count
0
5000
10000
15000
20000
25000
30000
35000
19.0
13.0
33083.0
35583.0
171.0
186.0
279.0
114.0
8248.0
8671.0
1.0
3.0
9628.0
11083.0
7.0
7.0
34.0
27.0
338.0
167.0
4817.0
771.0
0
1
2
3
4
5
6
7
8
9
10
WEATHER

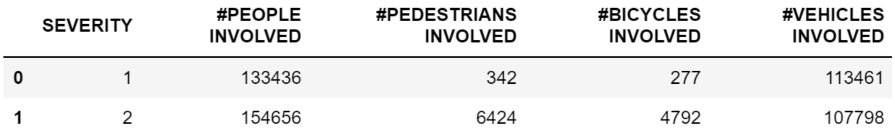

SEVERITY
1
2
count
40000
35000
30000
25000
20000
15000
10000
5000
0
36991.0
39739.0
424.0
267.0
15.0
24.0
33.0
42.0
23.0
22.0
372.0
165.0
34.0
29.0
4821.0
707.0
13912.0
15630.0
0
1
2
3
4
5
6
7
8
ROAD CONDITION


SEVERITY
1
2
count
40000
35000
30000
25000
20000
15000
10000
5000
0
492.0
328.0
383.0
313.0
14867.0
14351.0
1.0
4.0
711.0
819.0
33933.0
38256.0
1779.0
1926.0
79.0
52.0
4380.0
576.0
0
1
2
3
4
5
6
7
8
LIGHT CONDITION

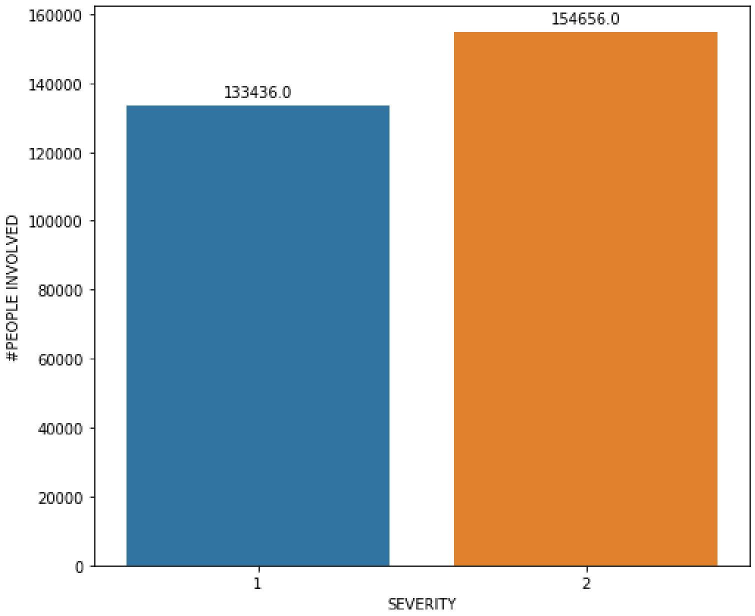

SEVERITY
1
2
count
50000
40000
30000
20000
10000
0
54248.0
56221.0
2377.0
404.0
0
1
PARKED CAR HIT

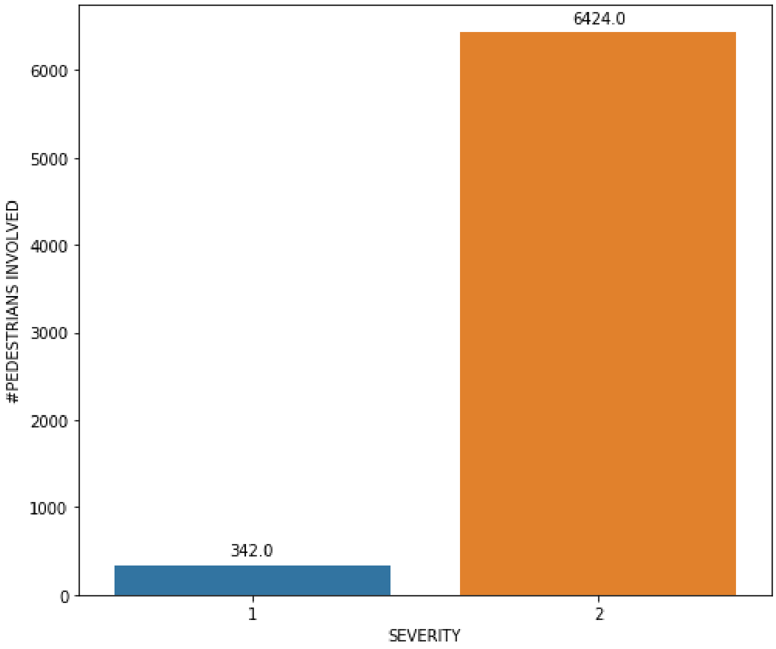
3.2 Numerical Variables

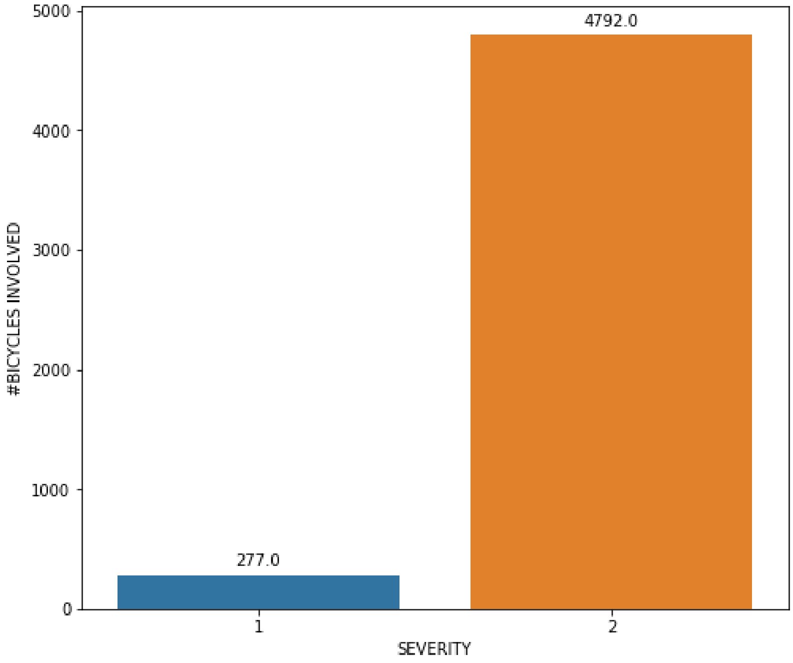
For numerical variables, they are grouped and aggregated as sum over both the categories of the target variable `'SEVERITYCODE'`. After which, bar plots is used to analyse the variables against the target variables.

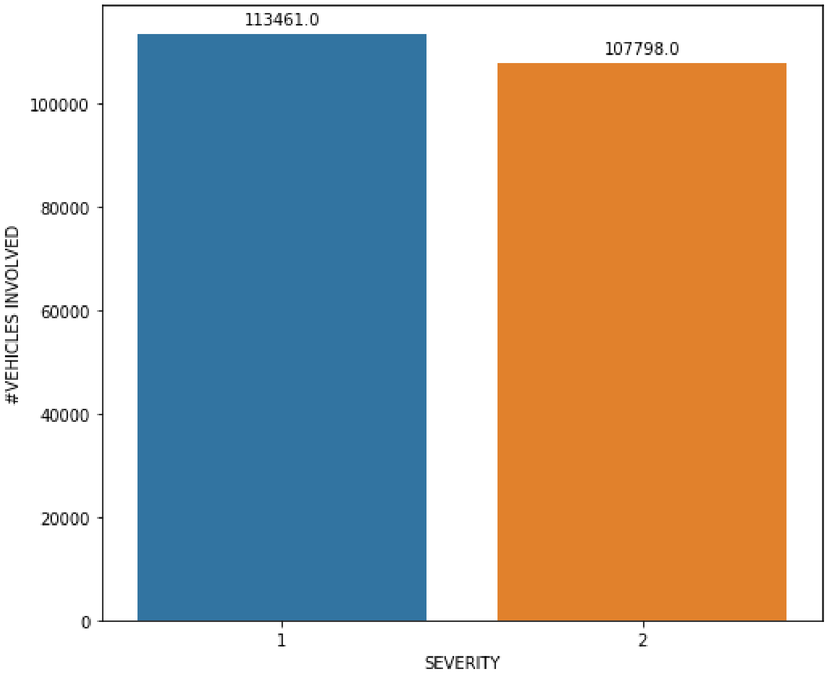


	SEVERITY	#PEOPLE INVOLVED	#PEDESTRIANS INVOLVED	#BICYCLES INVOLVED	#VEHICLES INVOLVED
0	1	133436	342	277	113461
1	2	154656	6424	4792	107798






5000
4000
3000
2000
1000
0
#BICYCLES INVOLVED
4792.0
277.0
1
2
SEVERITY


100000
80000
60000
40000
20000
0
#VEHICLES INVOLVED
113461.0
107798.0
1
2
SEVERITY

4 Predictive Modeling

4.1 Data Preparation

Firstly, the features and the target variables are saved into two different data frames. Next, the features' dataset is normalized using '`StandardScaler()`', to bring the dataset to zero mean and unit variance.

```
from sklearn import preprocessing
X = preprocessing.StandardScaler().fit(features).transform(features)
```

After the normalization process the whole dataset it divided into train and test datasets with a test size of 20% of the whole dataset.

```
Train set: (90600, 14) (90600,)
Test set: (22650, 14) (22650,)
```

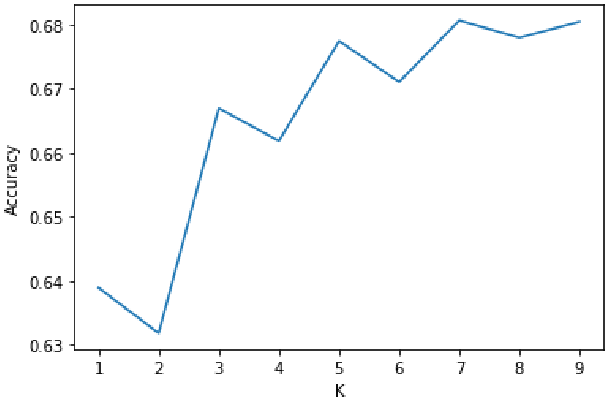
4.2 Data Modeling

To predict the severity of a traffic accident, 3 different classification methods are used. Since most of the features are categorical, appropriate classification methods are used. All the classification methods are implemented using the scikit-learn package that is available as a python library. For every classification, the best hyperparameter value is computed using grid search to maximum accuracy. The algorithms that were used are:

- K Nearest Neighbours
- Decision Tree
- Logistic Regression

4.2.1 KNN

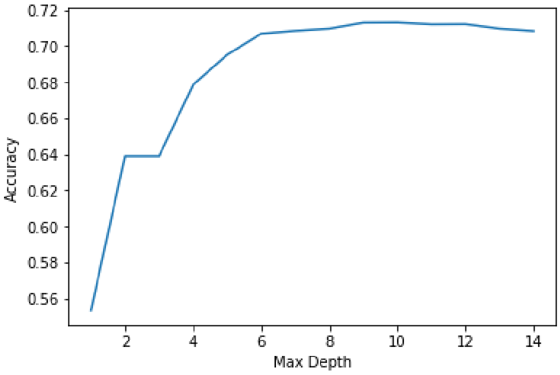
The best accuracy score achieved was 0.6806 with k=7.



```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=7, p=2,
                     weights='uniform')

Train set Accuracy:  0.7195

Test set Accuracy:  0.6806
```

4.2.2 Decision Tree

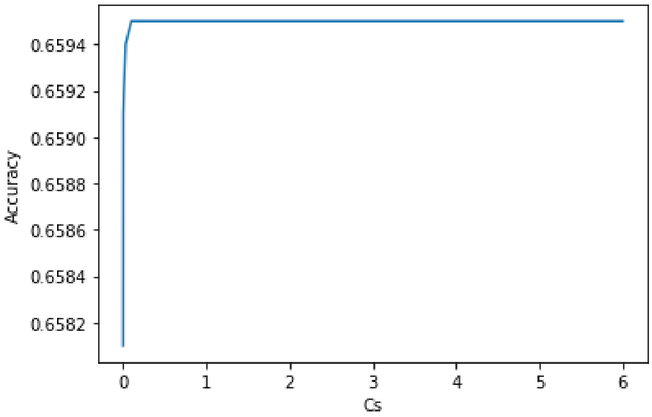
The best accuracy score achieved was 0.6806 with max_depth = 10



```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='entropy',
                       max_depth=10, 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='deprecated',
                       random_state=None, splitter='best')

Train set Accuracy:  0.7183

Test set Accuracy:  0.7128
```

4.2.3 Logistic Regression

The best accuracy score achieved was 0.6806 with C=0.1

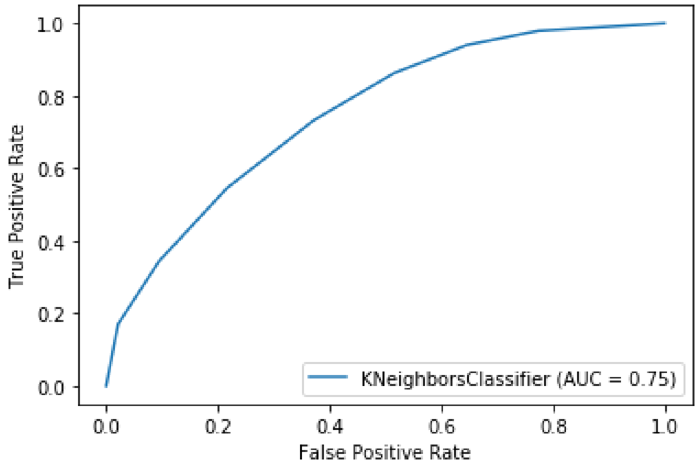


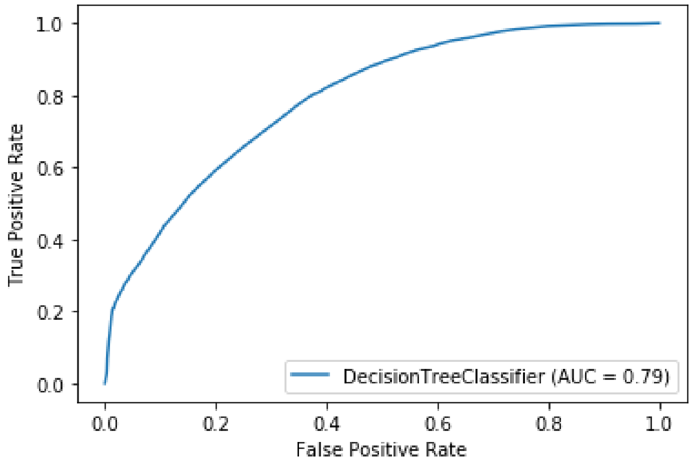
```
LogisticRegression(C=0.1, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='liblinear', tol=0.0001, verbose=0,
                   warm_start=False)

Train set Accuracy:  0.6591

Test set Accuracy:  0.6595
```

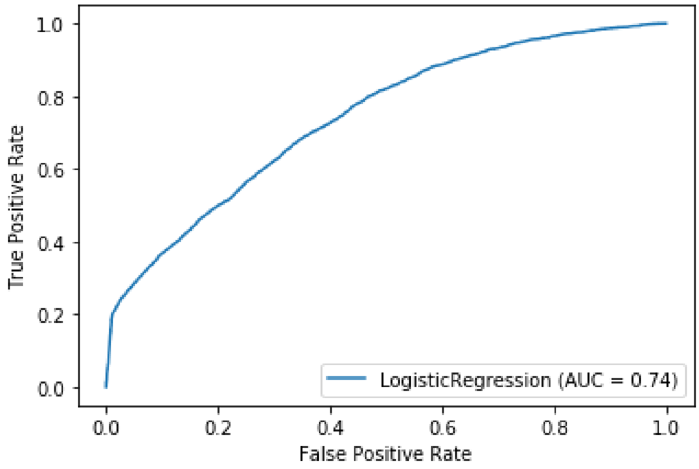
5 Results

All the classification models are evaluated against each other using evaluations metrics provided by the scikit learn library. The metrics are:

- Jaccard Index
- F1 Score
- Precision Score
- Recall Score
- ROC
  In this problem, lower false positive rate is less important than higher true positive rate. In other words, it is more important to properly predict the high-severity accident properly, if there is room for doubt it is better to prevent.
- Log Loss (Logistic Regression only)







Algorithm	Jaccard	F1 Score	Precision	Recall	AUC (ROC)	Log Loss
KNN	0.6806	0.6797	0.6991	0.6284	0.75	NA
Decision Tree	0.7128	0.7104	0.7582	0.6207	0.79	NA
Logistic Regression	0.6595	0.6585	0.6417	0.7144	0.74	0.5899

6 Conclusion

In the analysis done above and looking at the table, it can be said that Decision Tree is the best classifier out of the three with the best Jaccard Index 0.71, best F1 Score 0.75 and best AUC. Although, there is a possibility to achieve a better accuracy by a Random Forest Classifier, or with a reduced number of features, or adding relevant features to the dataset.

These models can have multiple application in real life. For instance, imagine that emergency services have an application with some default features such as date, time and department/municipality and then with the information given by the witness calling to inform on the accident they could predict the severity of the accident before getting there and so alert nearby hospitals and prepare with the necessary equipment and staff. Also, by identifying the features that favour the most the gravity of an accident, these could be tackled by improving road conditions or increasing the awareness of the population.

7 Discussion

For this study, only three classifiers were used, that is, KNN, Decision Tree and Logistic Regression but other classifiers can also be applied to the data such as Naïve Bayes, SVM and Random Forest, which might or might not produce a better accuracy metric.

As per the data, for this problem, i.e. to predict the severity of an accident, I feel a lot more relevant features could be added such as number of accidents per year which could be helpful to determine the frequency of accidents happening and also, the age of the person involved in the accident. This dataset had lots of missing values in two of the most important features of the dataset, SPEEDING (speeding or not) and INATTENTIONID (collision due to inattention), which could have increased the accuracy of the classification models.

The next step on this problem could be to add an accident prediction model able to not just predict the accuracy but also the critical time and spots where potential accidents can occur in advance.

8 References

Josep-at-work/Coursera_Capstone. (2020). Retrieved 9 October 2020, from https://github.com/Josep-at-work/Coursera_Capstone/blob/master/Predicting_Traffic_Accident_Severity_.pdf