Predicting Road Accident Severity

1. Introduction

Traffic accidents represent one of the leading causes of death worldwide and of economic expenditure. Despite the numerous measures and campaigns that are deployed every year to raise awareness of the seriousness of the problem, it still occurs quite frequently. The impact of road accidents on society and the economy is high, and human losses are compounded by large expenditures on health care, awareness campaigns, mobilization of specialized personnel, etc. The WHO sets the economic impact of road accidents in a developed country at 2 to 3% of GDP, a significant figure for any country. Collaboration to reduce these losses has become an important issue of general interest.

Analyzing a significant range of factors, including weather conditions, special events, roadworks, traffic jams among others, an accurate prediction of the severity of the accidents can be performed.

These insights, could allow law enforcement bodies to allocate their resources more effectively in advance of potential accidents, preventing when and where a severe-accidents can occur as well as saving both, time and money. In addition, this knowledge of a severe accident situation can be warned to drivers so that they would drive more carefully or even change their route if it is possible or to hospital which could have set everything ready for a severe intervention in advance.

Governments should be highly interested in accurate predictions of the severity of an accident, in order to reduce the time of arrival 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 Understanding

For an accurate prediction of the magnitude of damage caused by accidents, they require a large number of reports on traffic accidents with accurate data to train prediction models. The data set provided for this work allows the analysis of a record of 200,000 accidents in the state of Seattle, from 2004 to the date it is issued, in which 37 attributes or variables are recorded and the codification of the type of accident is allowed, grouped according to 84 codes. The information can be extracted from it: speed information, information on road conditions and visibility, type of collision, affected persons etc.

The data will be used so that we can determine which attributes are most common in traffic accidents in order to target prevention at these high-incidence points.

Data Source: These data have been collected and shared by the Seattle Police Department (Traffic Records)

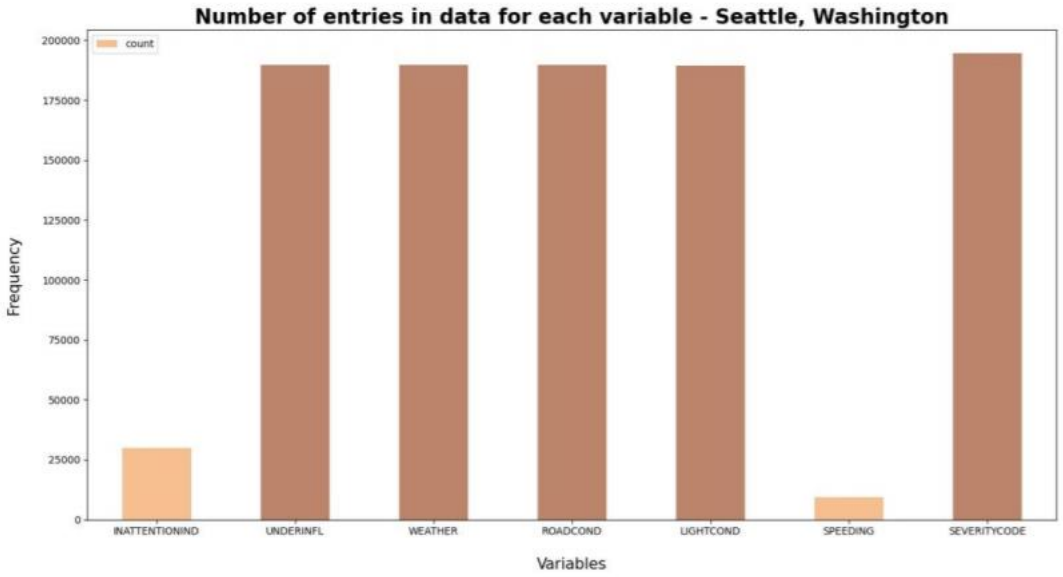
2.2 Data Cleaning

There are a lot of problems with the data set keeping in mind that this is a machine learning project which uses classification to predict a categorical variable. The dataset has total observations of 194673 with variation in number of observations for every feature. First of all, the total dataset was high variation in the lengths of almost every column of the dataset. The dataset had a lot of empty columns which could have been beneficial had the data been present there. These columns included pedestrian granted way or not, segment lane key, cross walk key and hit parked car.

2.2.1 Identifying and Handling missing values

To identify columns and rows with missing values is the next step. Empty boxes, 'Unknown' and 'Other' were values considered as missing values. These were replaced with NA to make the dataset uniform.

```
df.replace(r'^\s*$', np.nan, regex=True)
df.replace("Unknown", np.nan, inplace = True)
df.replace("Other", np.nan, inplace = True)
```



The above figure shows the number of observations for the columns mentioned in the figure. In order to deal with the issue of columns having a variation in frequency, I have removed columns with more than 20% values missing (INATTENTIONIND, PEDROWNOTGRNT, SPEEDING), I have also removed rows for columns with less than 20% values missing and also removed all irrelevant features such as IDs and descriptions.

Once the above strategies were performed, the dataset reduced from having 194673 rows and 15 columns to having 143747 rows and 15 columns.

```
df['SEVERITYCODE'].value_counts()
```

```
1    94821
2    48926
Name: SEVERITYCODE, dtype: int64
```

Our target variable SEVERITYCODE is not well balanced. In fact, severity code in class 1 is nearly twice the size of class 2. I have fixed this by down-sampling the majority class. Down-sampling involves randomly removing observations from the majority class to prevent its signal from dominating the learning algorithm.

The most common heuristic for doing so is resampling without replacement.

1. First, we'll separate observations from each class into different dataframes.
2. Next, we'll resample the majority class without replacement, setting the number of samples to match that of the minority class.
3. Finally, we'll combine the down-sampled majority class Dataframe with the original minority class Dataframe.

After down-sampling the dataset, the ratio of the two classes is has become 1:1 and the dataset has been perfectly balanced.

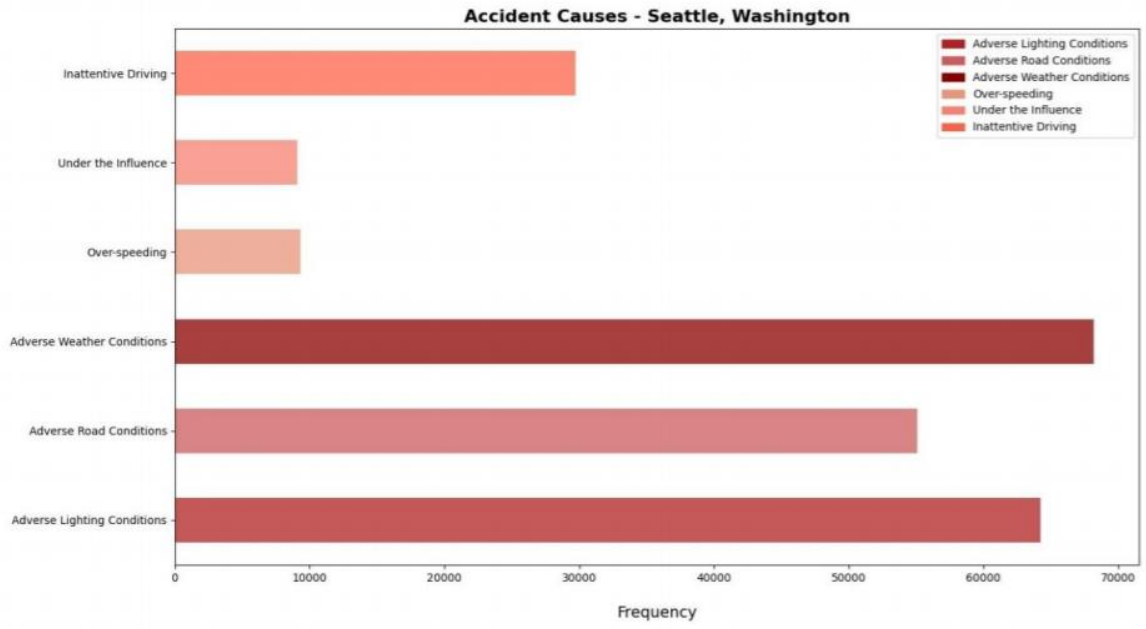
2.3 Feature Selection

Our predictor or target variable will be 'SEVERITYCODE' because it is used to measure the seve rity of an accident within the dataset. Attributes used to weigh the severity of an accident are 'X', ‘Y’, ‘ADDRTYPE’, ‘COLLISIONTYPE’, ‘PEDCOUNT’, ‘PEDCYLCOUNT’, ‘UNDERINFL’, ‘WEATHER', 'ROADCOND', 'JUNCTIONTYPE', 'HITPARKEDCAR', 'LIGHTCOND', 'VEHC OUNT' and 'PERSONCOUNT'.

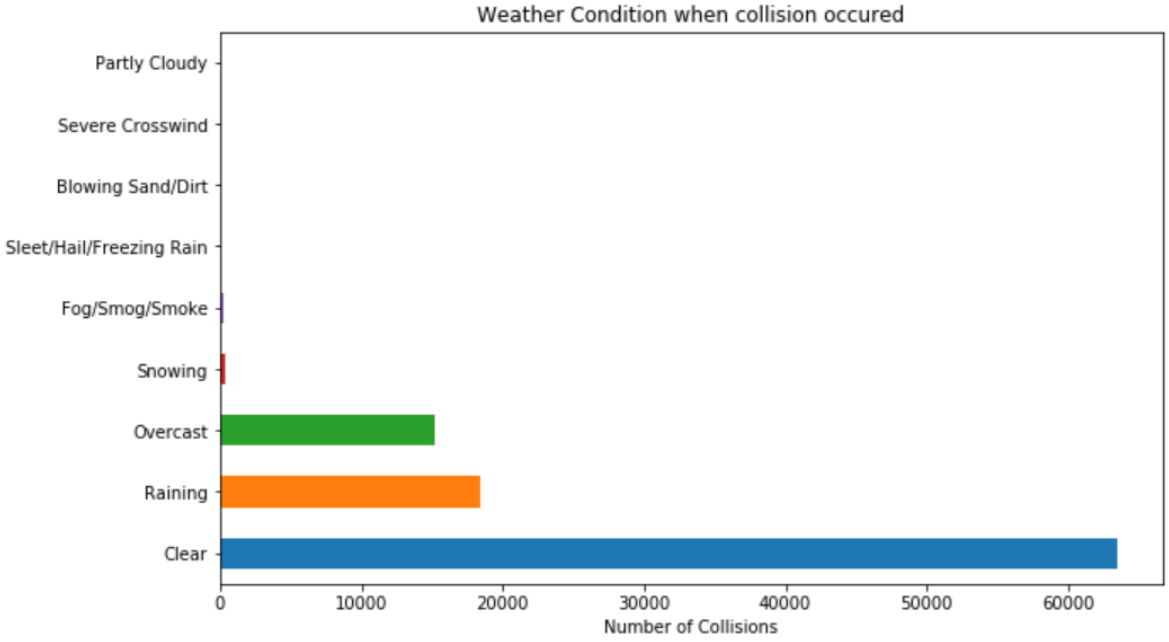
In its original form, this data is not fit for analysis. Most of the features are of type object, when t hey should be numerical type. Machine Learning models are trained only on numerical data; henc e all categorical features in the dataset have to be encoded so that the algorithms can be trained o n those features. The ‘get_dummies’ method from pandas library is used to convert/encode each a nd every categorical feature.

3. Methodology

3.1 Exploratory Data Analysis

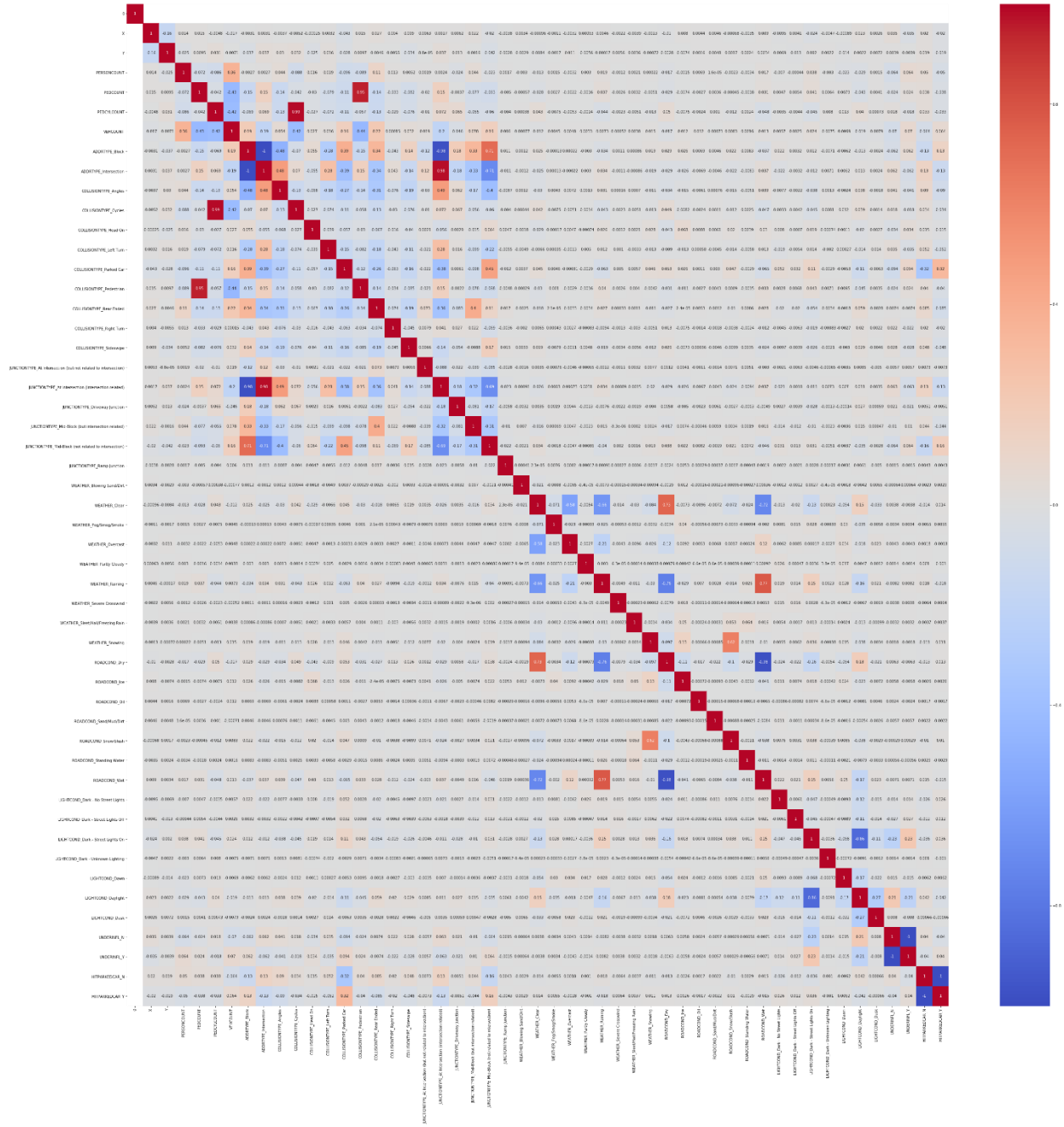


In the graph above we can see the frequency of accidents which took place under adverse conditi ons. The factor which had the greatest number of accidents under adverse conditions was adverse weather conditions while adverse lighting condition had the second greatest number of accidents caused by it. The factors which contributed the least to an instance of an accident are over-speedi ng and the driver being under the influence.



The above figure shows the weather condition when an accident has taken place. Almost 50% of the accidents have occurred when the weather is clear.

Understanding Correlation in Dataset



Correlation is a statistical technique that can show whether and how strongly pairs of variables ar e related. Finding the correlation among the features of the dataset helps understand the data bette r. For example, in the below figure (correlation plot using matplotlib), it can be observed that som e features have a strong positive/negative correlation while most of them have weak/ no correlati

on. Examples, there is a strong positive correlation between ‘PEDCYLCOUNT’ and ‘COLLISIONTYPE_Cycles’. This means that if the collision involves cycles, at-least one cyclist is involved in the accident. There is a strong negative correlation between ‘ROADCOND_Wet’ and ‘ROADCOND_Dry’, meaning that if the road is wet it cannot be dry. This is how we can get a deeper understanding of the data using correlation plots.

3.2 Splitting the Dataset into Training and Testing Datasets

The dataset was split into X_train, y_train, X_test, and y_test. The first two will be used for training purposes and the last two will be used for testing purposes. The split ratio is 0.8, 80% of data is used for training and 20% of is used for testing.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=4)
print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

```
Train set: (78281, 50) (78281,)
Test set: (19571, 50) (19571,)
```

3.3 Normalizing/ Feature scaling of the data

Feature scaling of data is done to normalize the data in a dataset to a specific range. It also helps improve the performance of the ML algorithms. Standard Scaler metric is used to scale/normalize all the numerical data for both, the X_train and X_test datasets. This completes the pre-processing stage, we can move on to training our models.

3.4 Applying Machine Learning Algorithms

A total of four ML algorithms were trained on the pre-processed dataset and their accuracies were compared. A brief explanation on how each of them works along with their results in shown below.

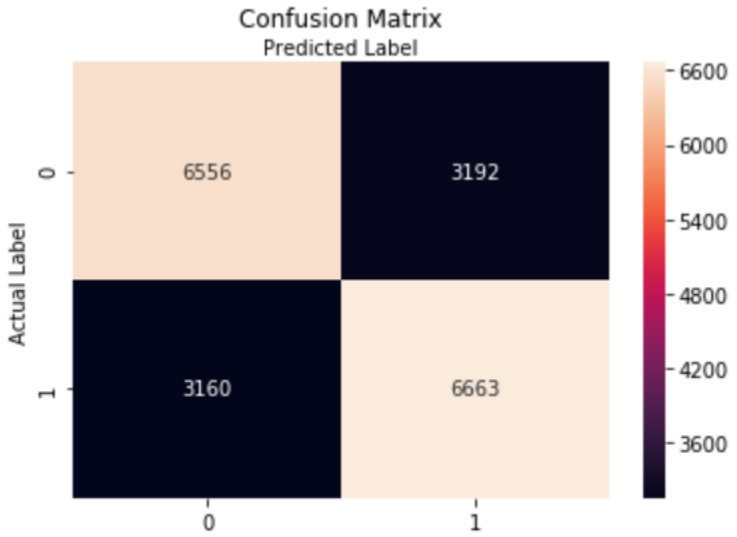
- **K Nearest Neighbors Classifier**
  K nearest neighbors algorithm used for both classification and regression problems. It basically stores all available cases to classify the new cases by a majority vote of its k neighbors. The case assigned to the class is most common amongst its K nearest neighbors measured by a distance function (Euclidean, Manhattan, Minkowski, and Hamming).

  In order to arrive at the optimum values for nearest neighbors (k) and the distance metric (Euclidean and Manhattan), a hyper parameter KNN was used. The best accuracy was obtained for 7 nearest neighbors with Euclidean being the distance metric when applied for the problem.

  The confusion matrix and accuracy are:

```
print("Train set Accuracy: ", metrics.accuracy_score(y_train, neigh.predict(X_train)))
print("Test set Accuracy: ", metrics.accuracy_score(y_test, yhat))
```

```
Train set Accuracy:  0.755777263959326
Test set Accuracy:  0.6754381482806193
```



- **Decision Tree Classifier**

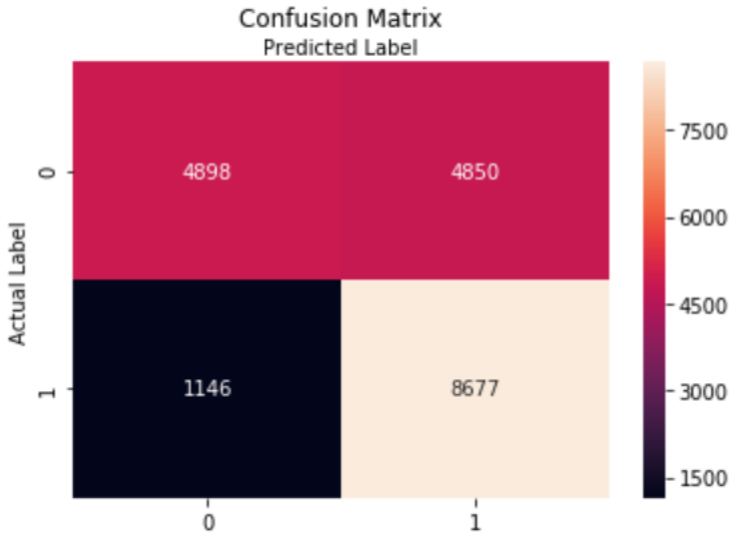
  Decision Tree makes decision with tree-like model. It splits the sample into two or more homogenous sets (leaves) based on the most significant differentiators in the input variables. To choose a differentiator (predictor), the algorithm considers all features and does a binary split on them (for categorical data, split by category; for continuous, pick a c ut-off threshold). It will then choose the one with the least cost (i.e. highest accuracy), and repeats recursively, until it successfully splits the data in all leaves (or reaches the maxim um depth).

  Information gain for a decision tree classifier can be calculated either using the Gini Index measure or the Entropy measure, whichever gives a greater gain. A hyper parameter Decision Tree Classifier was used to decide which tree to use, DTC using entropy had greater information gain; hence it was used for this classification problem.

  The confusion matrix and accuracy are:

```
print('Decision Tree\'s Accuracy: ', metrics.accuracy_score(y_test, predtree))
```

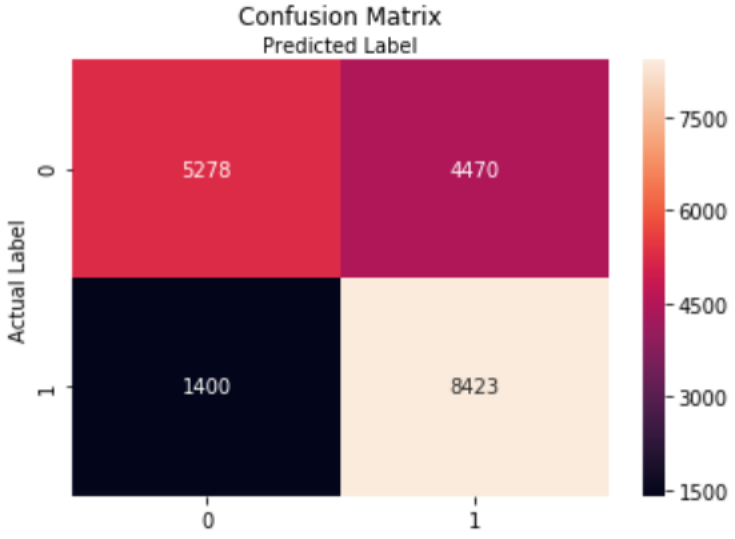
```
Decision Tree's Accuracy:  0.6936283276276123
```



- **Support Vector Machine Classifier**

  In the SVM algorithm, each data item is plotted as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, classification is performed by finding the hyper-plane that differentiates the two classes. Hyper parameter SVC was used to choose between Linear SVC and a Kernel SVC and the latter arrived on top with a greater accuracy when applied on the dataset in question. It used the ‘radial basis function’ kernel for performing the clas sification.

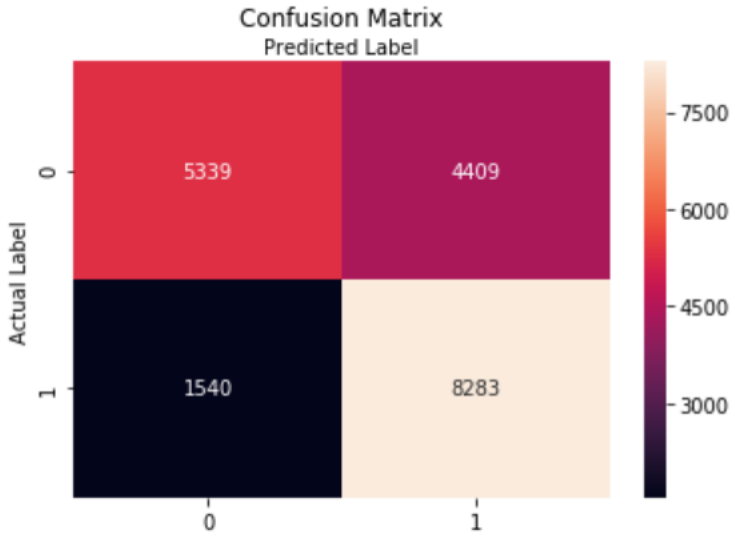
  The confusion matrix and accuracy are:



- **Logistic Regression Classifier**

  Logistic Regression is a classifier that estimates discrete values (binary values like 0/1, yes/no, true/false) based on a given set of an independent variables. It basically predicts the probability of occurrence of an event by fitting data to a logistic function. Hence it is also known as logistic regression.

  The chosen dataset has only two target categories in terms of the accident severity code assigned; hence it was possible to apply this model to the same. The confusion matrix, f1 score and jaccard similarity score are:




```
lr_acc = jaccard_similarity_score(y_test, LR_yhat)
lr_acc
```

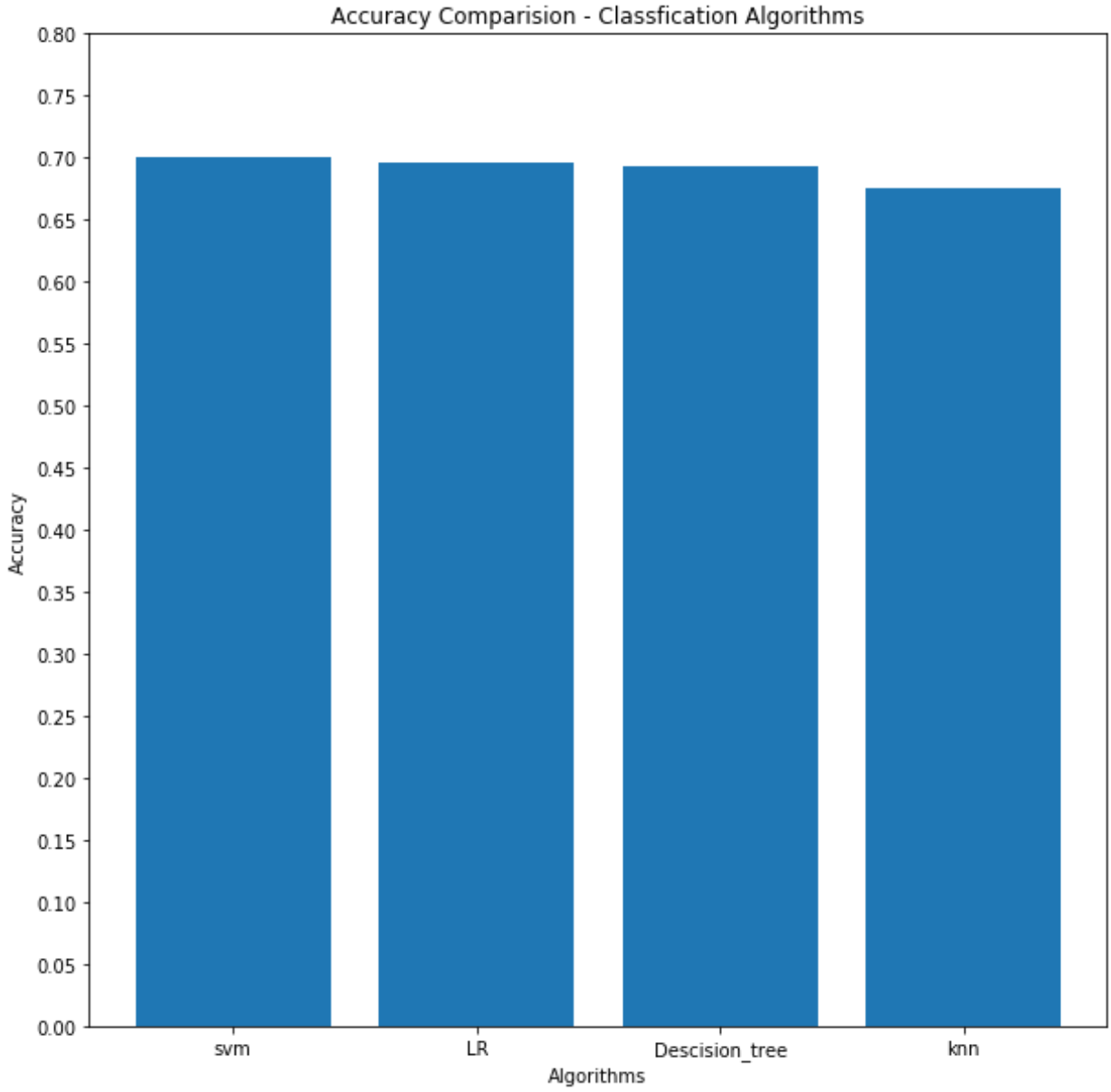
```
0.6960298400694905
```

```
f1_score(y_test, LR_yhat, average='weighted')
```

```
0.6891715877694646
```

4. Results

None of the algorithms implemented above gave an accuracy score equal to or greater than 0.71, they all ranged from 0.6 to 0.7. Meaning, these models can predict the severity code of an accident with an accuracy equaling 60-70%. A bar plot is plotted below with the bars representing the accuracy of each model in descending order respectively.



Clearly, Kernel Support Vector Machine is the best classifier for this classification problem based on accuracy, followed by the Logistic Regression model.

5. Conclusion

The accuracy of the classifiers is not great, highest being 70%. This usually means that the model is under fitted i.e. it needs to be trained on more data. Though the dataset has a lot of variety in terms of scenarios, more volume of the data for such scenarios has to be collected.

Certain features with missing values were removed, this reduced the dimensionality of the dataset, these features could have been correlated to other important features but they had to be removed. A better effort has to be made to collect data to reduce the number of missing values.

6. Discussion

As mentioned above, the amount of data available to train the above mentioned models is not sufficient and it does not seem to have enough data of all varieties. Hence, integrating Cross Validation methods with hyper parameter model would help in training and possibly increase the accuracy of every classification model.