Accident Severity Prediction

Introduction/ Business Problem

The Washington State Department of Transportation Crash Data Portal provides crash information for accidents that occurred state-wide. According to the 2019 data, there were 45,524 accidents on all roads. Of those:

- 235 were fatal crashes
- 973 were suspected of serious injury accidents
- 2,798 were suspected of minor injury accidents
- 9,412 were possible injury crashes
- 32,106 were no apparent injury collisions

Our motivation is to use the weather, location and road condition data provided in the dataset, made available by the Seattle Department of Transportation Traffic Management Division, to arrive at a correlation to predict the severity of road accidents. This tool/data can then be made available to the public and the Seattle traffic authorities to possibly prevent/reduce severe or fatal accidents in the future by taking precautionary measures.

Data Understanding

We chose the unbalanced dataset provided by the Seattle Department of Transportation Traffic Management Division with 194673 rows (accidents) and 37 columns (features) where each accident is given a severity code. It covers accidents from January 2004 to May 2020. Some of the features in this dataset include and are not limited to Severity code, Location/Address of accident, Weather condition at the incident site, Driver state (whether under influence or not), collision type. Hence we think its a good generalized dataset which will help us in creating an accurate predictive model.

The unbalance with respect to the severity code in the dataset is as follows.

SEVERITY CODE	Count
1	136485
2	58188

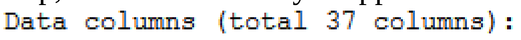
Data Pre-processing

An unbalanced dataset is used, provided by the Seattle Department of Transportation Traffic Management Division with 194673 rows (accidents) and 37 columns (features) where each accident is given a severity code. The steps taken in pre-processing the dataset are as follows.

1. **Removal of irrelevant columns or features**

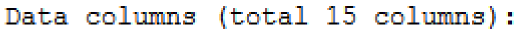
   Columns containing descriptions and identification numbers that would not help in the classification are dropped from the dataset to reduce the complexity and dimensionality of the dataset. 'OBJECTID', 'INCKEY', 'COLDETKEY', 'REPORTNO', 'STATUS', 'INTKEY', 'EXCEPTRSNCODE' and more belong to this category. Certain other categorical features were removed as they had a large number of distinct values, example: ‘LOCATION’.

After performing this step, the dimensionality dropped from 37 to 18.



```
Data columns (total 37 columns):
```

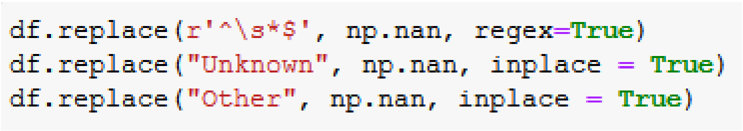
|
|
V



```
Data columns (total 15 columns):
```

2. Identification and handling missing values

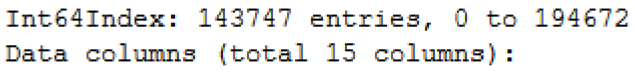
To identify columns and rows with missing values is the next step. Empty boxes, ‘Unknown’ and ‘Other’ were values considered as missing values. These were replaced with NA to make the dataset uniform.



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

Columns ("INATTENTIONIND","PEDROWNOTGRNT","SPEEDING") which had more than 20% of its values missing were noted down and were dropped. For columns ("X","Y","COLLISIONTYPE","JUNCTIONTYPE"….) which had less than 20% of its values missing, the respective rows were removed since most of the columns in this dataset are categorical type, goal was to not impute the non-numerical columns; hence it did not make sense to replace the values.

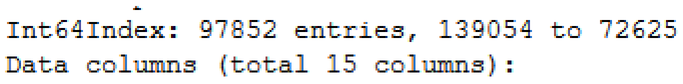
Once the above two strategies were performed, the dataset reduced from having 194673 rows and 15 columns to having 143747 rows and 15 columns.



```
Int64Index: 143747 entries, 0 to 194672
Data columns (total 15 columns):
```

3. Balancing the dataset

With the above two pre-processing steps complete, a dataset (143747 rows) with 94821 rows for severity code 1 and 48926 rows for severity code 2 is obtained. Training an algorithm on an unbalanced dataset w.r.t the target category will result in a biased model. The model will have learnt more about one the category that has more data. In order to prevent this, a new balanced dataset (97852 rows) is created by randomly sampling out 48926 rows with severity code 2 and then concatenating it with 48926 rows with severity code 1. The dataset is then shuffled to randomize the rows.

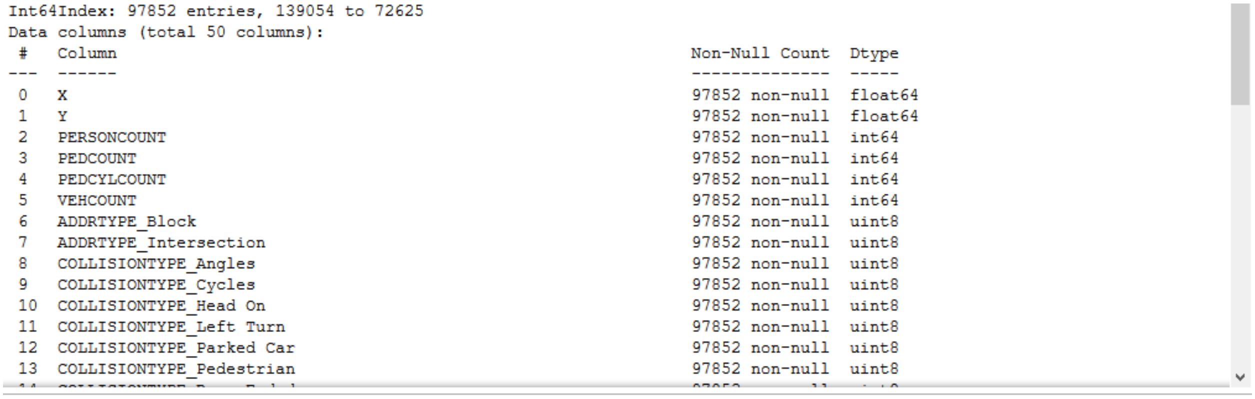


```
Int64Index: 97852 entries, 139054 to 72625
Data columns (total 15 columns):
```

4. Encoding of data

The dataset is split into two datasets, X and Y, where Y contains the target feature (SEVERITYCODE) and X contains all the independent features/variables.

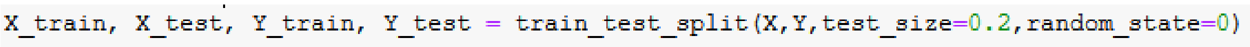
Machine Learning models are trained only on numerical data; hence all categorical features in the dataset have to be encoded so that the algorithms can be trained on those features. The ‘get_dummies’ method from pandas library is used to convert/encode each and every categorical feature. After application, number of features in dataset X increased from 14 to 50.



```
Int64Index: 97852 entries, 139054 to 72625
Data columns (total 50 columns):
 #   Column                                                             Non-Null Count  Dtype
---  ------                                                             --------------  -----
 0   X                                                                  97852 non-null  float64
 1   Y                                                                  97852 non-null  float64
 2   PERSONCOUNT                                                        97852 non-null  int64
 3   PEDCOUNT                                                           97852 non-null  int64
 4   PEDCYLCOUNT                                                        97852 non-null  int64
 5   VEHCOUNT                                                           97852 non-null  int64
 6   ADDRTYPE_Block                                                     97852 non-null  uint8
 7   ADDRTYPE_Intersection                                              97852 non-null  uint8
 8   COLLISIONTYPE_Angles                                               97852 non-null  uint8
 9   COLLISIONTYPE_Cycles                                               97852 non-null  uint8
 10  COLLISIONTYPE_Head On                                              97852 non-null  uint8
 11  COLLISIONTYPE_Left Turn                                            97852 non-null  uint8
 12  COLLISIONTYPE_Parked Car                                           97852 non-null  uint8
 13  COLLISIONTYPE_Pedestrian                                           97852 non-null  uint8
```

5. **Splitting into training and testing datasets**

The datasets X and Y are 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=0)
```

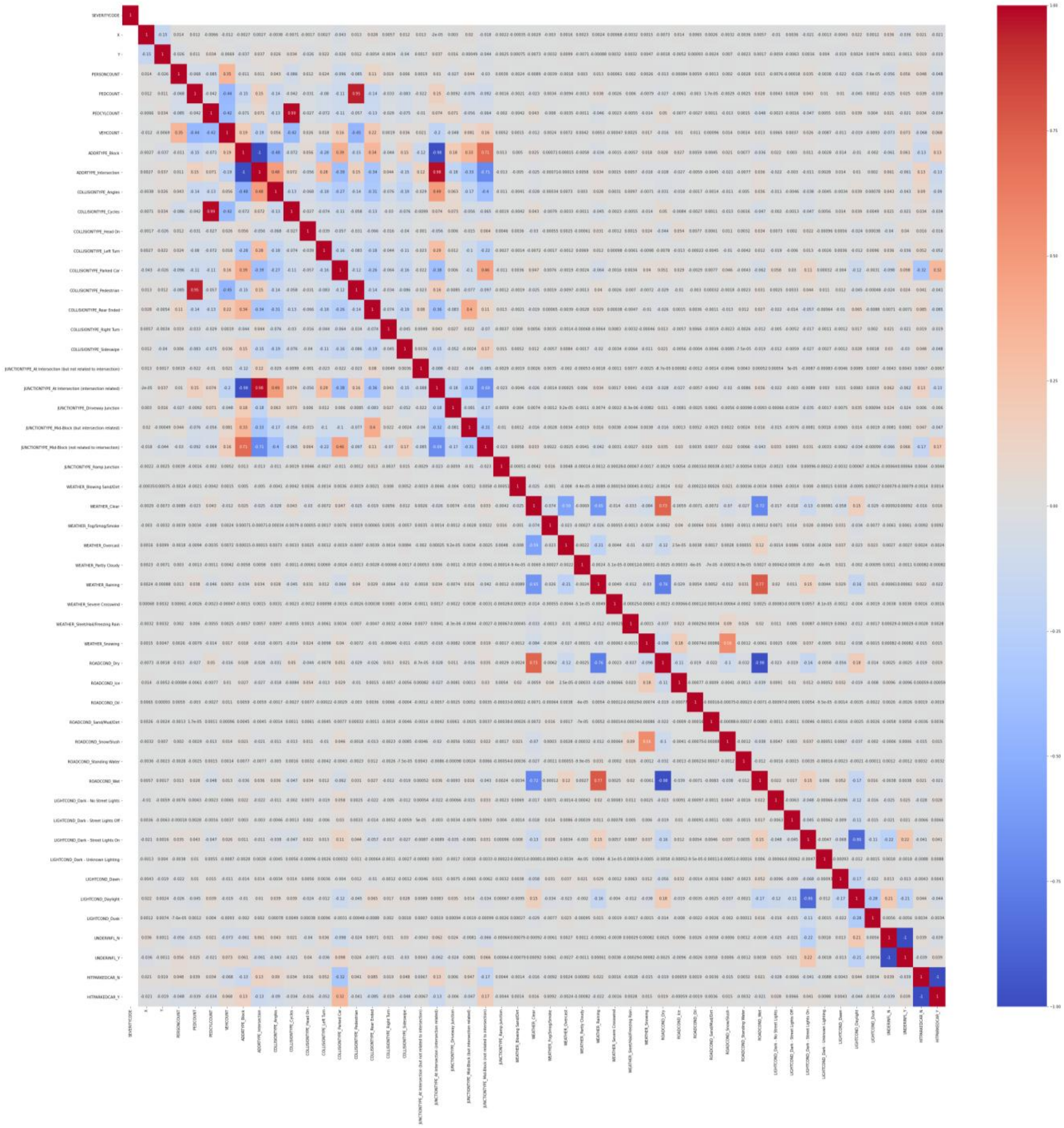
6. **Normalizing/ Feature scaling of data**

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.

Understanding Correlation in Dataset

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

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.



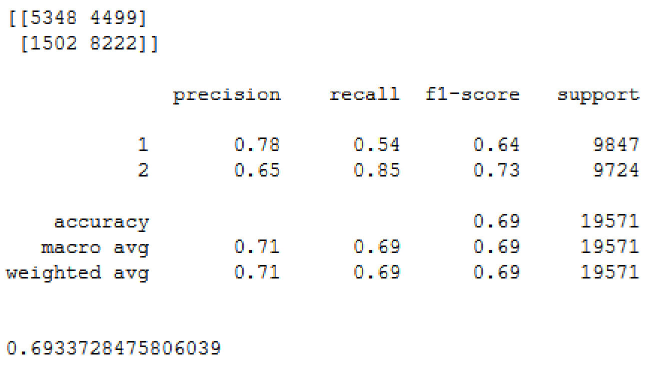
Machine Learning Algorithms

A total of six 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.

- **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 values obtained would always lie within 0 and 1 since it predicts the probability.

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 results, confusion matrix, classification report and accuracy, are:



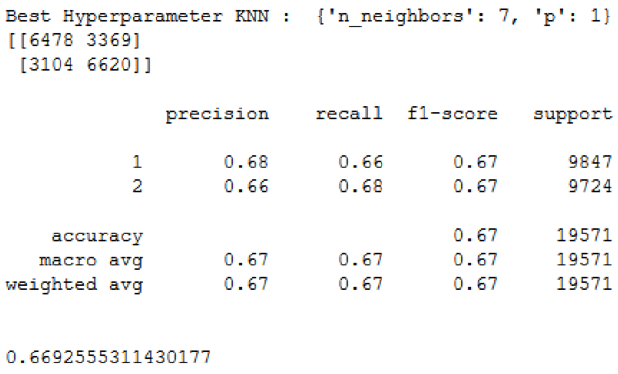
```
[[5348 4499]
 [1502 8222]]

              precision    recall  f1-score   support

           1       0.78      0.54      0.64      9847
           2       0.65      0.85      0.73      9724

    accuracy                           0.69     19571
   macro avg       0.71      0.69      0.69     19571
weighted avg       0.71      0.69      0.69     19571


0.6933728475806039
```

- **K Nearest Neighbours Classifier**

K nearest neighbours 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 neighbours. The case assigned to the class is most common amongst its K nearest neighbours measured by a distance function (Euclidean, Manhattan, Minkowski, and Hamming).

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

The results, confusion matrix, classification report and accuracy, are:

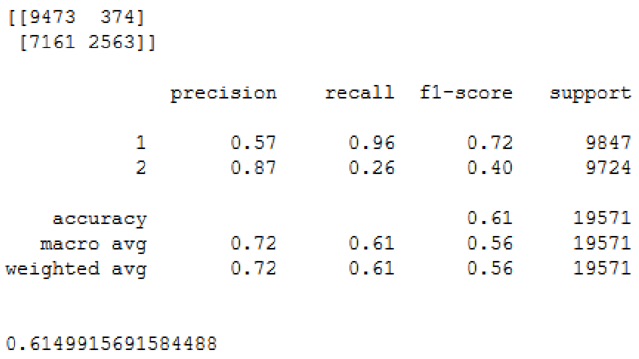


```
Best Hyperparameter KNN :  {'n_neighbors': 7, 'p': 1}
[[6478 3369]
 [3104 6620]]

              precision    recall  f1-score   support

           1       0.68      0.66      0.67      9847
           2       0.66      0.68      0.67      9724

    accuracy                           0.67     19571
   macro avg       0.67      0.67      0.67     19571
weighted avg       0.67      0.67      0.67     19571


0.6692555311430177
```

- **Naïve Bayes Classifier**

Naive Bayes classifies objects based on Bayes' Theorem with an assumption that the predictors (features) are independent of each other. Bayes theorem is a way to calculate posterior probability P(c|x) from the P(c), P(x), P(x|c). Naive Bayes is naive because it assumes the presence of a particular feature is completely unrelated to the presence of another, and each of them contributes to the posterior probability independently.

The results, confusion matrix, classification report and accuracy, when Naïve Bayes was applied to the pre-processed accident severity dataset are:



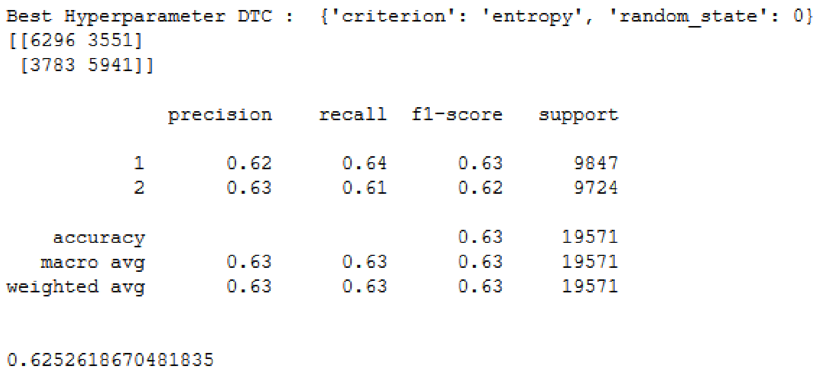
```
[[9473  374]
 [7161 2563]]
              precision    recall  f1-score   support

           1       0.57      0.96      0.72      9847
           2       0.87      0.26      0.40      9724

    accuracy                           0.61     19571
   macro avg       0.72      0.61      0.56     19571
weighted avg       0.72      0.61      0.56     19571

0.6149915691584488
```

- **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 cut-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 maximum 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 results, confusion matrix, classification report and accuracy, are:



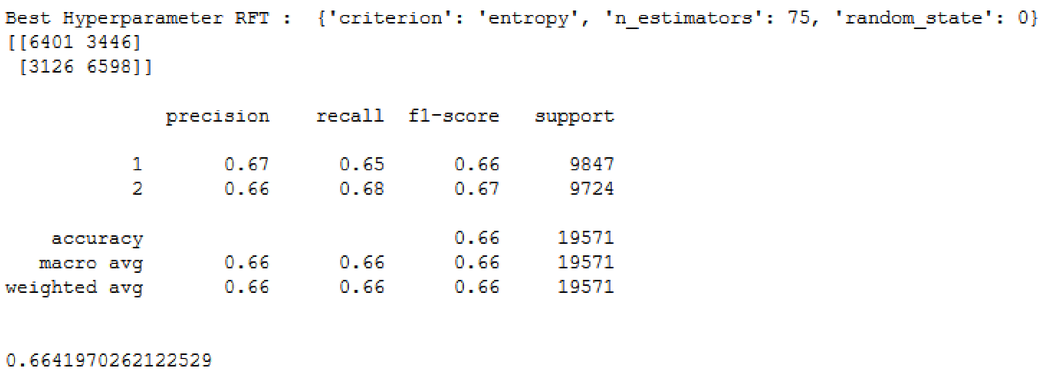
```
Best Hyperparameter DTC :  {'criterion': 'entropy', 'random_state': 0}
[[6296 3551]
 [3783 5941]]
              precision    recall  f1-score   support

           1       0.62      0.64      0.63      9847
           2       0.63      0.61      0.62      9724

    accuracy                           0.63     19571
   macro avg       0.63      0.63      0.63     19571
weighted avg       0.63      0.63      0.63     19571

0.6252618670481835
```

- **Random Forest Tree Classifier**

  Random Forest Classifier is an ensemble (algorithms which combines more than one algorithms of same or different kind for classifying objects) tree-based learning algorithm. RFC is a set of decision trees from randomly selected subset of training set. It aggregates the votes from different decision trees to decide the final class of the test object. Used for both classification and regression.

  Similar to DTC, RFT requires an input that specifies a measure that is to be used for classification, along with that a value for the number of estimators (number of decision trees) is required. A hyper parameter RFT was used to determine the best choices for the above mentioned parameters. RFT with 75 DT's using entropy as the measure gave the best accuracy when trained and tested on pre-processed accident severity dataset.

The results, confusion matrix, classification report and accuracy, are:



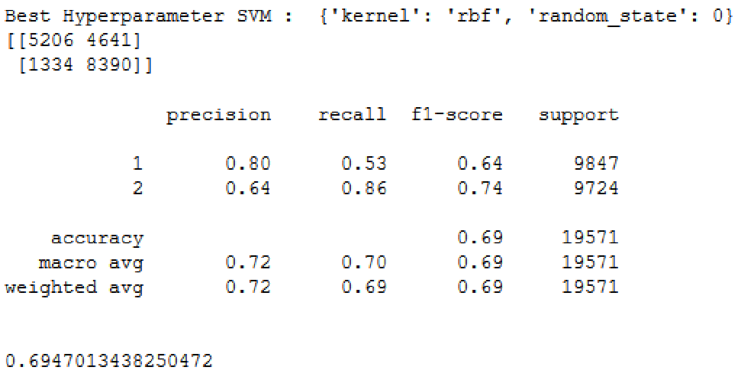
```
Best Hyperparameter RFT :  {'criterion': 'entropy', 'n_estimators': 75, 'random_state': 0}
[[6401 3446]
 [3126 6598]]

              precision    recall  f1-score   support

           1       0.67      0.65      0.66      9847
           2       0.66      0.68      0.67      9724

    accuracy                           0.66     19571
   macro avg       0.66      0.66      0.66     19571
weighted avg       0.66      0.66      0.66     19571


0.6641970262122529
```

- **Support Vector Machine Classifier**

  Support Vector Machine is an algorithm which can be used for both classification and regression challenges. However, it is mostly used in classification problems. 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 classification.

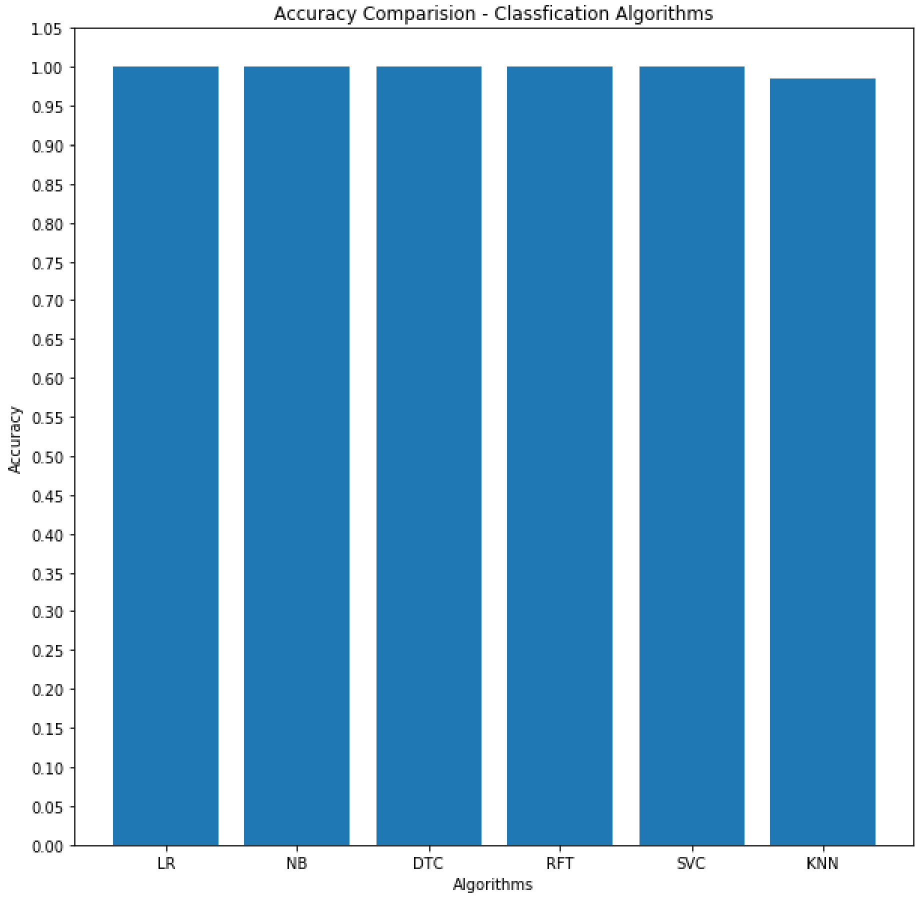
  The results, confusion matrix, classification report and accuracy, are:



```
Best Hyperparameter SVM :  {'kernel': 'rbf', 'random_state': 0}
[[5206 4641]
 [1334 8390]]

              precision    recall  f1-score   support

           1       0.80      0.53      0.64      9847
           2       0.64      0.86      0.74      9724

    accuracy                           0.69     19571
   macro avg       0.72      0.70      0.69     19571
weighted avg       0.72      0.69      0.69     19571


0.6947013438250472
```

Results

None of the algorithms implemented above gave an accuracy score equal to or greater than 0.7, they all ranged from 0.6 to 0.7. Meaning, these models can predict the severity code of an accident with an accuracy equalling 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.

Conclusion

The accuracy of the classifiers is not great, highest being 69%. 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.

Future Work

As mentioned above, the amount of data available to train the above mentioned models in 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.