Accident Severity Prediction Report

Created by: Mohammed Abdul Qayyum Khan

Business Problem

The key objective of this project is to predict the severity of accidents in the Seattle area using the historical data provided by the Seattle Police Department (SPD). Road accidents are a major problem causing thousands of deaths and even greater number of injuries each year. They also cause significant amount of property damage. They are especially common in densely populated urban cities. Due to the rapid urbanization, the rate of road traffic accidents has been continually increasing each year. There are several reasons which contribute towards accidents such as conditions of roads, weather conditions, traffic situations etc. By performing machine learning, we try to predict the severity of road accidents which will help us identify the major contributing factors behind them. This information can then be used by the authorities to take appropriate preventive measures that will reduce the risk of accidents. Individual commuters can also utilize this information to plan their travel accordingly.

Data

For our project, we utilize the road collisions data in Seattle region provided by the Seattle Police Department. The scope of our data is from the year 2004 to present. We have a sufficiently large dataset consisting of 194673 records along with a wide variety of 38 attributes. Some of the major attributes in the data include location information, weather conditions, road conditions, along with the date and time of the incident. We use the 'SEVERITY CODE' column in our data as the target label for our modelling. Severity code of 1 corresponds to accidents involving only property damage whereas a value of 2 corresponds to an injury. We access the richness of our data with an exploratory data analysis using visualization techniques.

Methodology

a. Data Preparation

The dataset used is the road traffic collisions data acquired from the Seattle PD. The data is available from the year 2004 to present in a CSV file format. It consists of 38 columns and over 190,000 rows.

We use 'SEVERITY CODE' column as the target label for our modelling. Upon further analysis of the data, I found that that the SEVERITY CODE column is duplicated and there is also 'SEVERITY DESC' column that provides a description of the accident. We cannot use these two features in our training as they cause target leakage. Target leakage occurs when our features include data which will not be available at the time of actual prediction. After dropping these two columns, I extracted our target label into a separate dataframe.

I also dropped the arbitrary key columns from the dataset as they are just unique identifiers and do not actually offer any meaningful value to our data. Although this is optional, I have done it for the sake of performance improvements.

In the target column, there are two possible values for severity code:1 and 2. Since this is a binary class classification problem, I replaced the severity code of 1 to 0 in order to indicate low severity. Similarly, I replaced the severity code 2 with a value of 1 to indicate high severity.

b. Data Cleaning

In this step, I performed several data cleansing tasks to prepare our data for modelling. The richness of our data directly determines the accuracy of our model.

There are two features in our dataset - EXCEPTRSNCODE, EXCEPTRSNDESC which have significant amount of blank values and rest of the values are labelled as 'Not Enough Information'. This indicates that these two features don't contain any meaningful information and can be be removed from our data.

There are multiple binary valued features in our data of 'Yes/No' type values. These are 'SPEEDING', 'PEDROWNOTGRNT', 'INATTENTIONIND', 'HITPARKEDCAR'. I converted these into boolean type values by replacing 'Yes' values with 1 and 'No' values with 0.

Machine learning algorithms do not work with NULL/NaN values in data. Upon plotting a bar chart of the percentage of NULLs in each feature, I identified that few of our features have NaN values.The highest % of NULLs is in the 'JUNCTION TYPE' feature at 3%. We need to impute these missing values with either an arbitrary or calculated value. For non-numeric features, I replaced the NULLs with a value 'Unknown' and for numeric features, the value was 0.

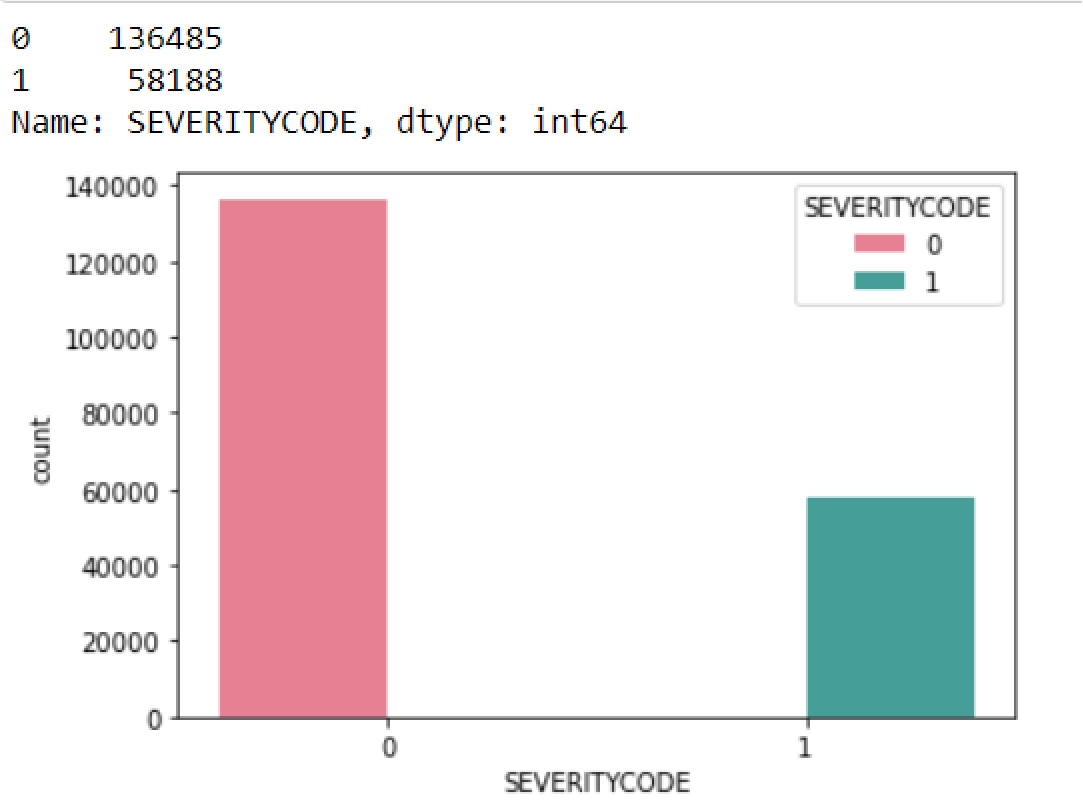
c. Feature Engineering

This step involves extracting new features out of existing features based on our understanding of the data. Using the 'INCDTTM' feature which indicates the date and time of the incident, I extracted several date-time features such as day, week, month, year, hour, minute etc. We add theses as new features to our dataset.

We also need to convert the categorical features into numeric values as most machine learning algorithms do not work with non-numeric data. Using the label encoder from the Scikit-learn library assigns a numerical value to each category of values.

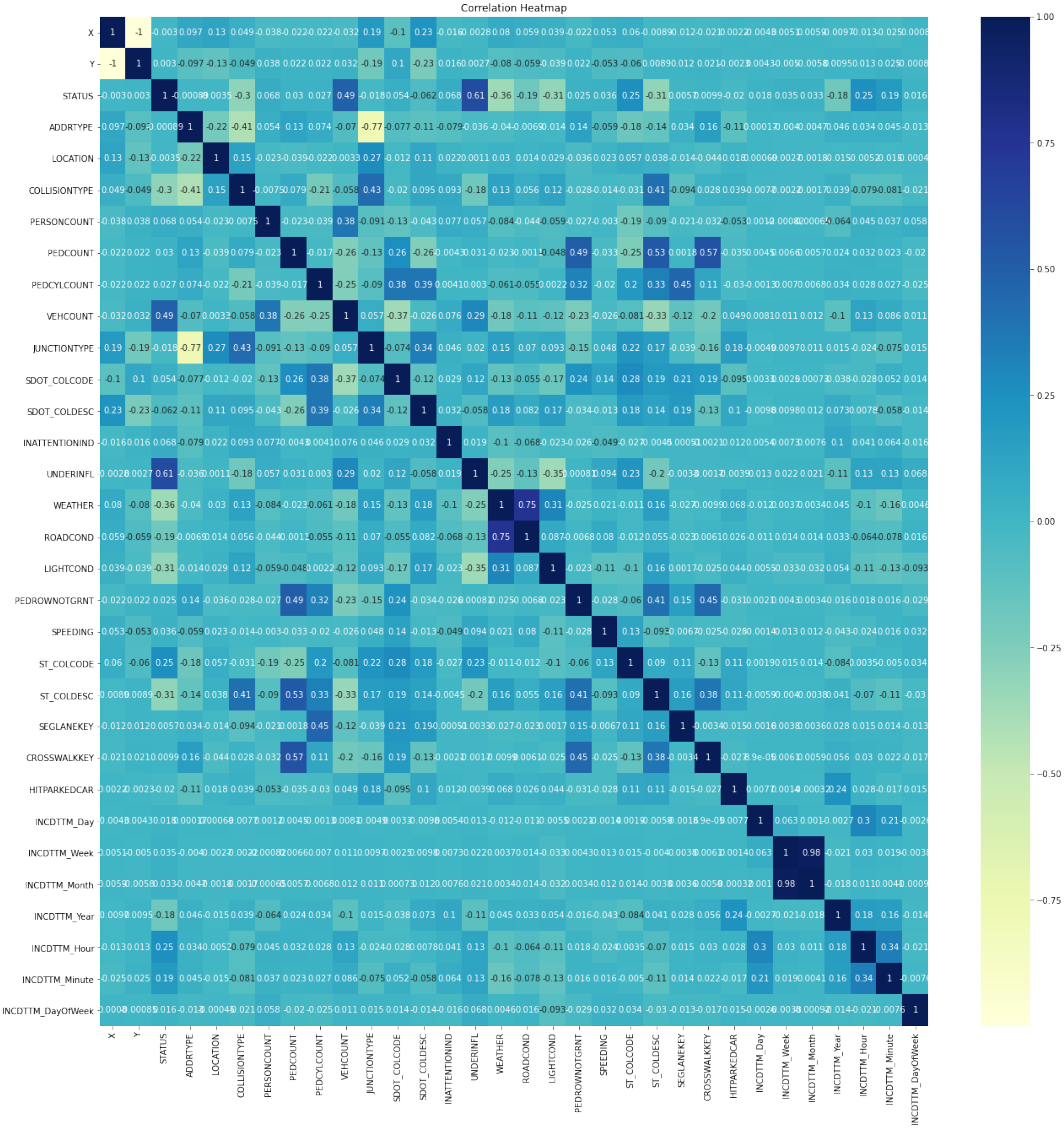
d. Exploratory Data Analysis

Upon visualizing a count plot of our target label to identify the class imbalance, we see that 70% of our records have target value of 0 (minority class) and the rest 30% are values of 1 (minority). Due to this large imbalance in the distribution of our target values, I resampled the training data using SMOTE oversampling function. SMOTE is a popular up-sampling technique that balances class distribution by randomly increasing minority class examples by replicating them.



```
0    136485
1     58188
Name: SEVERITYCODE, dtype: int64
```

Visualizing a correlation heat-map of our features, we identify several pair of features with correlations between them. Features 'WEATHER' and 'ROADCOND' have a strong correlation between them which is expected as weather conditions usually contribute to the condition of roads. Severe weather conditions can adversely impact the condition of roads.



e. Modelling

Now that the dataset is prepared, it is ready for modelling. I used different types of machine learning algorithms and compared the results. I did a random split of 25% data to be used for testing and the remaining 75% used for training. Non-tree based algorithms require the features to be normalized to represent them on a same scale. Tree based models don't rely on feature scaling. I observed that non-tree based models usually run for longer compared to the rest.

Algorithms used -

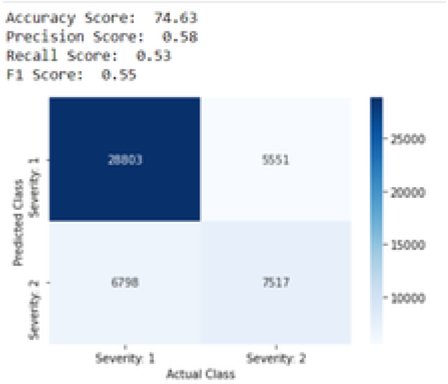
1. Decision Tree
2. Random Forest
3. XGBoost
4. Logistic Regressions

5. Support Vector Machines
6. K-Nearest Neighbours

Results

The best results were achieved with the **XGBoost classifier with an accuracy of 74.63% and F1 Score of 0.55.**

XGBoost Detailed Results:

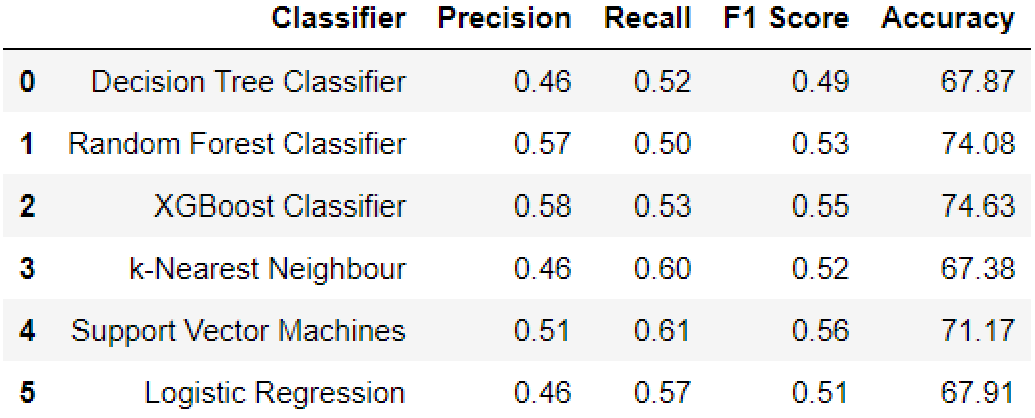


```
Accuracy Score:  74.63
Precision Score:  0.58
Recall Score:  0.53
F1 Score:  0.55
```

As we can see, the model did better in correctly predicting severity 1 compared to severity 2 incidents.
The top five features of our this XGBoost model are -

1. JUNCTIONTYPE
2. ST_COLDDESC
3. COLLISIONDESC
4. ST_COLCODE
5. ROADCOND

Comparison of different algorithms:



	Classifier	Precision	Recall	F1 Score	Accuracy
0	Decision Tree Classifier	0.46	0.52	0.49	67.87
1	Random Forest Classifier	0.57	0.50	0.53	74.08
2	XGBoost Classifier	0.58	0.53	0.55	74.63
3	k-Nearest Neighbour	0.46	0.60	0.52	67.38
4	Support Vector Machines	0.51	0.61	0.56	71.17
5	Logistic Regression	0.46	0.57	0.51	67.91

Conclusion

In this project, I analyzed the road traffic data for the Seattle area and applied machine learning in order to predict the severity of the accidents and also to identify the major factors influencing the severity of an accident. I started with some pre-processing steps such as cleaning our data to improve its quality. We used encoding techniques to convert all of our categorical data into numeric values. Since our target class was highly imbalanced, we used SMOTE up-sampling to balance our training data. We ran several popular classification algorithms and derived their evaluation metrics to identify the best performing models.

These models will help the road transport authorities to take appropriate actions that will reduce the risk of accidents. For individual commuters, they can plan their travel accordingly, such as based on the weather forecast as it is strongly correlated to the conditions of the road.

As part of the future steps, there are several ways in which we can improve the results of our models. One way to do this, is to get more relevant data by adding more attributes to our dataset. Another good way to improve our model results is by performing hyper-parameter tuning as we ran our models with their default parameters.