IST 718 – Big Data Analytics

US ACCIDENTS

Group 5
Hitesh Chandrakumar Thadhani
Nishitha Maniganahalli Venkatesh
Sakshi Raghuvanshi
Tanishk Parihar

ABSTRACT

Problem Statement

On an average around 37 thousand people die of road accidents in US and reducing road accidents is an important public safety challenge. Accidents can be reduced by identifying accident hotspot locations. Occurrence of accidents can be predicted based on the weather conditions, temperature, location, weekday etc. Using our analysis, we plan to find out the most important factors that cause accident. Highway assistance for casualties is a major task during accidents, which can be improved by casualty analysis. Another important outcome during any road accident is severity of traffic which depends on the severity of accidents. So, using the dataset we have decided to predict the severity of the accidents.

Dataset

The dataset is car accidents data for USA with data collected from Feb 2016 to December 2019. The dataset is taken from Kaggle (US car Accidents Feb16 Dec19). It has 2.9 million observations and 49 attributes with both numerical and categorical data. The dataset is for 49 contiguous US states collected through multiple API including 2 APIs which stream real time traffic data. The features can be categorized into geographical features, weather, date-time, POI (point of interest) annotation.

Proposed techniques and data science methodology

The methodology planned to be followed to solve the problem consists of different stages. The stages include Business Understanding, Data Preprocessing, Exploratory Data Analysis, Feature Engineering, Data Modeling, Model Evaluation, Recommendations and Insights. Business understanding and data preprocessing tasks comprise of data cleaning, imputation of null values using mean/median, transforming data into meaningful variables or combination of variables. Feature engineering to select the most important features in the dataset. Data modeling includes splitting data into train and validation, running different algorithms such as Decision Tree, Random Forest, Gradient Boost and tuning hyperparameters to build best model. Model evaluation using different metrics such as accuracy, precision, recall, F1 score, AUC/ROC curve to select the best

model. Recommendations and insights derived from exploratory analysis and models built to be implemented to solve business problems.

Business Questions intended to be analyzed

We can help the government and transport department by providing insights from our analysis to devise strategies for reducing the number of accidents.

- Which State and City recorded the maximum number of accidents?
- Which month or year had how many accidents and if there is a trend or a pattern?
- Relationship of environmental factors like weather contributing to the accident.
- Relationship of accident severity with the place it occurred for Ex: traffic signal or a junction.
- Which time of the day has most accidents?
- Which week of the year has most accidents?

By knowing the states and cities with the maximum number of accidents, government can take some to reduce the local accidents in these states and cities. The time when the accidents happen like for a particular hour, month and some of the weather conditions would be helpful for both the drivers to be more cautious about the driving.

INTRODUCTION

DATA DESCRIPTION

The dataset has 2.9 million observations and 49 observations, out of which we will be using a subset of around 1million observations. The important columns are listed below with their description.

- Source: Indicates source of the accident report (i.e., the API which reported the accident)
- TMC: A traffic accident may have a Traffic Message Channel (TMC) code which provides more detailed description of the event
- Severity: Shows the severity of the accident, a number between 1 and 4, where 1 indicates the least impact on traffic (i.e., short delay as a result of the accident) and 4 indicates a significant impact on traffic (i.e., long delay)
- Start_Time: Shows start time of the accident in local time zone
- End_Time: Shows end time of the accident in local time zone
- Distance: The length of the road extent affected by the accident (in miles)
- Temperature: Shows the temperature (in Fahrenheit)
- Humidity: Shows the humidity (in percentage)
- Pressure: Shows the air pressure (in inches)
- Visibility: Shows visibility (in miles)
- Wind_Direction: Shows wind direction
- Wind_Speed: Shows wind speed (in miles per hour)
- Precipitation: Shows precipitation amount in inches, if there is any
- Weather_Condition: Shows the weather condition (rain, snow, thunderstorm, fog, etc.)
- Amenity: A POI annotation which indicates presence of amenity in a nearby location
- Bump: A POI annotation which indicates presence of speed bump or hump in a nearby location
- Crossing: A POI annotation which indicates presence of crossing in a nearby location
- Junction: A POI annotation which indicates presence of junction in a nearby location

- No_Exit: A POI annotation which indicates presence of no exit in a nearby location
- Railway: A POI annotation which indicates presence of railway in a nearby location
- Roundabout: A POI annotation which indicates presence of roundabout in a nearby location
- Station: A POI annotation which indicates presence of station in a nearby location
- Stop: A POI annotation which indicates presence of stop in a nearby location
- Traffic_Calming: A POI annotation which indicates presence of traffic calming in a nearby location
- Traffic_Signal: A POI annotation which indicates presence of traffic signal in a nearby location
- Sunrise_Sunset: Shows the period of day (i.e. day or night) based on sunrise/sunset
- Civil_Twilight: Shows the period of day (i.e. day or night) based on civil twilight
- Nautical_Twilight: Shows the period of day (i.e. day or night) based on nautical twilight
- Astronomical_Twilight: Shows the period of day (i.e. day or night) based on astronomical twilight

DATA PREPROCESSING

For our analysis and modeling, we considered a subset of nearly 1 million observations. The first step in data cleaning was removing unwanted columns from the data which would not be helpful for the analysis. We decided to remove features which had unique values for all rows or a single unique value for all the entire column. Columns which had lots of missing values were also dropped.

- **Outlier detection and handling**

  Weather related data had extreme outliers which are practically not possible. In such cases the outlier values were clipped to the minimum or maximum thresholds. We used various exploratory analysis to find outliers and depending on that we handled our outliers.

- **Dealing with null values**

  Our dataset contained null values in most of the columns. In case of categorical and numerical attributes, missing values were replaced by mode and median of the columns respectively. For Precipitation, null values were replaced with zero assuming the fact that Precipitation values were missing for those instances when there was no rainfall. Null values for some columns were removed by dropping the entire row. It would have been misleading to impute these values so we decided to drop it.

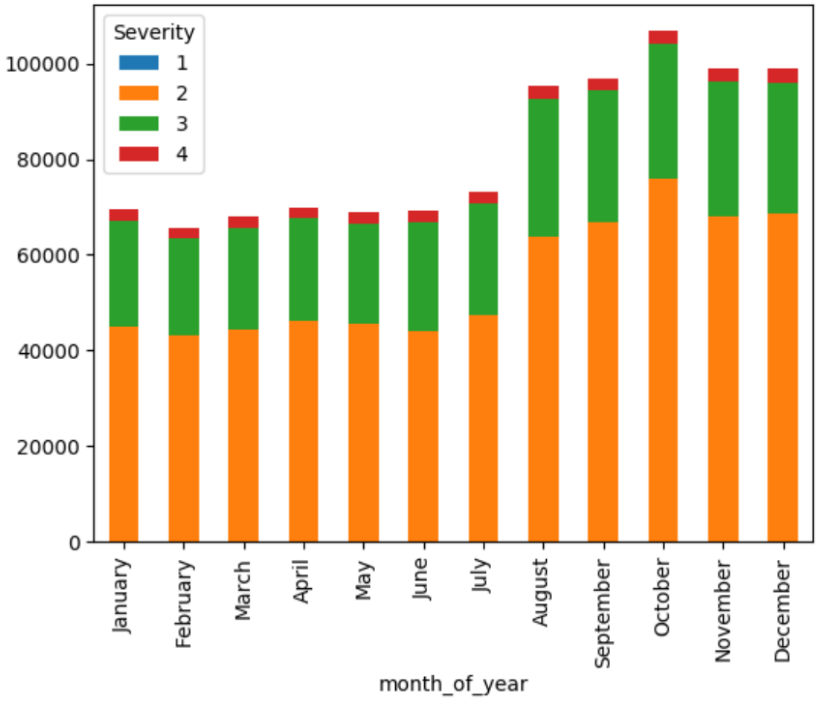
- **Handling duplicate values**

  Categorical attribute like Wind_Direction had duplicate values like ‘North’ and ‘N’. The duplicates were handled by keeping the single character representation and replacing the other. The same technique was followed in case of Weather_Condition where ‘Light Rain Shower’ and ‘Light Rain Showers’ meant the same.

DESCRIPTIVE STATISTICS

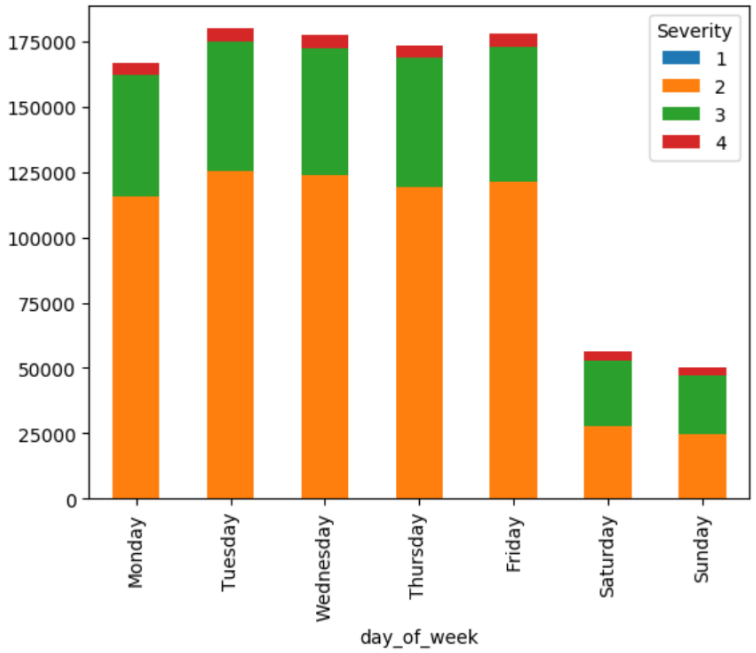
We performed Exploratory Data Analysis and used those Visualizations to find insights about the Accident dataset.

- **Monthly Accident count showing how distribution of accidents on 12 months of the year**



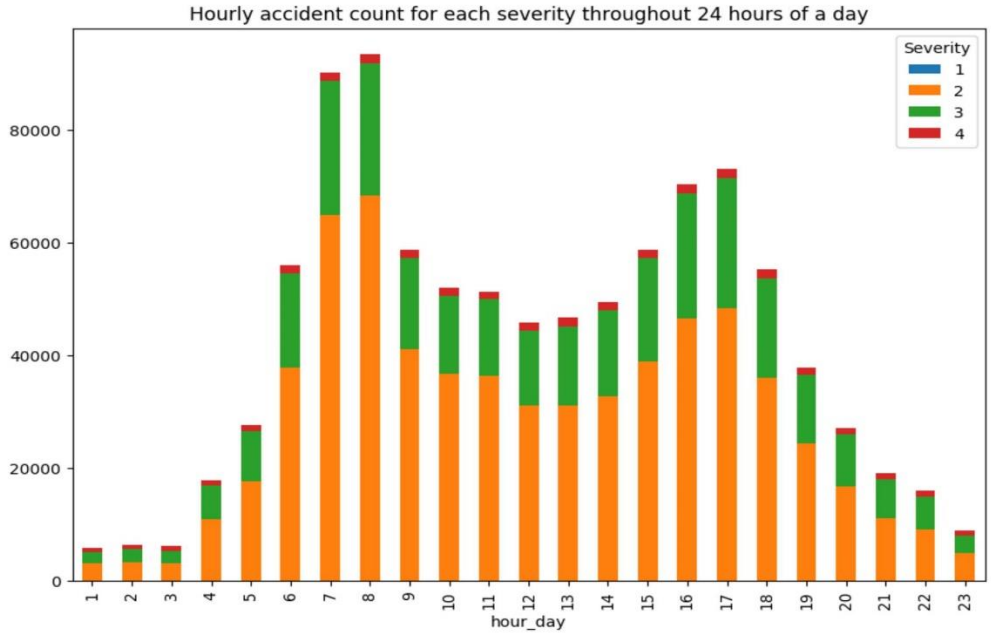
The above bar chart shows the count of accidents throughout the year. It shows that the maximum number of accidents happen in the month of October which are mostly of Severity 2. Also, we can see a spike in number of accidents between the month of August and December.

- **Daily Accident count showing how distribution of accidents on 7 days of the week**



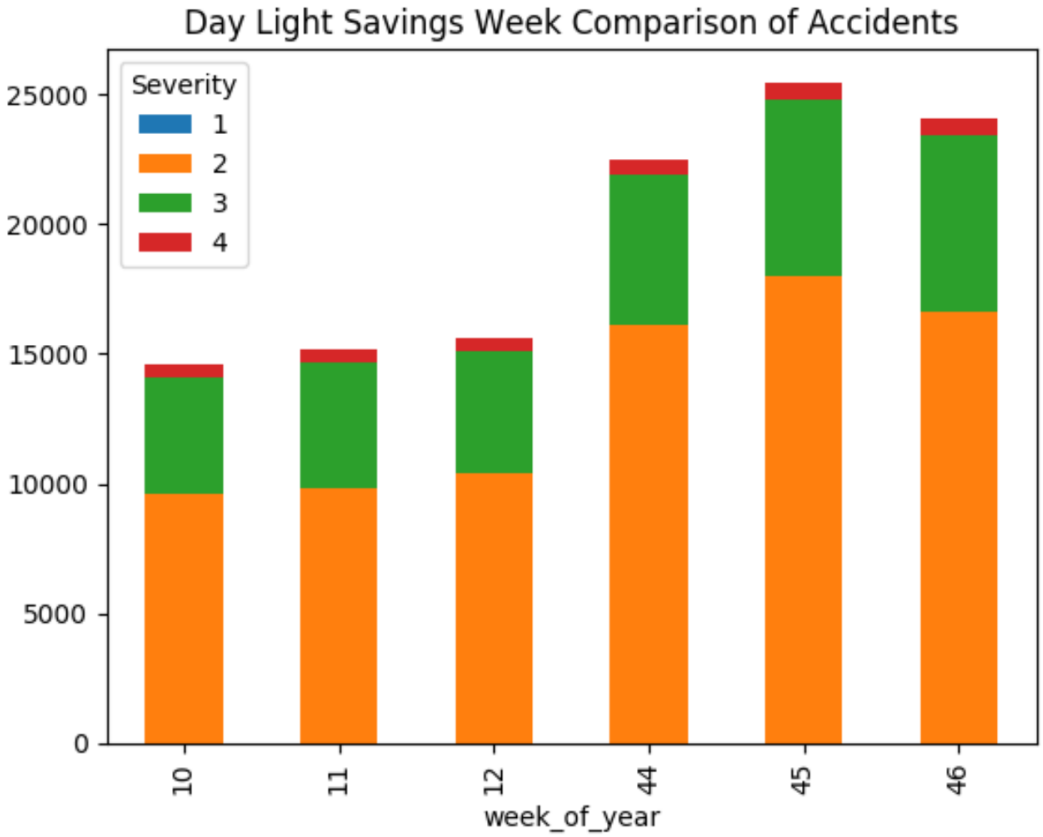
The above bar chart shows that the number of accidents for each weekday. From our graph, we can infer that the number of accidents drastically decrease on weekends.

- **Accident count showing how distribution of accidents on each hour days of the day for the whole 24 hours**



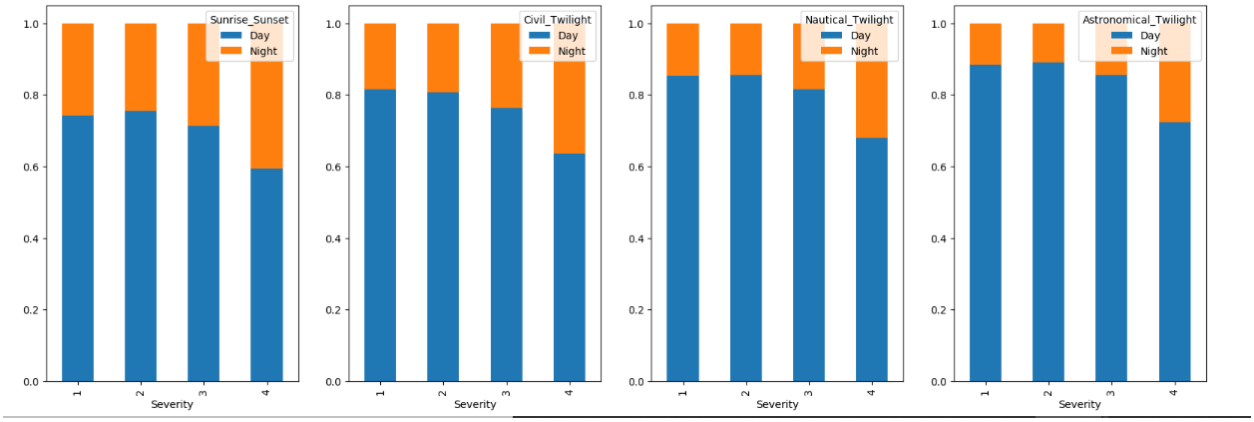
The above bar chart tells us that most number of accidents happen at the 7th and 8th hour of a day, which is basically 7 AM and 8 AM, which is not surprising as those are the hours at which people are rushing to reach work on time. Also, similar pattern can be seen around 4 PM and 5 PM.

- **For Day light Savings checking**



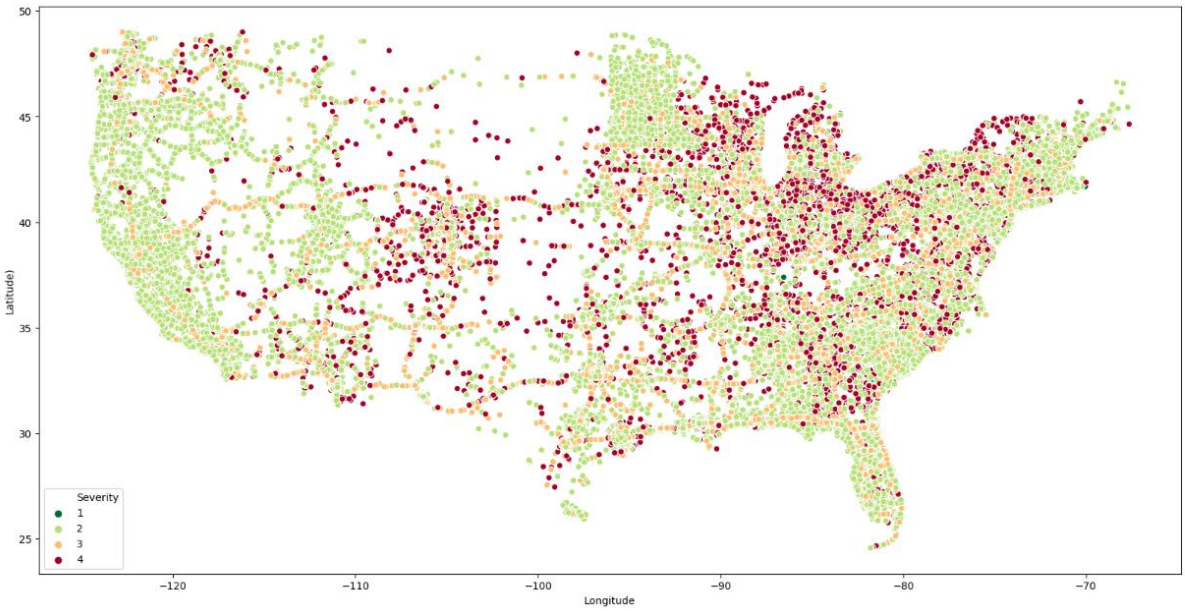
The above bar chart shows us that as compared to the 10th week, number of accidents rose for week 11. The reason being, that week 11 is mostly the week where day light savings start and in some particular year it is Week 12. This shows around 6% increase in the number of accidents for that week from previous week. Same goes for the 44th and the 45th week, as 45th week is the day light saving end week and it shows a sharp rise in the number of accidents.

- **Severity as a percentage on y axis showing how severe the accidents can be considering the various variables**



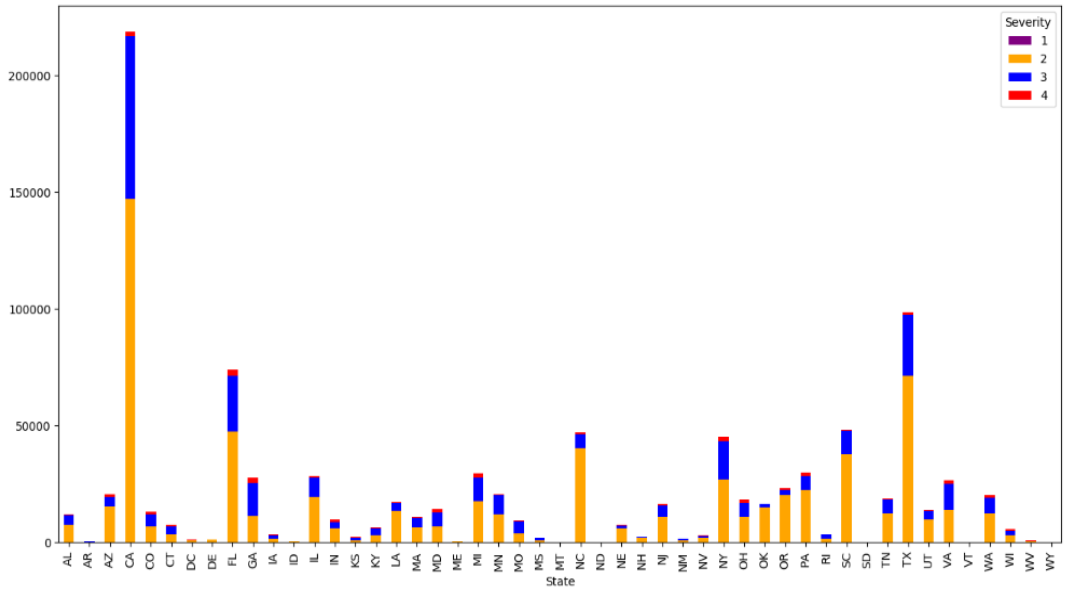
The above bar charts show accidents that take place during nighttime have more severity than the ones in the day.

- **Scatter plot of accidents on the US map where each point represents an accident**



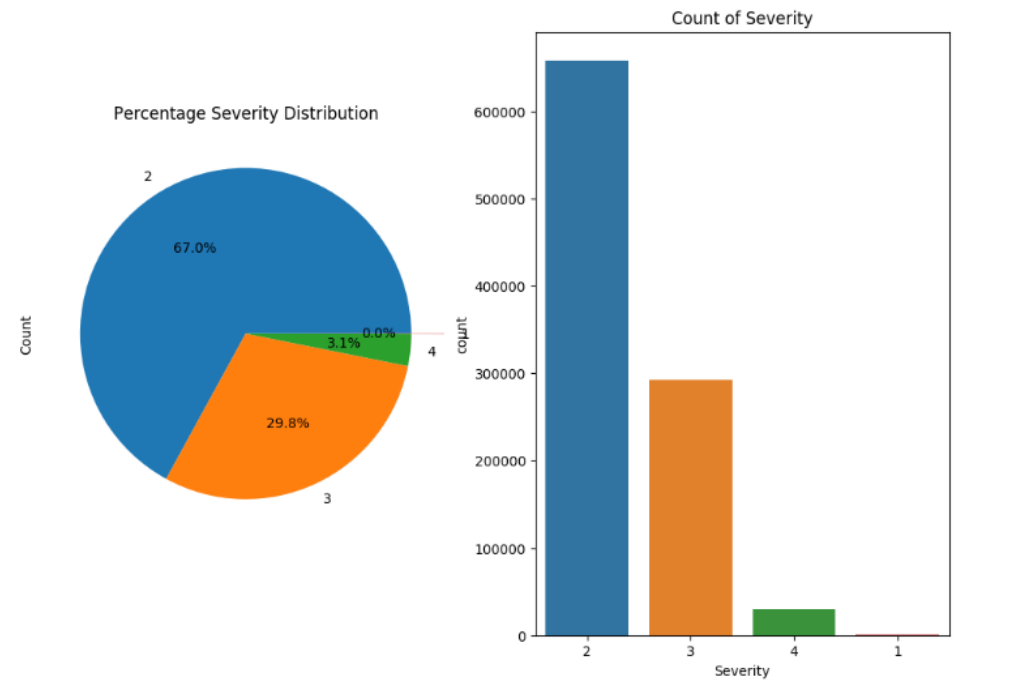
The above scatterplot shows the distribution of accidents across US. We can see that the number in the mid-west part is lower compared to other parts. Also, number of accidents with high severity take place on the east coast.

- **Accidents count by each state and for each state count by severity of accidents**



This tells us the number of accidents in each state of United states and the distribution of severity. We see that California and Texas has the most number of accidents.

- **Percent of severity**



The above plot shows the distribution of count of different Severities. We can see that the data is highly imbalanced because the count of Severity 2 is double of that of Severity 3 and the other severities are minimal.

FEATURE ENGINEERING

Feature engineering is an important process before building any Machine Learning model. Most of the algorithms we intend to run work well with numerical data. Dummies were created in order to transform categorical attributes to numerical attributes using One Hot Encoding.

Time related features which provide information regarding the start and end time of the accident aren't very helpful for our analysis. New features such as day of the week and hour of the day were extracted from these existing features which will be more helpful for our analysis.

We extracted some of the most important weather conditions from our weather condition column as using the column directly would not have added to the predicting capability of our model. Then we made dummy variables for each of this weather condition to show that during the accident if that weather condition was present or not.

TMC column had around 25000 missing values. It was an important column that could significantly improve our models. So, we made a new category for these values as we thought it was the best way to impute.

Our target variable Severity ranges between 1 and 4. There is an imbalance in the dataset as most of the data points belong to Severity 2 and there are fewer observations with Severity 1. We combine Severity 1 and 2 as 2 to overcome this problem.

Few Machine Learning algorithms like Gradient Boosted Trees work only for binary classification problem on Apache Spark. To support the minimum requirements for such models we convert our multiclass classification problem to a binary classification problem by merging Severity 1 and 2 as 0 and Severity 3 and 4 as 1.

We have split the entire dataset into training and validation data, where 80% of the observations is training data and the rest 20% is the validation data. The models are trained using the training

data and their performance is evaluated using validation data. The data thus processed can be used for both multiclass and binary classifications.

Handling Imbalance:

To handle the imbalance in our data, we used oversampling and undersampling technique. For multiclass target, we undersampled the data with severity 2 and oversampled the data with severity 4 to match the number of data points in class 3. In this way, all the classes had the same number of data points and our data was balanced. For binary classification, we undersampled class 0 such that it matched the number of data points in class 1.

MODELING

All the grid search models have been tuned by using 5-fold Cross Validation.

1. Logistic Regression

The first model we ran was Logistic Regression. Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. It transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes. We have used standard scaler to center the data to its mean to standardize the input being provided to Logistic Regression.

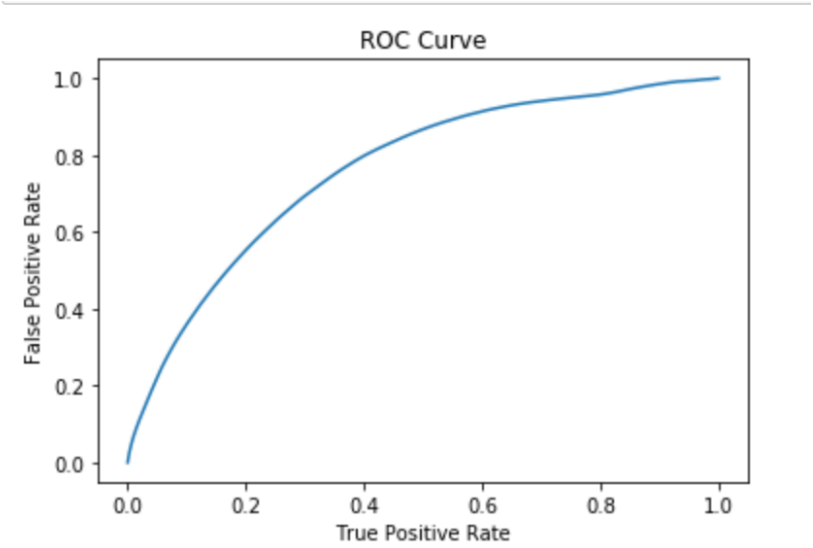
Base model was run and then tuned to run grid search to get the best model. The hyperparameters tuned in the grid search were regularization parameter and elastic net parameter. The use of L1 regularization is that eliminates some of the features from the input by reducing their coefficients to zero which comes in handy to reduce the dimensionality of the dataset. Elastic Net Parameter determines the ratio of L1 and L2 Regularization in the model and Regularization parameter to shrink the coefficients to zero.

- **Multiclass and Binary Classification:**

  For Multiclass regression classification, the values used for tuning the grid search for Regularization parameter were 0.01, 0.04, 0.07 and for Elastic Net Parameter were 0.2, 0.5 and 0.8. The best model parameters came out to be 0.01 for Regularization parameter and 0.2 for Elastic Net Parameter. The number of features eliminated by using L1 regularization for each class is different. For Class 1, the number of features eliminated were 60 out of 119 features, for 2nd class of target variable 84 features out of 119 features were eliminated and for the 3rd class in the target variable 64 out of 119 features were eliminated.

  For binary regression classification, the values tuned for Regularization Parameter were 0.01, 0.04, 0.07 and for Elastic Net Regularization 0.1, 0.4, 0.7. The Best values came out to be 0.1 for Elastic Net and 0.01 for Regularization parameter. The number of features that were eliminated by this tuned model were 28 out of 119 features in the model.

The ROC curve and AUC score for the best model in Binary Classification grid search is shown below.




```
Training set areaUnderROC: 0.761692898046687
```

2. Decision Tree

Decision Tree is a simple representation for classifying examples. It is a Supervised Machine Learning where the data is continuously split according to a certain parameter. Using the decision algorithm, we start at the tree root and split the data on the feature that results in the largest information gain (IG). In an iterative process, we can then repeat this splitting procedure at each child node until the leaves are pure. This means that the samples at each leaf node all belong to the same class. In practice, we may set a limit on the depth of the tree to prevent overfitting. We compromise on purity here somewhat as the final leaves may still have some impurity.

Base model was run and then tuned to run grid search to get the best model. The hyperparameters tuned in the grid search were maxDepth of the tree, minInstancesPerNode and maxBins. Max Depth of tree is how deep the tree is, the deeper the tree generally produces higher accuracy, so we need to tune this parameter to its optimum to get better model performance. Minimum Instances Per Node states that for a node to split further, it needs these many number of instances to be trained which helps to make more accurate splits based on the learning from the data. Finally, Max Bins

tuning helps to make fine-grained decisions based on considering more candidates for splitting the node, so we need to get the optimum value of this parameter.

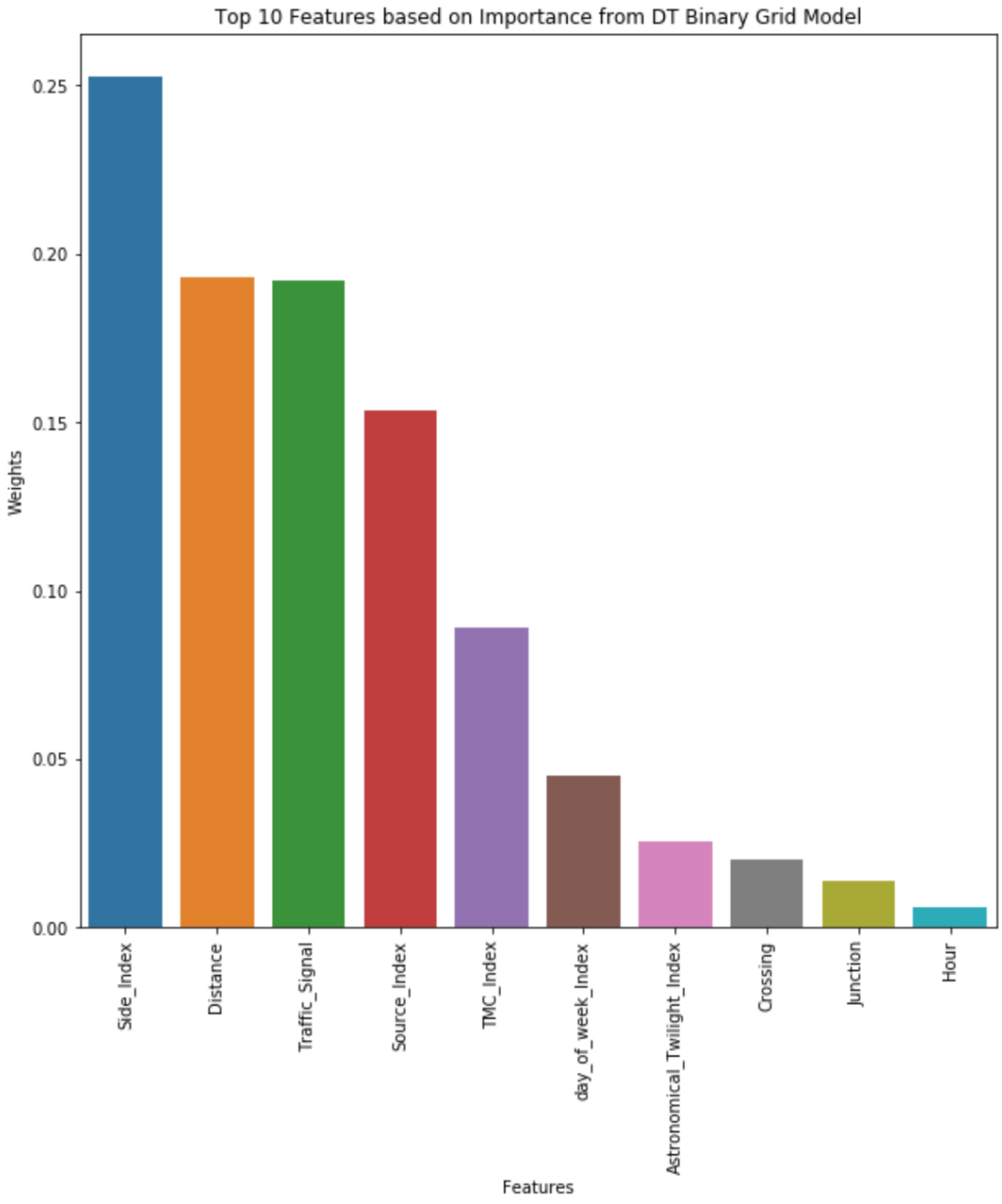
- **Multiclass and Binary Classification:**

Decision tree model for Multiclass problem and Binary Problem was tuned for maximum depth of the tree, minimum instances per node and maximum bins. Our model used the following hyperparameters in grid search max depth 10, 15, 30, for minimum instances per node 500, 1000, 1500 and 20, 35, 50 for maxBins.

For Multiclass regression classification, the best model parameters came out to be 30 for Max Depth, for Minimum Instances per Node 500 and 35 for Max Bins.

For binary regression classification, the best model parameters came out to be 10 for Max Depth, for Minimum Instances per Node 1500 and 50 for Max Bins.

Below are the top 10 important features from decision tree that contribute the most in determining severity of an accident.


Top 10 Features based on Importance from DT Binary Grid Model
Weights
0.25
0.20
0.15
0.10
0.05
0.00
Side_Index
Distance
Traffic_Signal
Source_Index
TMC_Index
day_of_week_Index
Astronomical_Twilight_Index
Crossing
Junction
Hour
Features

3. Random Forest

Random Forest is an ensemble tree-based learning algorithm. The Random Forest Classifier 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.

Random forest is one of the most accurate learning algorithms available. For many data sets, it produces a highly accurate classifier. It runs efficiently on large databases. It can handle thousands of input variables without variable deletion. It gives estimates of what variables that are important in the classification. It generates an internal unbiased estimate of the generalization error as the forest building progresses. It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.

Base model was run and then grid was tuned with the following hyperparameters for Random Forest. numTrees which is the number of the trees in the forest, as more of trees would be used for predicting if an instance belongs to a particular class there will be less overfitting. Max Depth means the depth of the trees in the forest, more the depth the better the model learns and increases accuracy of prediction, Impurity is the measure of randomness. As randomness reduces the information gain increases and we need to find the split where the information gain is maximum to use it as the decider to split the node, Gini Index is the measure of the impurity and Entropy is the measure of randomness.

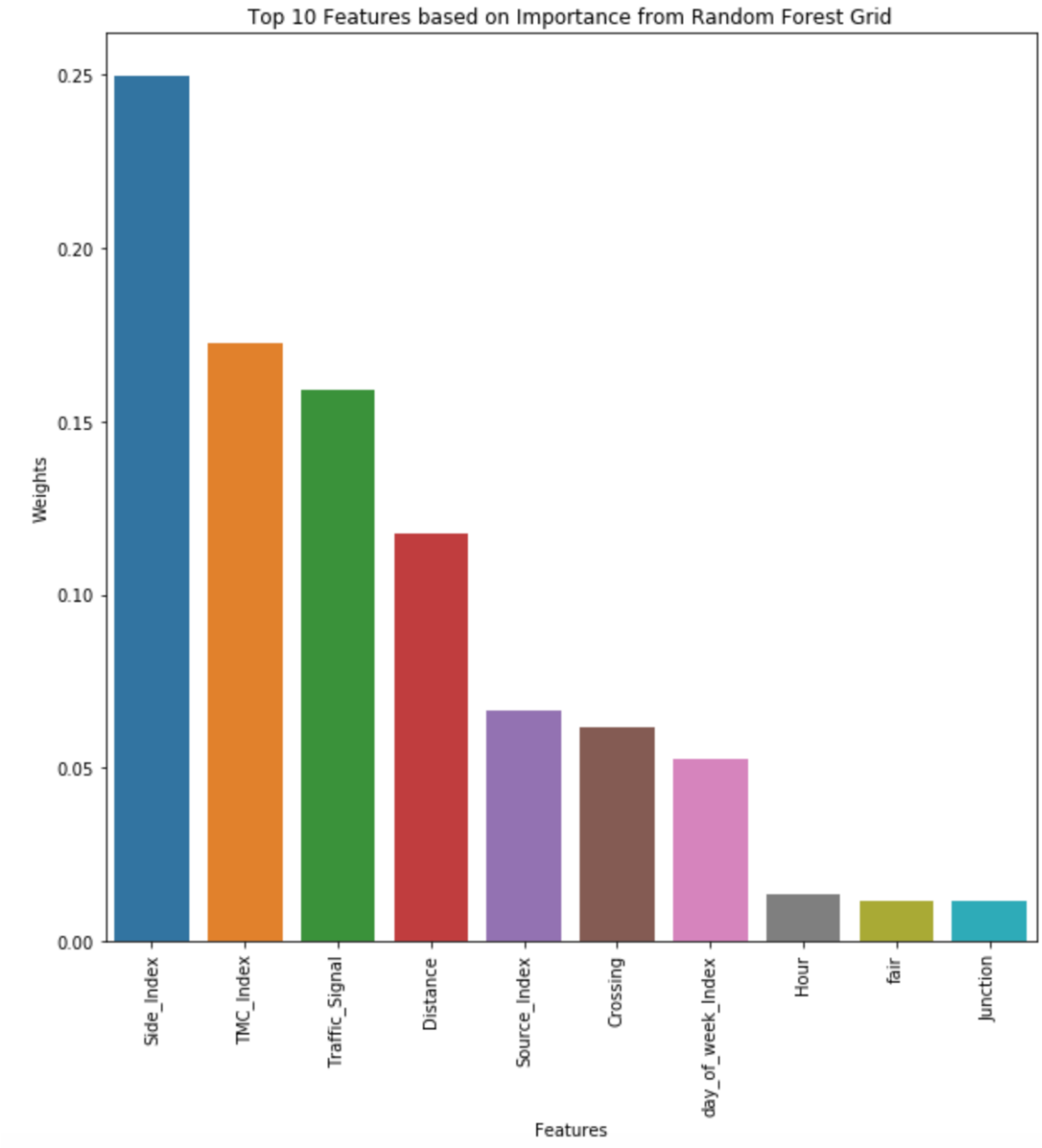
- **Multiclass and Binary Classification**

  In multiclass and binary random forest classifier, the values of hyperparameters used in the grid were 10, 30, 60 for numTrees and 3, 6, 10 for maxDepth, gini and entropy for impurity.
  The best model gave the following parameters numTrees 60, maxDepth 10 and gini impurity

  For Multiclass regression classification, the best model parameters came out to be 10 for Max Depth, for numTrees 60 and Gini for Impurity.
  For binary regression classification, the best model parameters came out to be 10 for Max Depth, for numTrees 60 and Entropy for Impurity.

Top 10 features that help in determining severity using multiclass random classifier are show below along with their weights.

Top 10 features that help in determining severity using binary random classifier are show below along with their weights.



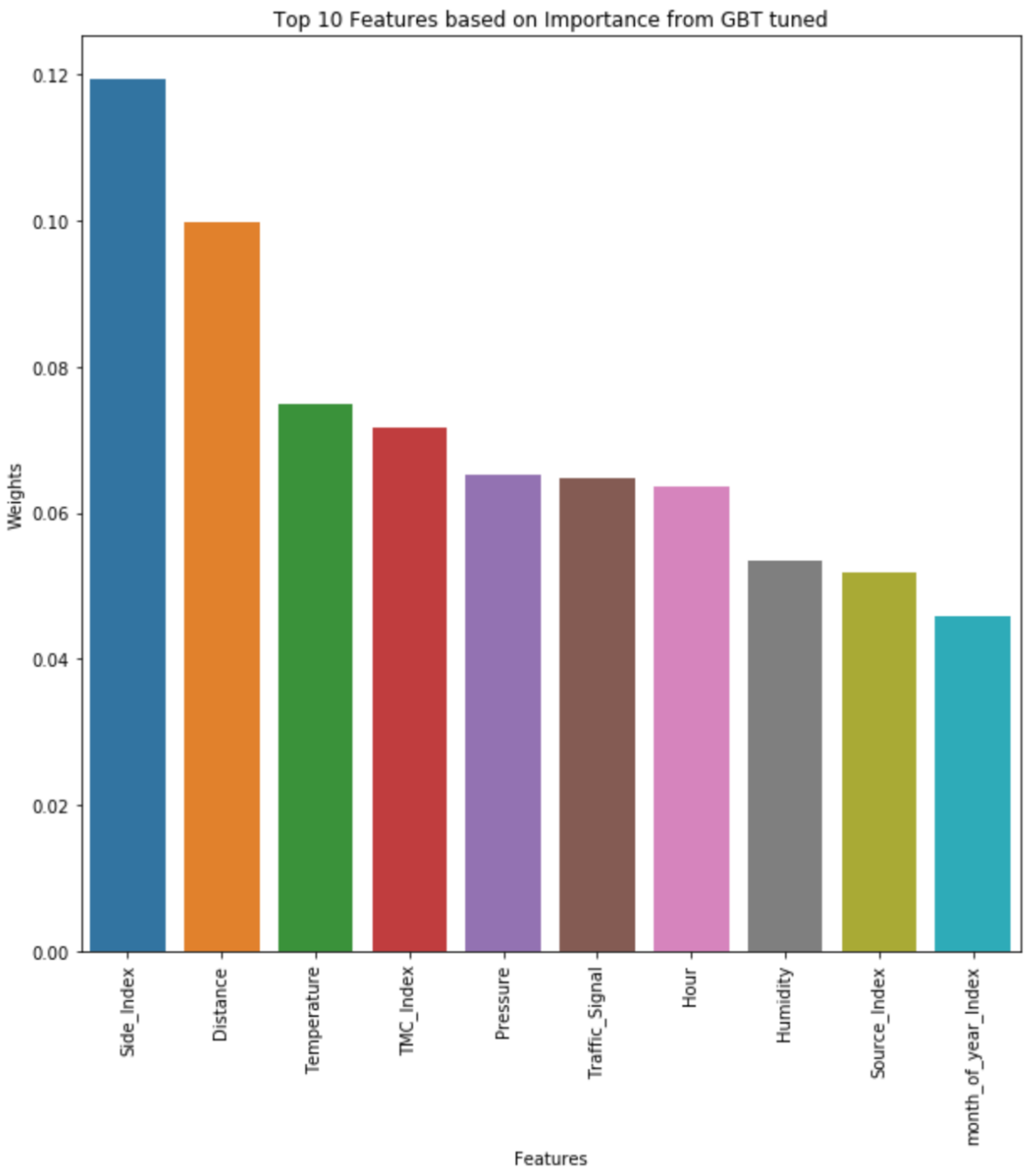
4. Gradient Boost Trees

Boosting is a method of converting weak learners into strong learners. In boosting, each new tree is a fit on a modified version of the original data set. Gradient Boosting trains many models in a gradual, additive and sequential manner. The trees are fed with residuals from the previous tree to improve the error and minimize this error as the residual is passed from one tree to another serially. Base model was run, and grid search was performed to tune the hyperparameters which were StepSize and Max Depth. MaxIter parameter was tuned outside of the grid to the best performance of the model. StepSize is the learning rate or the residual which is fed from 1 to another tree also called as lambda. MaxDepth is the depth of the trees in the GBT model which is the length of the trees and it usually ranges from 2 to 8. The MaxIter is the number of trees which is number of the stumps for training the model. The values for which the model was tuned was 3, 5, 8 for maxDepth and 0.2, 0.4, 0.01 for stepSize. MaxIter with values 50 and 55 was tested to find the best value. Currently Apache Spark supports only binary classification problems for GBT.

- **Binary Classification**

  The best model gave the following parameters of maxDepth 8 and stepSize of 0.2. The best model performance was reported for MaxIter parameter value of 55 over 50 keeping all the grid parameters to be same.

  The top 10 features along with their weights from GBT that contribute in determining severity are as follows.


Top 10 Features based on Importance from GBT tuned
Weights
0.12
0.10
0.08
0.06
0.04
0.02
0.00
Side_Index
Distance
Temperature
TMC_Index
Pressure
Traffic_Signal
Hour
Humidity
Source_Index
month_of_year_Index
Features

MODEL COMPARISION

For Balanced Data:

	Base Model		Tuned Model	
	Multiclass Classification	Binary Classification	Multiclass Classification	Binary Classification
Logistic Regression	Accuracy - 0.57	Accuracy - 0.67 AUC ROC - 0.75	Accuracy - 0.54	Accuracy - 0.67 AUC Score - 0.76
Decision Tree	Accuracy - 0.55	Accuracy - 0.65 AUC ROC - 0.70	Accuracy - 0.61	Accuracy - 0.66 AUC ROC - 0.68
Random Forest	Accuracy - 0.55	Accuracy - 0.66 AUC ROC - 0.75	Accuracy - 0.57	Accuracy - 0.68 AUC ROC - 0.78
Gradient Boosting	-	Accuracy - 0.68 AUC ROC - 0.78	-	Accuracy - 0.70 AUC ROC - 0.81

For Imbalanced Data:

	Base Model		Tuned Model	
	Multiclass Classification	Binary Classification	Multiclass Classification	Binary Classification
Logistic Regression	Accuracy - 0.72	Accuracy - 0.72 AUC ROC - 0.76	Accuracy - 0.72	Accuracy - 0.72 AUC Score - 0.76
Decision Tree	Accuracy - 0.71	Accuracy - 0.72 AUC ROC - 0.66	Accuracy - 0.73	Accuracy - 0.73 AUC ROC - 0.57
Random Forest	Accuracy - 0.69	Accuracy - 0.69 AUC ROC - 0.76	Accuracy - 0.72	Accuracy - 0.73 AUC ROC - 0.79
Gradient Boosting	-	Accuracy - 0.73 AUC ROC - 0.79	-	Accuracy - 0.74 AUC ROC - 0.80

The models did not overfit as the train and test accuracy and AUC scores were almost the same. For the table of model comparison, we can see that Gradient Boosting (GBT) performs the best on both the metrics Accuracy and AUC ROC Score. The highest AUC ROC score that was achieved was 0.81 and accuracy 0.70 from all the models for the balanced data whereas for the imbalanced data the highest accuracy was 0.74 and the highest AUC ROC score was 0.80. Accuracy was reduced for the models which were run using balanced data as compared to the imbalanced data and the AUC ROC score was higher for the balanced dataset. Accuracy was more for the imbalanced dataset as the models were predicting the class with highest instances most of the times.

CONCLUSION

From our Exploratory Data Analysis, we found that in various variables like day of the week, hour of the day, month of the year and side of the road for which the number of accidents are very high for certain values. This can help the authorities to prepare in advance during these times and implement stricter traffic rules to decrease the number. Also, we found that during the nighttime the severity of accidents is more which is intuitive. This can indicate that more safety measures need to be taken at night time. From the feature importance plots, we found that TMC, Side, Source, Hour, Distance were some common important features from our models that can be used in predicting the severity of an accident. Using L1 regularisation and feature importance from tree-based models, we found out that many features were eliminated. So, we should consider removing those features. GBT was the best model in case of both imbalanced and balanced dataset for both metrics accuracy and AUC ROC score.

Python Notebook files for all the Methodologies:

Data Cleaning and Exploratory Data Analysis – Data_Cleaning_EDA.ipynb
String Indexing and One Hot Encoding – StringIndexing_OHE_Conversion.ipynb
Oversampling and Undersampling – Undersampling_Oversampling.ipynb

Models:

Imbalanced Dataset:

Binary Classification:
Logistic Regression – LR_Binary.ipynb
Random Forest, Decision Tree and GBT – RF_DT_GBT_Binary.ipynb
Multiclass Classification:
Logistic Regression – LR_Multiclass.ipynb
Random Forest and Decision Tree– RFDT_Multiclass.ipynb

Balanced Dataset:

Binary Classification:
Logistic Regression, Random Forest, Decision Tree and GBT –
RF_DT_GBT_LR_Binary_Bal.ipynb
Multiclass Classification:
Logistic Regression – LR_Multiclass_Bal.ipynb
Random Forest and Decision Tree– RFDT_Multiclass_Bal.ipynb