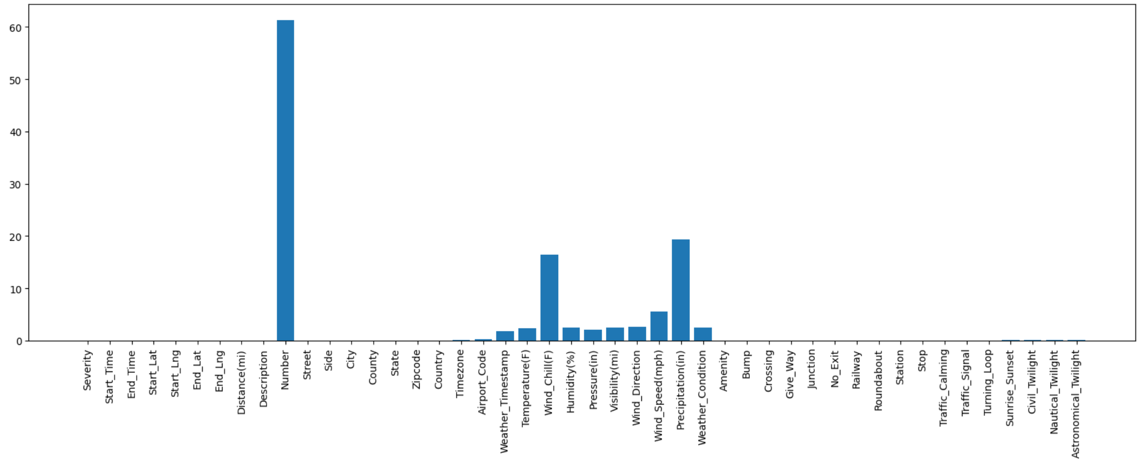
REPORT

This is a countrywide car accident dataset, which covers 49 states of the USA. The accident data are collected from February 2016 to Dec 2021, using multiple APIs that provide streaming traffic incident (or event) data. These APIs broadcast traffic data captured by a variety of entities, such as the US and state departments of transportation, law enforcement agencies, traffic cameras, and traffic sensors within the road-networks. Currently, there are about 2.8 million accident records in this dataset.

PREPROCESSING AND EDA :

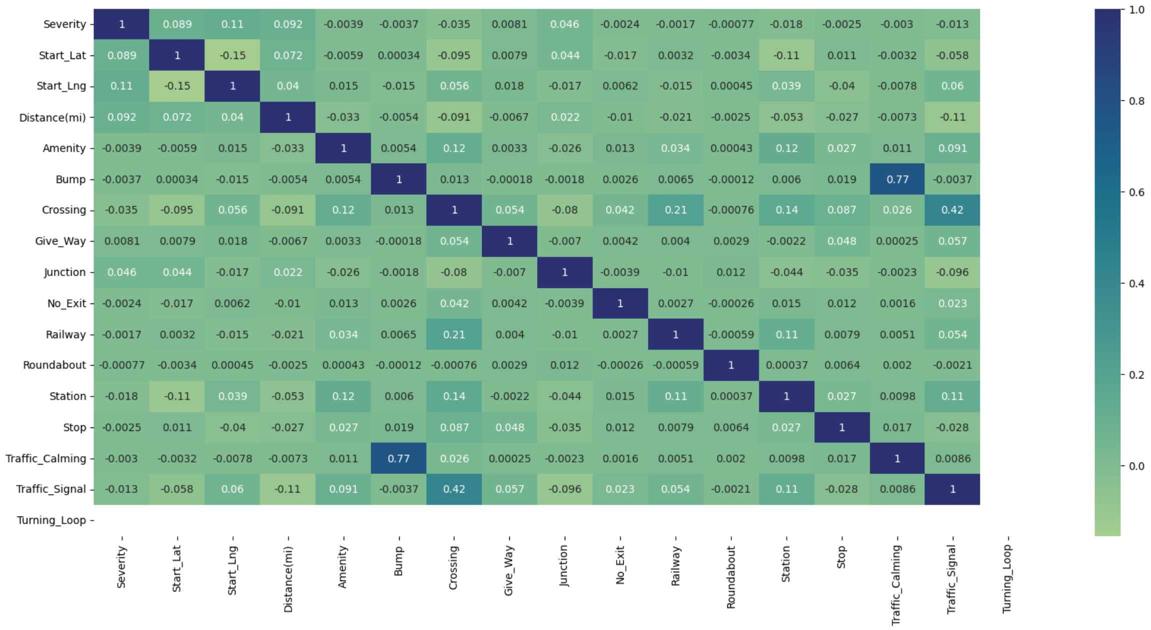
- We loaded “ US Car accidents” dataset and we have done the preprocessing on the features and how they are related to the target variables
- Our dataset contains 2.8 million(approx) samples and 47 features
- We found that the dataset contains missing values and we plotted the graph representing the missing values



- We created a separate data frame for analyzing the data
- We can also say that we don't need the data of **end_lat** and **end_lng** also because we care at the point of collision only
- We can also drop the columns of **sunrise_sunset**, **civil_twilight**, **nautical_twilight**, **Astronomical_twlight**
- **sunrise_sunset** data doesn't matter because the accident dataset clearly independent of time and the other three data is completely not useful so we drop the above columns for analyzing the data

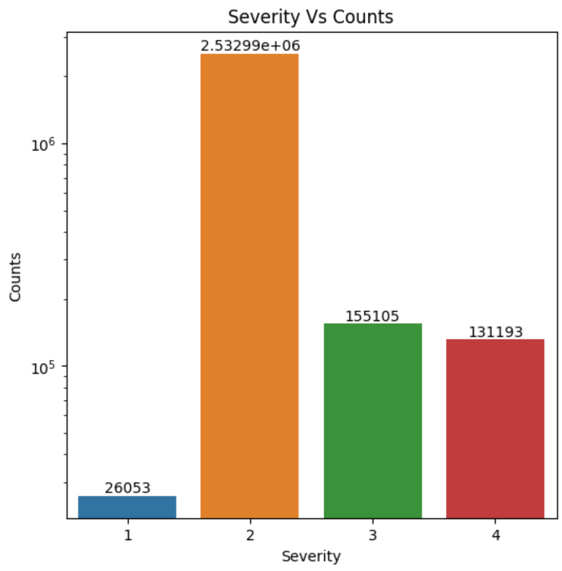
- **Time_zone** column is clearly not necessary because the accidents almost lie in the same region
- **Airport_Code** is clearly a geographical characteristic so it not necessary
- As the whole dataset has only one country US so we can drop that column
- As we have the latitude and longitude data we can remove the addresses/Descriptive columns in the datasets
- We can remove Description that barely defines the way of the description reported by the first witness so we won't depend on it completely
- We can drop all the weather related factors from our dataset as it does not affect the way of occurring of Accidents
- We see that **Number**,**Wind_chill** and **Precipitation** have more than 10% of missing values do we will drop them so that they do not cause any problem while analyzing the data

Covariance matrix using for visualizing the dataset

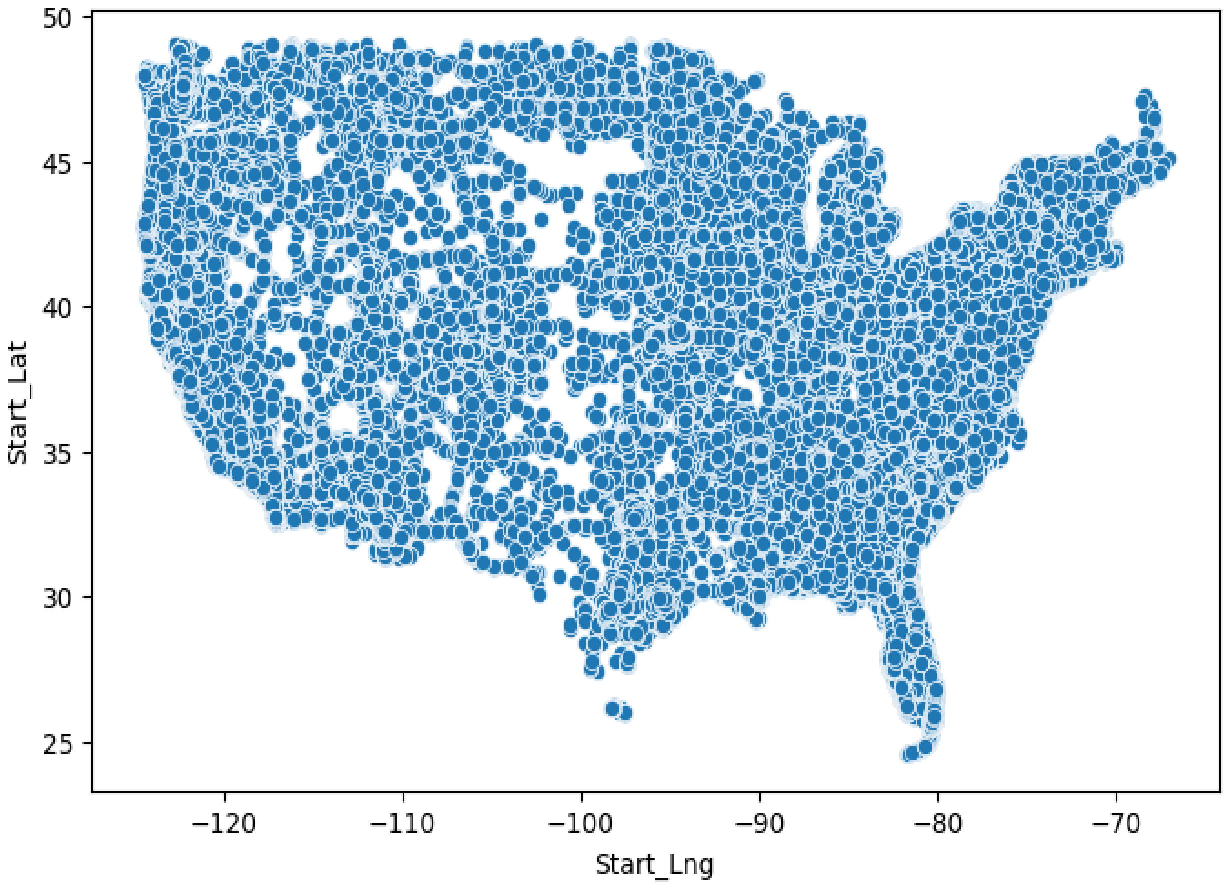


	Severity	Start_Lat	Start_Lng	Distance(mi)	Amenity	Bump	Crossing	Give_Way	Junction	No_Exit	Railway	Roundabout	Station	Stop	Traffic_Calming	Traffic_Signal	Turning_Loop
Severity	1	0.089	0.11	0.092	-0.0039	-0.0037	-0.035	0.0081	0.046	-0.0024	-0.0017	-0.00077	-0.018	-0.0025	-0.003	-0.013	
Start_Lat	0.089	1	-0.15	0.072	-0.0059	0.00034	-0.095	0.0079	0.044	-0.017	0.0032	-0.0034	-0.11	0.011	-0.0032	-0.058	
Start_Lng	0.11	-0.15	1	0.04	0.015	-0.015	0.056	0.018	-0.017	0.0062	-0.015	0.00045	0.039	-0.04	-0.0078	0.06	
Distance(mi)	0.092	0.072	0.04	1	-0.033	-0.0054	-0.091	-0.0067	0.022	-0.01	-0.021	-0.0025	-0.053	-0.027	-0.0073	-0.11	
Amenity	-0.0039	-0.0059	0.015	-0.033	1	0.0054	0.12	0.0033	-0.026	0.013	0.034	0.00043	0.12	0.027	0.011	0.091	
Bump	-0.0037	0.00034	-0.015	-0.0054	0.0054	1	0.013	-0.00018	-0.0018	0.0026	0.0065	-0.00012	0.006	0.019	0.77	-0.0037	
Crossing	-0.035	-0.095	0.056	-0.091	0.12	0.013	1	0.054	-0.08	0.042	0.21	-0.00076	0.14	0.087	0.026	0.42	
Give_Way	0.0081	0.0079	0.018	-0.0067	0.0033	-0.00018	0.054	1	-0.007	0.0042	0.004	0.0029	-0.0022	0.048	0.00025	0.057	
Junction	0.046	0.044	-0.017	0.022	-0.026	-0.0018	-0.08	-0.007	1	-0.0039	-0.01	0.012	-0.044	-0.035	-0.0023	-0.096	
No_Exit	-0.0024	-0.017	0.0062	-0.01	0.013	0.0026	0.042	0.0042	-0.0039	1	0.0027	-0.00026	0.015	0.012	0.0016	0.023	
Railway	-0.0017	0.0032	-0.015	-0.021	0.034	0.0065	0.21	0.004	-0.01	0.0027	1	-0.00059	0.11	0.0079	0.0051	0.054	
Roundabout	-0.00077	-0.0034	0.00045	-0.0025	0.00043	-0.00012	-0.00076	0.0029	0.012	-0.00026	-0.00059	1	0.00037	0.0064	0.002	-0.0021	
Station	-0.018	-0.11	0.039	-0.053	0.12	0.006	0.14	-0.0022	-0.044	0.015	0.11	0.00037	1	0.027	0.0098	0.11	
Stop	-0.0025	0.011	-0.04	-0.027	0.027	0.019	0.087	0.048	-0.035	0.012	0.0079	0.0064	0.027	1	0.017	-0.028	
Traffic_Calming	-0.003	-0.0032	-0.0078	-0.0073	0.011	0.77	0.026	0.00025	-0.0023	0.0016	0.0051	0.002	0.0098	0.017	1	0.0086	
Traffic_Signal	-0.013	-0.058	0.06	-0.11	0.091	-0.0037	0.42	0.057	-0.096	0.023	0.054	-0.0021	0.11	-0.028	0.0086	1	
Turning_Loop																	

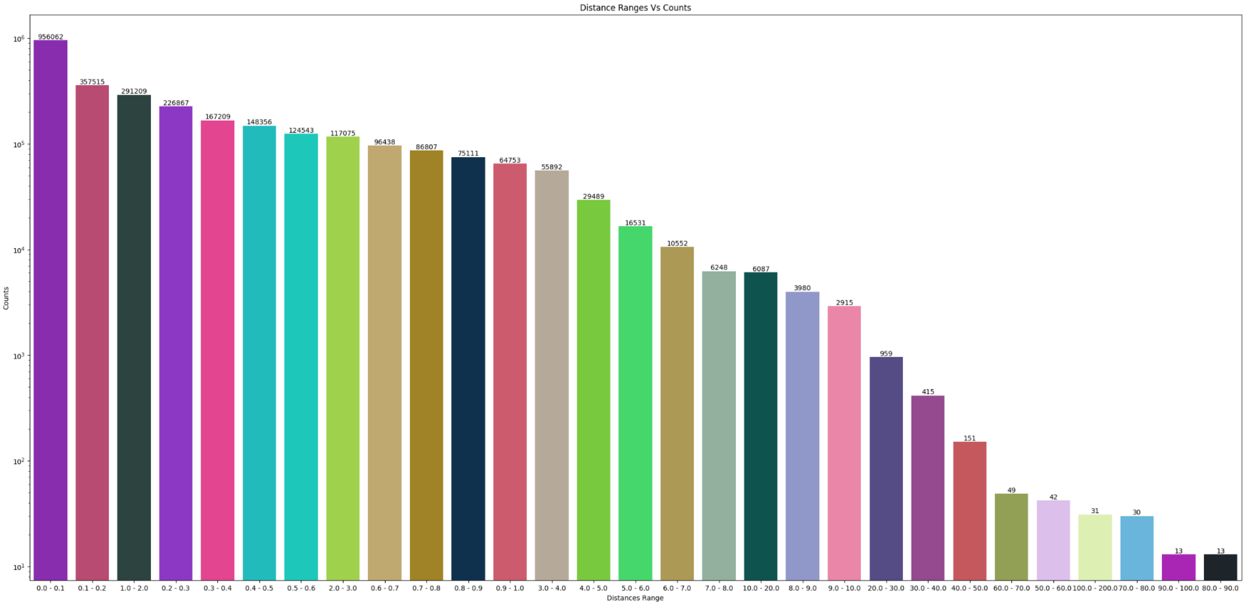
Number of unique values in the column of severity

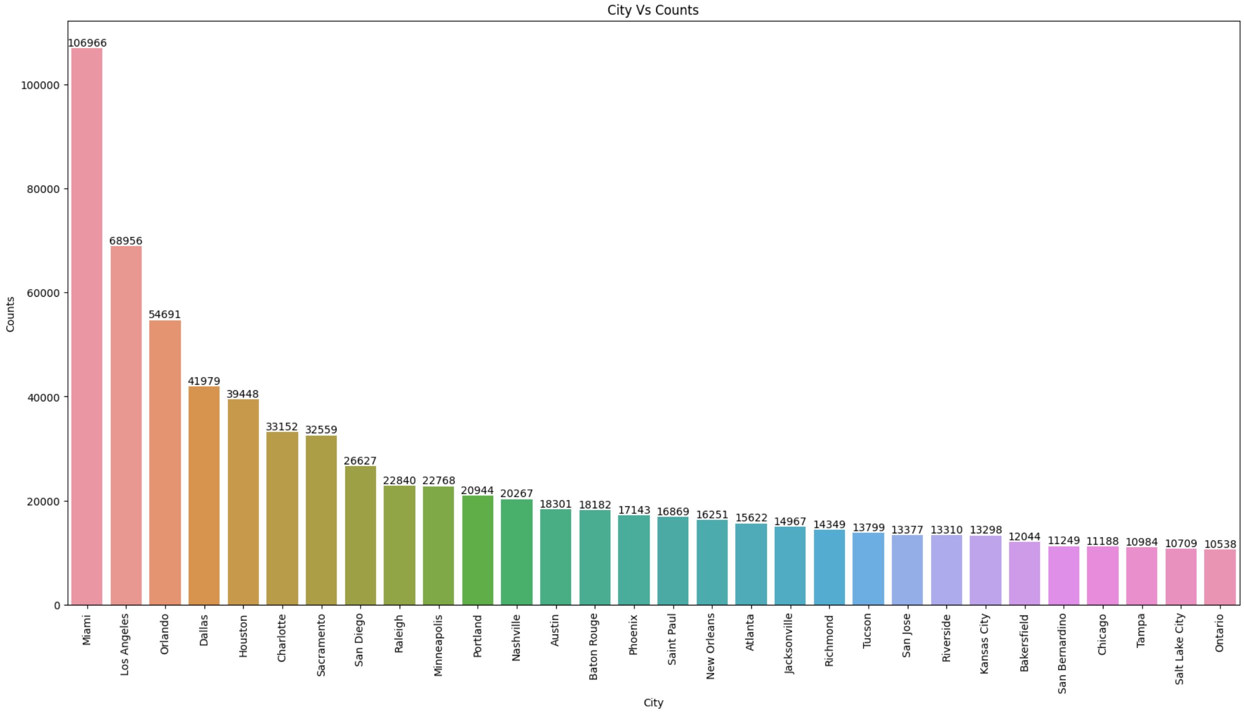


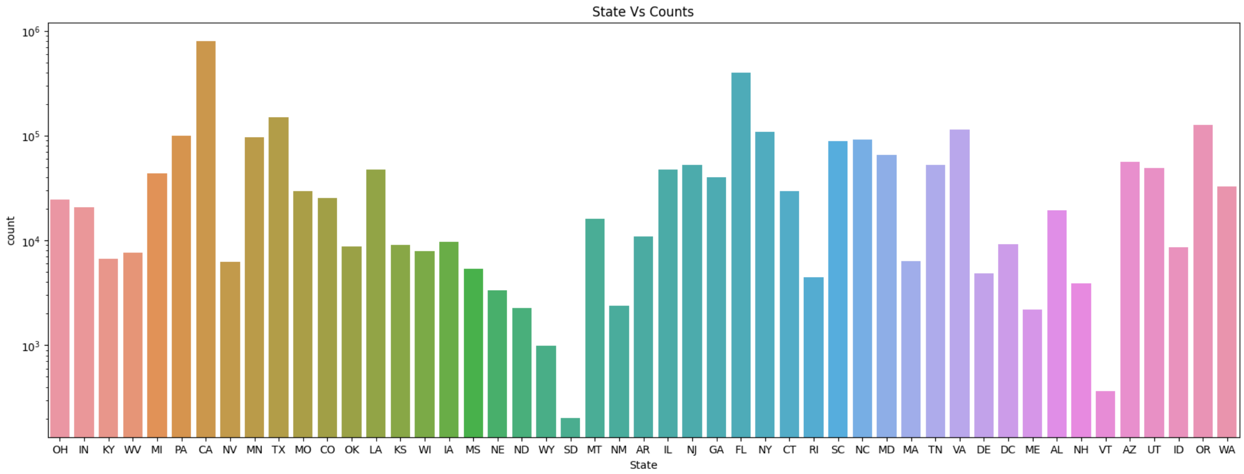
Map plotted from the samples in the dataset

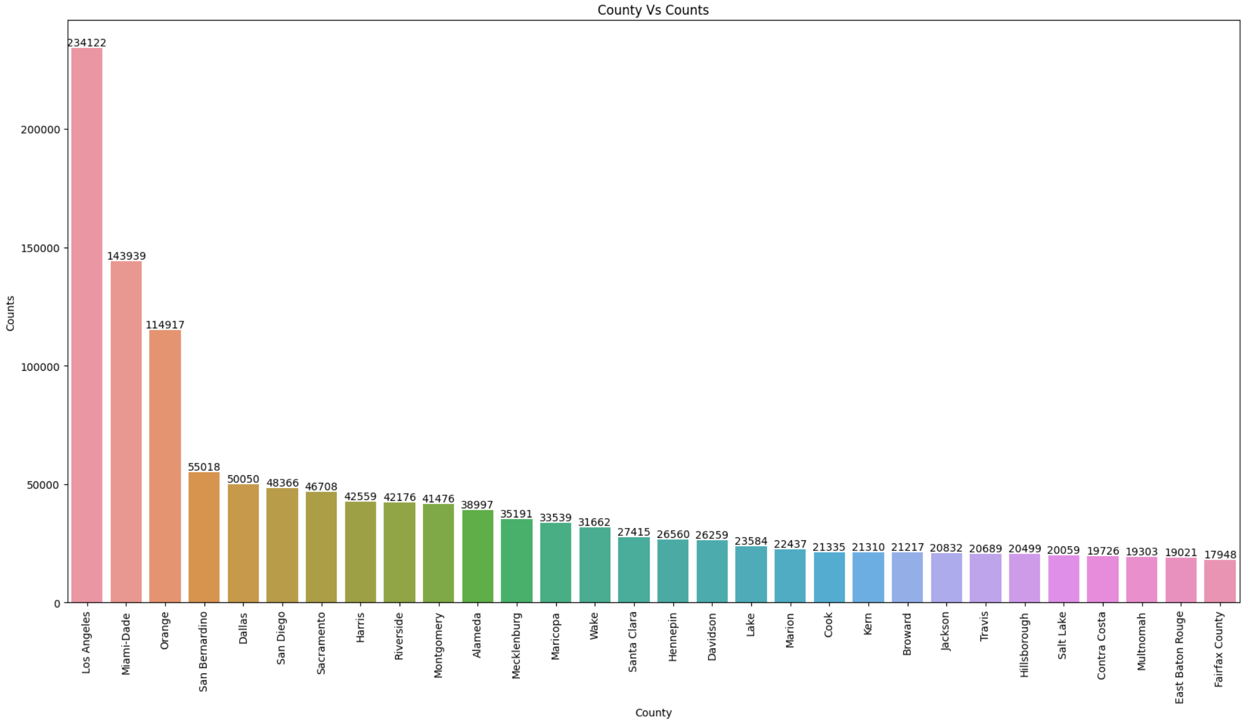


Graphs of number accidents occurred in the the above interval of distances, city, county and the states:

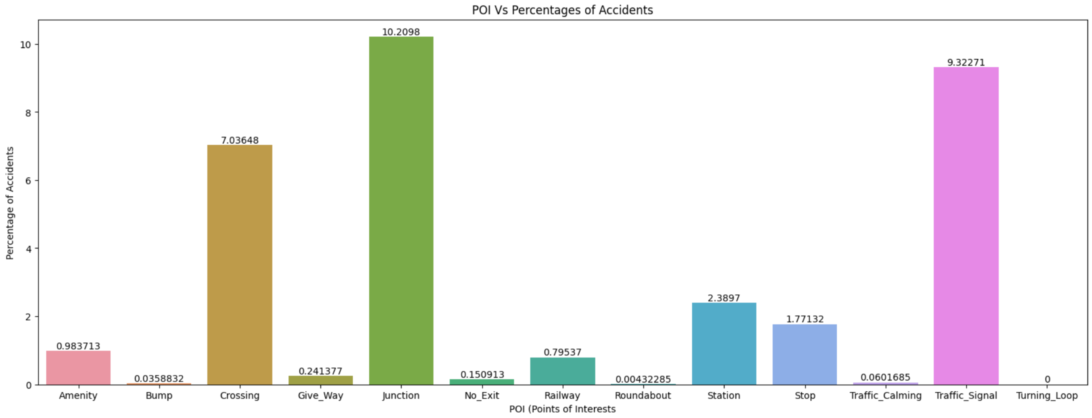




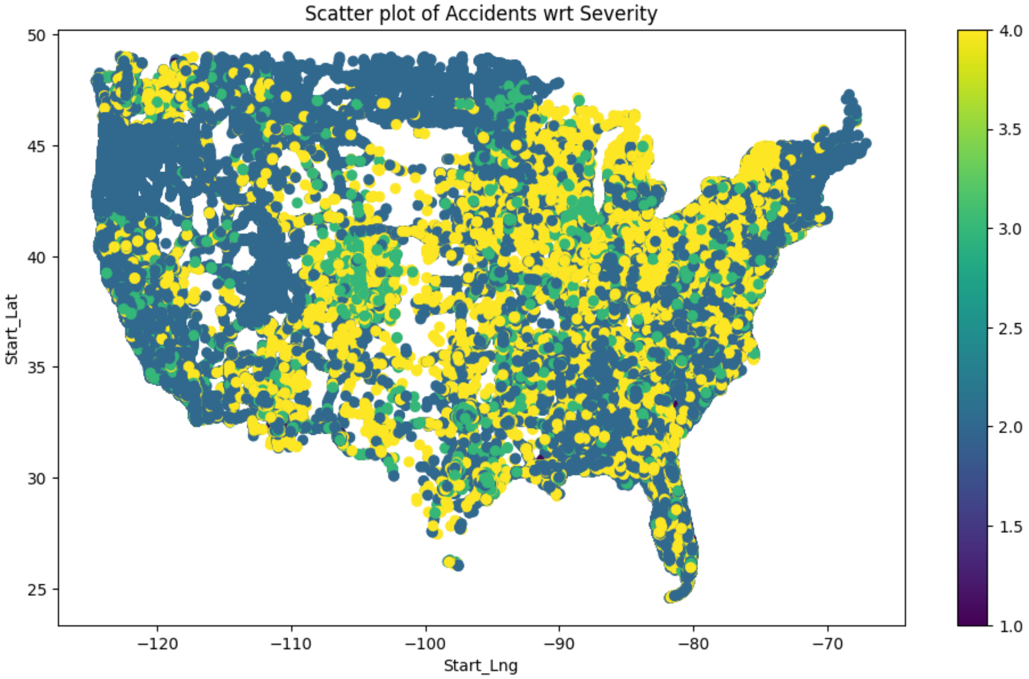




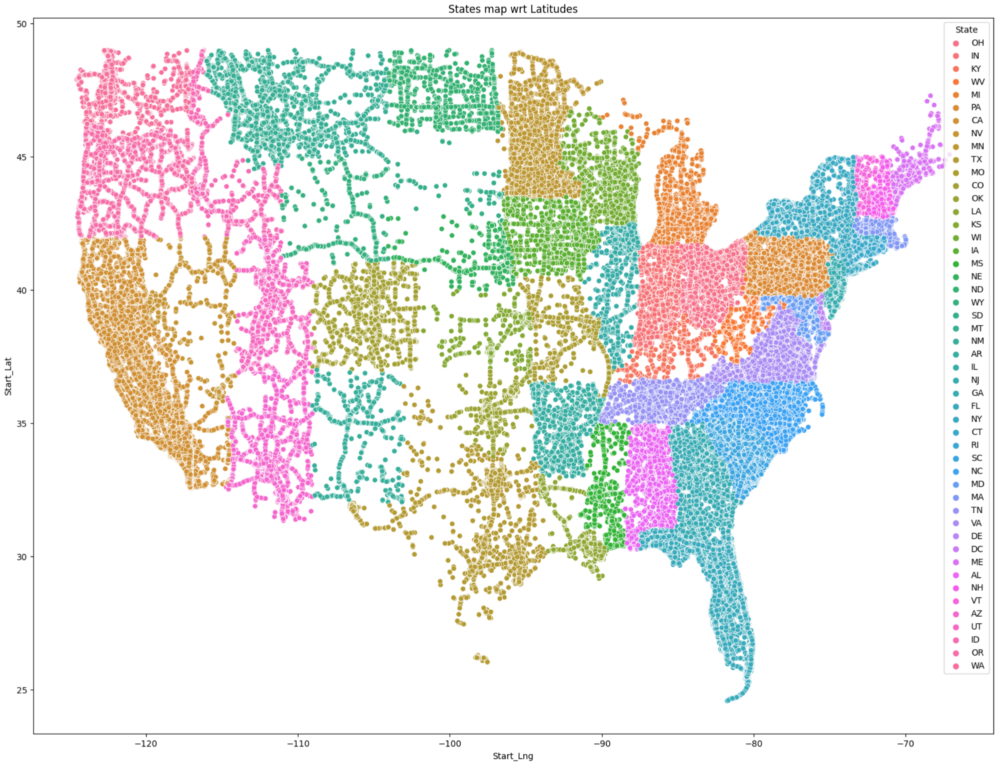
Graphs of number of accidents occurred due to POI Attributes:



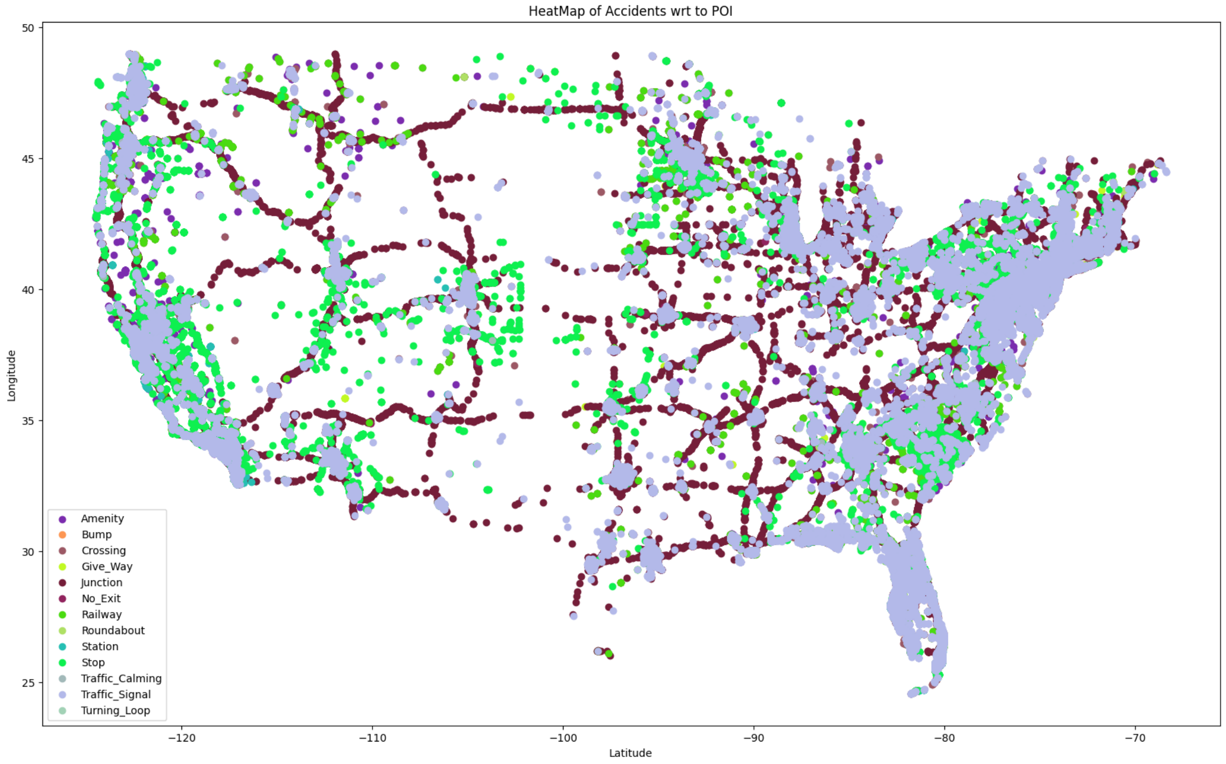
Plotting the map from the dataset based on severity:



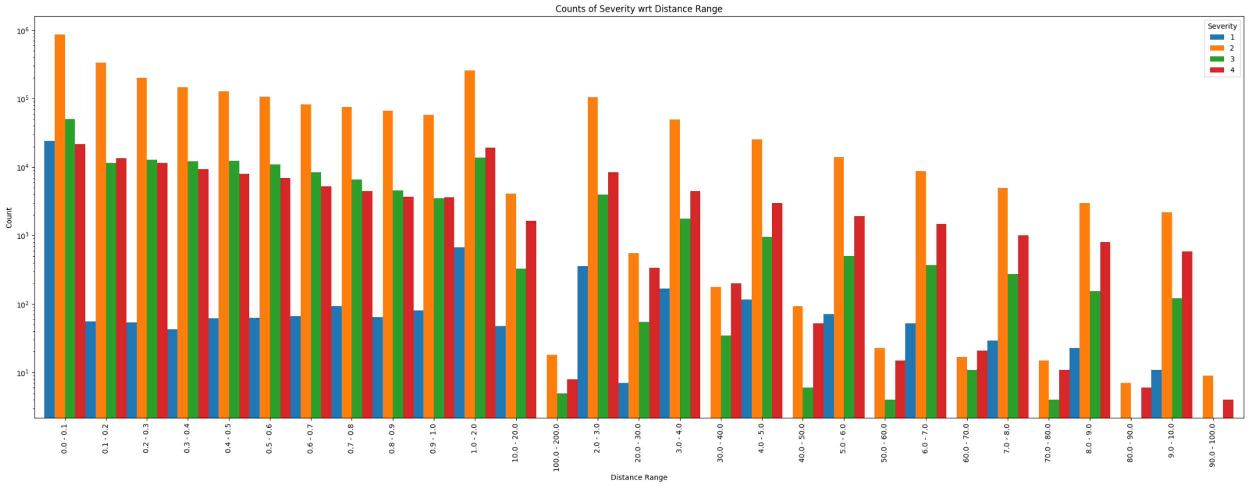
Map of location of accidents in state by latitudes:



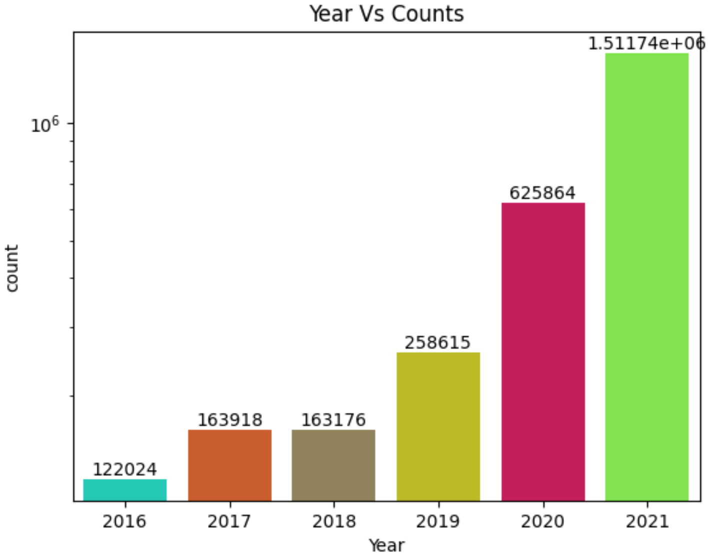
Heatmap of accidents wrt POI:



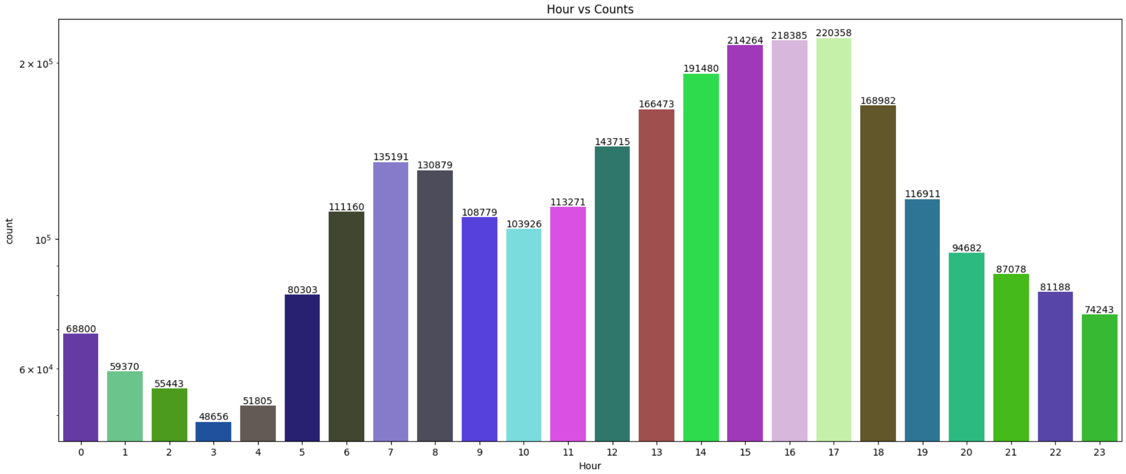
Visualization between distance and severity levels:



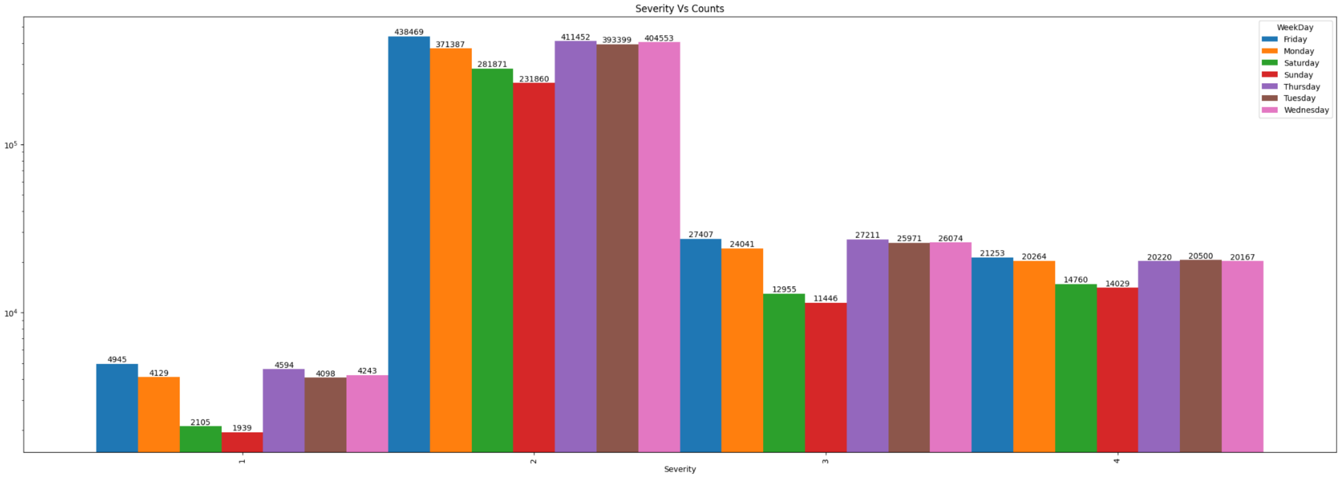
Plotting the graph of number of accidents occurred in a particular year:



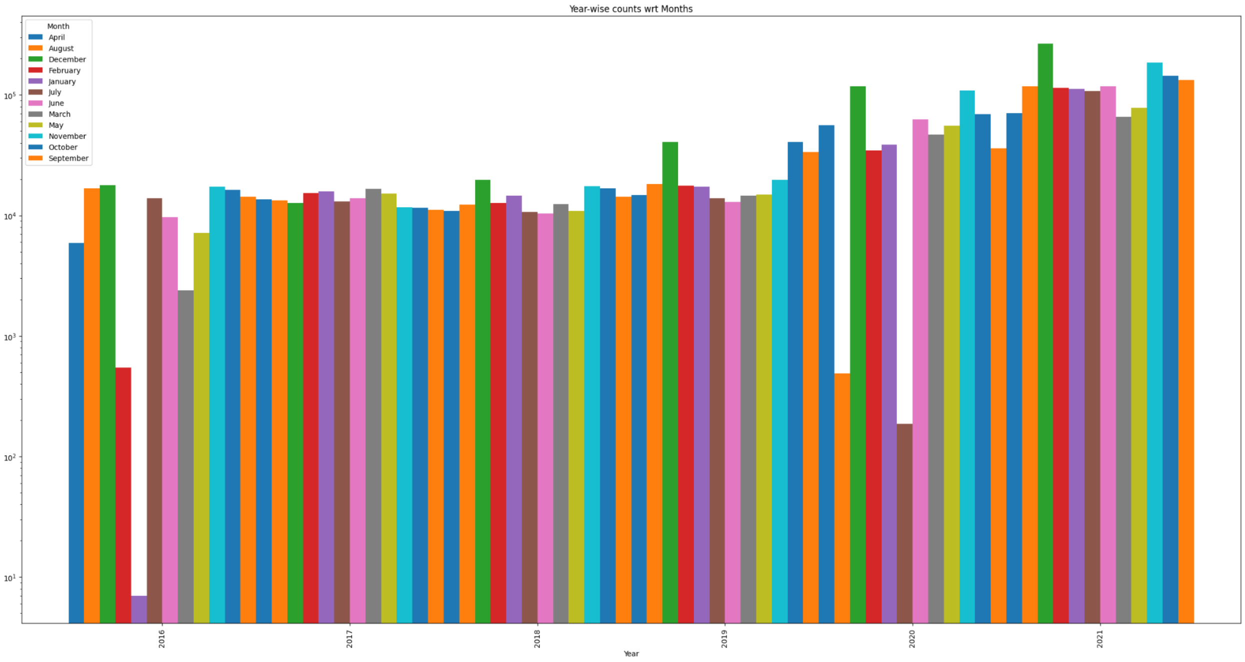
Distribution of accident counts across a day:



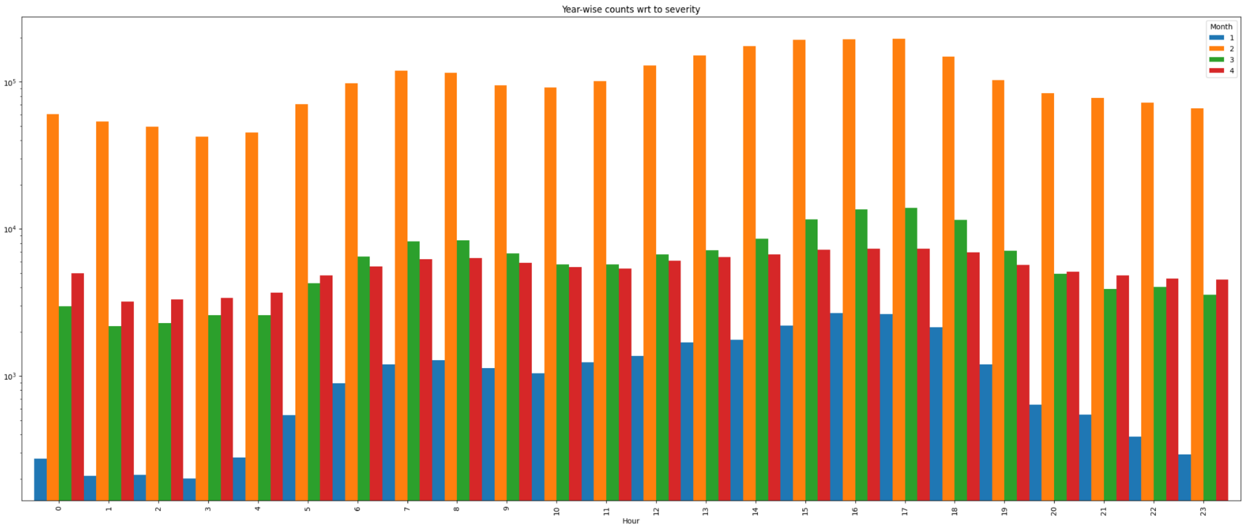
Graph of severity of accidents and the day of the week they occur:



Graph of counts of accidents wrt month of an year :



Graph of hourly count of accidents for each severity level :



Pipelining:

- We are dropping the above columns from the dataset for better accuracy and overfitting **'Wind_Chill(F)','ID','Start_Time','End_Time','End_Lat','End_Lng','Description','Number','Street','Side','County','State','Zipcode','Country','Timezone','Airport_Code','Weather_Timestamp','Sunrise_Sunset','Civil_Twilight','Nautical_Twilight','Astronomical_Twilight','Wind_Direction','Amenity','Bump','Give_Way','No_Exit','Railway','Roundabout','Traffic_Calming','Turning_Loop'**
- As mentioned above **'End_Time','End_Lat','End_Lng','Description','Number','Street','Side','County','State','Zipcode','Country','Timezone','Airport_Code','Weather_Timestamp','Sunrise_Sunset','Civil_Twilight','Nautical_Twilight','Astronomical_Twilight','Wind_Direction'** these features don’t help in predicting the severity and location
- And we dropped **'Start_Time'** because we have stored date,year,minute,hour in separate columns
- We dropped **‘ID’** because it actually doesn't help in predicting
- We dropped these columns because number of accidents occurred when these attributes present are very low so it doesn't help in predicting **'Amenity','Bump','Give_Way','No_Exit','Railway','Roundabout','Traffic_Calming','Turning_Loop'**
- **'Wind_Chill(F)'** dropped because it is not correlated with target variable and even it has missing values
- We replaced the missing of the columns ["Temperature(F)", "Humidity(%)", "Pressure(in)", "Visibility(mi)", "Wind_Speed(mph)", "Precipitation(in)"] with the median
- And drop the remaining columns containing the null values
- We encoded the features of boolean type with 0,1 and we label encoded the features ‘City’ and ‘Weather_Condition’
- And we normalized the features

First we are predicting Severity:

Handling the imbalanced dataset:

- The dataset is highly imbalanced so we need to do sampling
- Here we need to apply undersampling because the data is large

Dividing the dataset into train,validation and test into 70:10:20

Without applying LDA:

When we are training the RFC :

Random forest is a robust algorithm that can handle noisy and high-dimensional data, capture nonlinear relationships, and has good generalization performance. These factors make it a good choice for accident prediction datasets, which often have these characteristics. Random forest's ensemble approach also allows it to combine multiple decision trees to create a more accurate model.

Accuracy Score of test dataset is : 77.62958472264323 %
Accuracy Score for valid dataset is : 76.96155206386096 %

Logistic Regression:

Logistic regression is effective for predicting the severity of accidents due to its ability to model the relationship between a binary outcome variable and predictor variables, handle both categorical and continuous variables, provide interpretable coefficients, and be computationally efficient for large datasets.

Accuracy Score of test dataset is : 37.32444175002526 %
Accuracy Score for valid dataset is : 36.108725306926694 %

Knn:

KNN is used for accident prediction due to its ability to handle non-linear relationships and lack of assumptions about the data. This makes it a useful algorithm for dealing with complex and varied datasets. By finding the K-nearest neighbors to a data point, KNN can predict the label of the new data point, making it a powerful tool for classification tasks. However, KNN can be computationally expensive and sensitive to the choice of distance metric used, so careful selection of parameters is necessary for optimal performance.

Accuracy Score of test dataset is : 60.73557643730423 %
Accuracy Score for valid dataset is : 60.34456626079927 %

Gradient Boosting:

Gradient boosting is a powerful machine learning algorithm that is often used in accident prediction due to its ability to handle noisy and high-dimensional data, capture nonlinear relationships, handle missing data and categorical variables, and create strong predictive models. Additionally, it is a flexible algorithm that can be used for both classification and regression tasks, making it well-suited for accident prediction problems.

Accuracy Score of test dataset is : 73.48691522683642 %
Accuracy Score for valid dataset is : 73.5664123680089 %

Decision tree:
Decision trees are popular for accident prediction due to their ability to capture complex relationships, handle both categorical and continuous variables, identify important variables and interactions, and produce easy-to-understand models. They are useful for both classification and regression tasks and can be used to make predictions for new, unseen data. Overall, decision trees are a versatile and effective algorithm for accident prediction.

Accuracy Score of test dataset is : 68.53591997575023 %
Accuracy Score for valid dataset is : 68.03415348860709 %

Multi-layer perceptron:
Multi-layer perceptron (MLP) is a type of artificial neural network that is commonly used for prediction analysis because it can handle non-linearly

separable data and capture complex relationships between features and the target variable. MLP can learn to make predictions based on a set of input features, and can be trained to classify or regress the target variable.

Accuracy Score of test dataset is : 61.79650399110842 %
Accuracy Score for valid dataset is : 61.233769514474815 %

With applying LDA

When we applied LDA on the dataset their is no much difference in the accuracies on the above applied models

Even though the number of features are almost the same this occurred because we removed the features which doesn’t contribute to the severity . So even after applying LDA the number of features doesn’t change.

So applying LDA doesn’t improve the accuracy of the above models. So applying LDA is ineffective.

We are predicting Location

- For Location prediction we are taking ‘**state**’ into consideration.
- But their 49 states so we regrouped classes into 4 classes on basis on location
- The 4 classes are **Western, Southern,Midwestern,Northeastern** for easy prediction
- The data is now balanced when target variable is **State** and we encoded the 4 classes
- We again divided the dataset into train,validation,test and then we applied LDA on the dataset

After applying LDA the number features are reduced to 3 so it is effective for using this

When we are training the RFC:

Accuracy Score of test dataset is : 99.17515543358606 %
Accuracy Score for valid dataset is : 99.17205290954018 %

Logistic Regression:

Accuracy Score of test dataset is : 98.82320107911995 %
Accuracy Score for valid dataset is : 98.78521246184408 %

KNN:

Accuracy Score of test dataset is : 99.0317379102904 %
Accuracy Score for valid dataset is : 99.0484035079219 %

Decision tree:

Accuracy Score of test dataset is : 98.80963455664605 %
Accuracy Score for valid dataset is : 98.78928242647414 %

Gradient Boosting:

Accuracy Score of test dataset is : 99.08367831061909 %
Accuracy Score for valid dataset is : 99.04685304520568 %

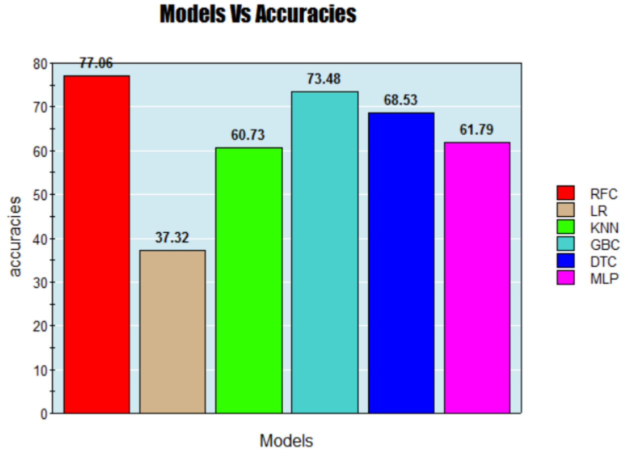
Multi-Layer Perceptron:

Accuracy Score of test dataset is : 99.08251546583561 %
Accuracy Score for valid dataset is : 99.08173845632055 %

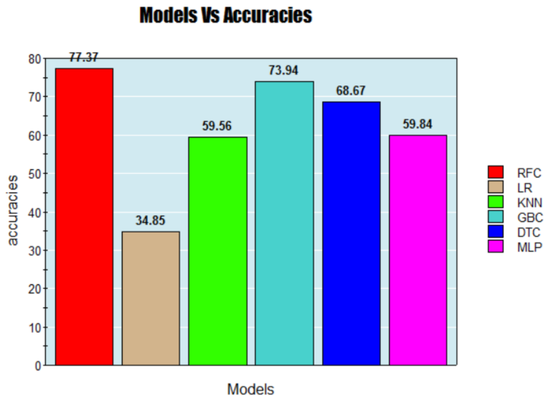
Conclusion

- When we handling with big data we encounter with 2 issues
    1) Unbalanced data
    2) More features
- We have done the unbalanced by undersampling and applied LDA for handling more features
- We cannot decide that a particular model is best for all datasets, but we can find the best performing models according to our dataset
- We observed that random forest is giving more accuracy than other due to can handle noisy and high-dimensional data, capture non-linear relationships while predicting severity
- Logistic regression is not working well because it struggles to handle high-dimensional and noisy data
- While we are predicting Location we regrouped the classes and then we have done the prediction
- We applied the LDA we observed that the features are reduced and its giving good accuracies
- Random Forest gave good accuracy because it can handle noisy and high-dimensional data, capture non-linear relationships while predicting location

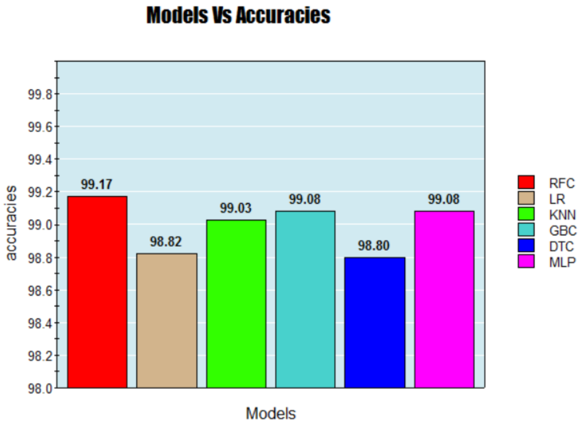
Accuracies of models for Severity prediction before applying LDA



Accuracies of models for Severity prediction after applying LDA

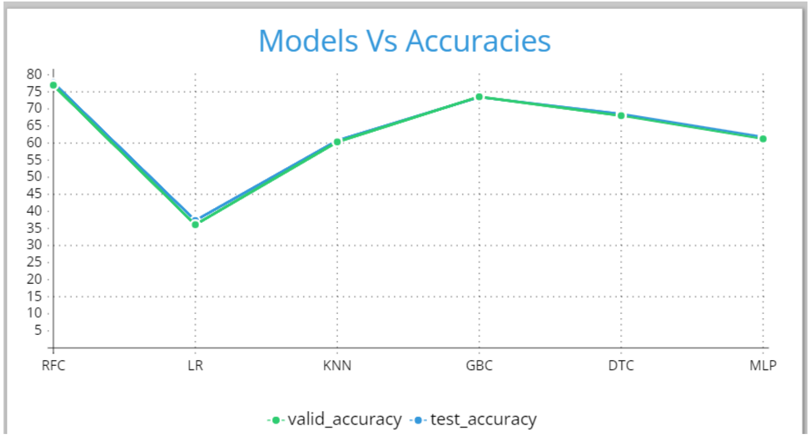


Accuracies of the models for location prediction applying LDA

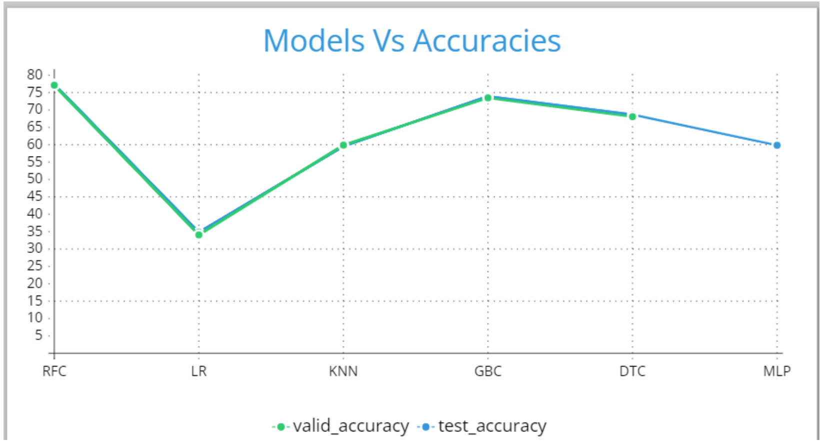


Line graphs depicting valid and test datasets accuracies for the above 3 graphs

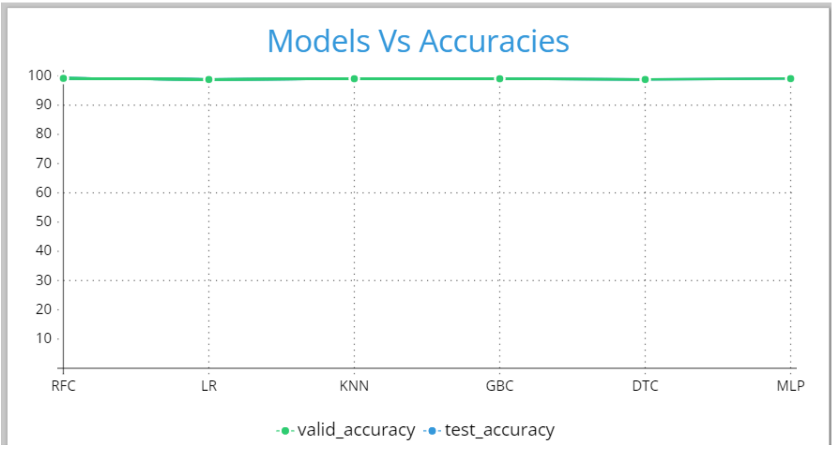
Before applying LDA for severity prediction



After applying LDA for severity prediction



LDA applied data for Location prediction


Models Vs Accuracies
RFC
LR
KNN
GBC
DTC
MLP
● valid_accuracy
● test_accuracy