BRAC UNIVERSITY

Inspiring Excellence

CSE422: Artificial Intelligence

Project Title: Road Accident Severity Prediction

Lab Section: 10
Group: 10
Spring 2024

ID	**Name**
21301470	Syeda Ifroza Ahmed
21301196	Swachcha Podder

Table of Contents

INTRODUCTION

Road accident severity prediction

The road accident severity prediction aims to develop a robust predictive model that can accurately assess the severity of road accidents. By analyzing various features such as Driving_experience, Area_accident_occured, Weather_conditions, and Number_of_casualties, etc, the project seeks to anticipate the potential severity of accidents.

The problem it is aiming to solve

The project addresses the significant issue of road accidents and their associated consequences, including fatalities. By developing a predictive model for road accident severity, the project aims to provide stakeholders with valuable insights to proactively mitigate the impact of accidents and improve road safety measures. The ultimate goal is to provide valuable insights to stakeholders such as law enforcement agencies, transportation authorities, and emergency responders, enabling them to allocate resources more effectively, implement preventive measures, and improve overall road safety.

Motivation Behind the Project

The motivation stems from the urgent need to enhance road safety and reduce the human and economic toll of road accidents. With road accidents posing a continuous threat to public safety worldwide, there's a pressing need for proactive measures to address this issue. By leveraging machine learning models and techniques to predict accident severity, the project aims to empower stakeholders with the tools and information needed to prevent accidents, optimize emergency response efforts, and ultimately save lives.

DATASET DESCRIPTION

Source: Kaggle
Link: **https://www.kaggle.com/datasets/kanuriviveknag/road-accidents-severity-dataset**
Reference: Vivek Nag Kanuri

Dataset description
1. How many features:
There are 10 features initially. Such as: Day_of_week, Age_band_of_driver, Sex_of_driver, Educational_level, Vehicle_driver_relation, Driving_experience, Type_of_vehicle, Owner_of_vehicle, Service_year_of_vehicle, Defect_of_vehicle, Area_accident_occured, Lanes_or_Medians, Road_allignment, Types_of_Junction, Road_surface_type, Road_surface_conditions, Light_conditions, Weather_conditions, Type_of_collision, Number_of_vehicles_involved, Number_of_casualties, Vehicle_movement, Casualty_class, Sex_of_casualty, Age_band_of_casualty, Casualty_severity, Work_of_casuality, Fitness_of_casuality, Pedestrian_movement, Cause_of_accident, Accident_severity.

The outcome has 3 classes, 'Slight Injury', 'Serious Injury', 'Fatal Injury'
After pre-processing there are 17 features.

2. Is this a classification or regression problem? Why do you think so?

For our project, we are addressing a multivalued classification problem. Classification involves the task of predicting categories or classes based on given input features. In the context of road accident severity prediction, our aim is to predict whether features like Driving experience, Weather conditions affect the accident severity. These outcomes, " Serious Injury " and "Slight Injury", "Fatal Injury" are distinct categories that we are trying to predict based on various accident related features. Therefore, the nature of our task, which involves predicting a categorical label (Serious Injury, Slight Injury, Fatal Injury) signifies that our project falls within the domain of classification problems.

3. How many data points?

Initially 12316
After preprocessing 10932

4. What kind of features are in your dataset? Quantitative or categorical?

A combination of quantitative and categorical variables are present in the dataset for road accident severity prediction. Quantitative features are those that involve numerical measurements, such as Age_band_of_driver, Number_of_vehicles_involved, Number_of_casualties etc. These features have values that can be quantified and compared. Categorical features represent different categories or labels. For instance, features like "Area_accident_occured", "Light_conditions" provide descriptions of the area conditions during the accident in categorical terms. Similarly, "Accident_severity" is a categorical feature indicating whether it's serious, slight or fatal. These features help classify accident severity conditions into specific categories based on their descriptions.

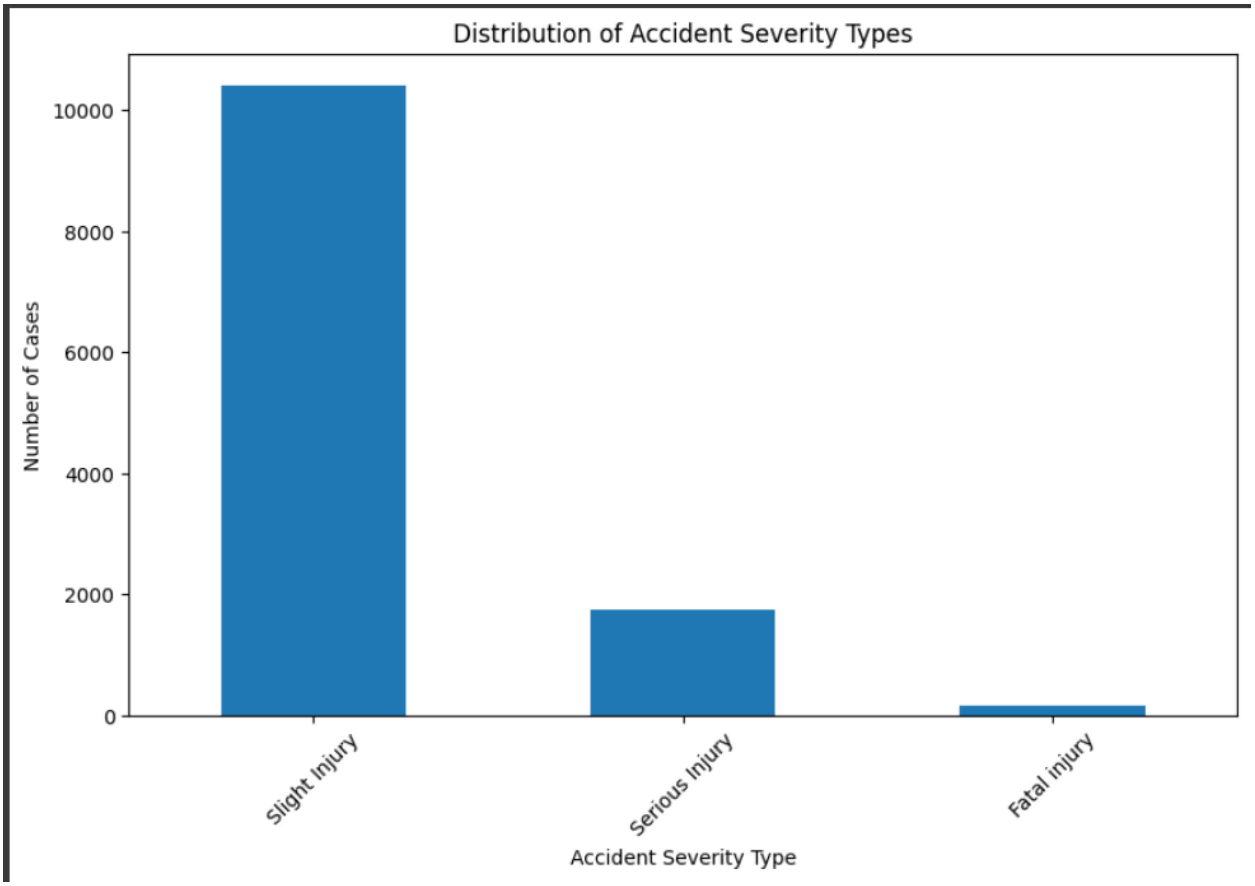
5. Correlation of all features:

Imbalanced Dataset

1. For the output feature, do all unique classes have an equal number of instances or not?

Ans: No

2. Represent using a bar chart of N classes (N=number of classes you have in your dataset).



DATASET PRE-PROCESSING

Faults: Null values

```
1 road.isnull().sum()
```

```
Time                           0
Day_of_week                    0
Age_band_of_driver             0
Sex_of_driver                  0
Educational_level            741
Vehicle_driver_relation      579
Driving_experience           829
Type_of_vehicle              950
Owner_of_vehicle             482
Service_year_of_vehicle     3928
Defect_of_vehicle           4427
Area_accident_occured        239
Lanes_or_Medians             385
Road_allignment              142
Types_of_Junction            887
Road_surface_type            172
Road_surface_conditions        0
Light_conditions               0
Weather_conditions             0
Type_of_collision            155
Number_of_vehicles_involved    0
Number_of_casualties           0
Vehicle_movement             308
Casualty_class                 0
Sex_of_casualty                0
Age_band_of_casualty           0
Casualty_severity              0
Work_of_casuality           3198
Fitness_of_casuality        2635
Pedestrian_movement            0
Cause_of_accident              0
Accident_severity              0
dtype: int64
```

Drop null value columns:

```
road2.drop(['Time','Sex_of_driver','Defect_of_vehicle',
```

```
            'Vehicle_driver_relation','Owner_of_vehicle',
'Service_year_of_vehicle','Lanes_or_Medians','Road_allignment',
'Types_of_Junction','Casualty_severity','Casualty_class',
'Sex_of_casualty','Age_band_of_casualty','Work_of_casuality',
            'Fitness_of_casuality'],axis=1,inplace=True)
```

Drop null value rows:

```
road2 = road2.dropna(axis = 0, subset = ['Vehicle_movement','Type_of_collision','Type_of_vehicle'])
```

Impute values:

```
#Imputing values

from sklearn.impute import SimpleImputer

imputer = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
imputer.fit(road2[['Educational_level']])
road2[['Educational_level']] = imputer.transform(road2[['Educational_level']])

imputer = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
imputer.fit(road2[['Driving_experience']])
road2[['Driving_experience']] = imputer.transform(road2[['Driving_experience']])

imputer = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
imputer.fit(road2[['Area_accident_occured']])
road2[['Area_accident_occured']] = imputer.transform(road2[['Area_accident_occured']])

imputer = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
imputer.fit(road2[['Road_surface_type']])
road2[['Road_surface_type']] = imputer.transform(road2[['Road_surface_type']])
```

```
road2.isnull().sum()
```

Fault: Categorical values

```
1 road.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12316 entries, 0 to 12315
Data columns (total 32 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   Time                     12316 non-null  object
 1   Day_of_week              12316 non-null  object
 2   Age_band_of_driver       12316 non-null  object
 3   Sex_of_driver            12316 non-null  object
 4   Educational_level        11575 non-null  object
 5   Vehicle_driver_relation  11737 non-null  object
 6   Driving_experience       11487 non-null  object
 7   Type_of_vehicle          11366 non-null  object
 8   Owner_of_vehicle         11834 non-null  object
 9   Service_year_of_vehicle  8388 non-null   object
 10  Defect_of_vehicle        7889 non-null   object
 11  Area_accident_occured    12077 non-null  object
 12  Lanes_or_Medians         11931 non-null  object
 13  Road_allignment          12174 non-null  object
 14  Types_of_Junction        11429 non-null  object
 15  Road_surface_type        12144 non-null  object
 16  Road_surface_conditions  12316 non-null  object
 17  Light_conditions         12316 non-null  object
 18  Weather_conditions       12316 non-null  object
 19  Type_of_collision        12161 non-null  object
 20  Number_of_vehicles_involved  12316 non-null  int64 
 21  Number_of_casualties     12316 non-null  int64 
 22  Vehicle_movement         12008 non-null  object
 23  Casualty_class           12316 non-null  object
 24  Sex_of_casualty          12316 non-null  object
 25  Age_band_of_casualty     12316 non-null  object
 26  Casualty_severity        12316 non-null  object
 27  Work_of_casuality        9118 non-null   object
 28  Fitness_of_casuality     9681 non-null   object
 29  Pedestrian_movement      12316 non-null  object
 30  Cause_of_accident        12316 non-null  object
 31  Accident_severity        12316 non-null  object
dtypes: int64(2), object(30)
memory usage: 3.0+ MB
```

The object types are qualitative data while the int type data are quantitative.

Encoding:

It is necessary to replace categorical data with numerical data so that the computer can understand and recognize a pattern to make a prediction.

```
#Encoding features

from sklearn.preprocessing import LabelEncoder

# List of columns to label encode
columns_to_encode = ['Day_of_week',
 'Age_band_of_driver',
 'Educational_level',
 'Driving_experience',
 'Type_of_vehicle',
 'Area_accident_occured',
 'Road_surface_type',
 'Road_surface_conditions',
 'Light_conditions',
 'Weather_conditions',
 'Type_of_collision',
 'Vehicle_movement',
 'Pedestrian_movement',
 'Cause_of_accident',
 'Accident_severity']

# Initialize LabelEncoder
label_encoder = LabelEncoder()

# Apply label encoding to each column
for column in columns_to_encode:
    road2[column] = label_encoder.fit_transform(road2[column])
```

FEATURE SCALING

One of the problems that the algorithm often faces is that it gets biased towards larger values. To avoid this problem, we scale the numerical data to bring it in a small range.

```
x = road2.drop(['Accident_severity'], axis = 1)
y = road2[['Accident_severity']]
```

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

scaler.fit(x)
x = scaler.transform(x)
```

```
'''from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x,y,test_size=0.30, random_state=0, stratify = y)'''
```

```
'from sklearn.model_selection import train_test_split\nxtrain, xtest, ytrain, ytest = train_test_split(x,y,test_size=0.30, random_state=0, stratify = y)'
```

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x,y,test_size=0.30)
```

```
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score, f1_score, ConfusionMatrixDisplay, classi
```

DATASET SPLITTING

The data from the dataset was split into 70% for the training model, and 30% for the testing model.

MODEL TRAINING

In total, six machine learning models have been used for training in both scenarios of before oversampling and after oversampling. They are- Naive Bayes, Decision Tree, K-nearest Neighbors, Random Forest, Support Vector Machine, and MLP.

MODEL TESTING

After training, the six mentioned models were tested.
Before oversampling, the results of performance metrics like accuracy, precision, recall, F1-Score are given below-

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Naive Bayes	82.74	76.60	82.74	76.59
Decision Tree	74.09	74.97	74.09	74.52
KNN	80.49	74.93	80.49	76.98
Random Forest	84.27	80.66	84.27	78.30
SVM	84.05	85.50	84.05	76.80
MLP	83.57	76.46	83.57	79.00

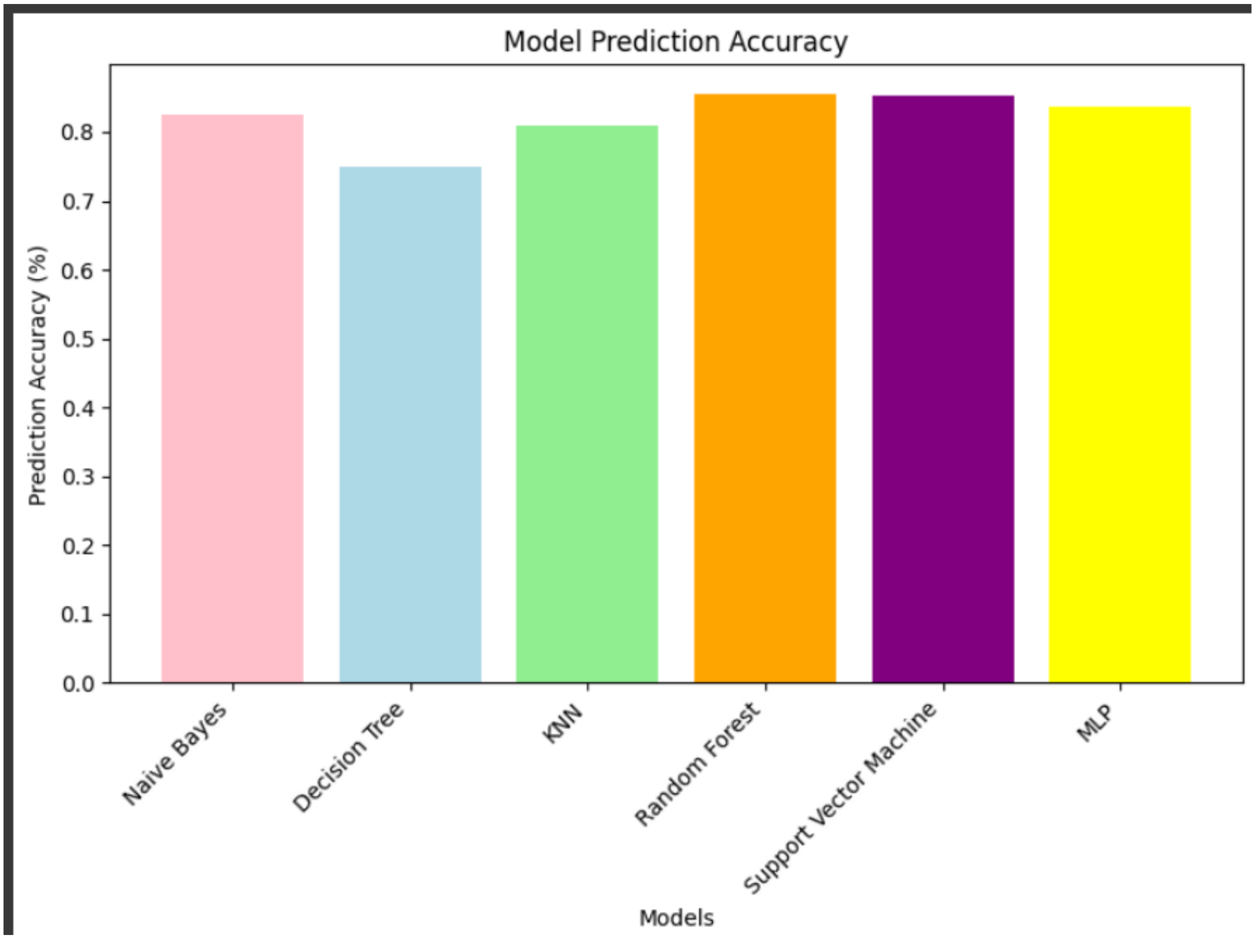
After oversampling, the results of performance metrics like accuracy, precision, recall, F1-Score are given below-

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Naive Bayes	43.93	43.70	43.93	42.15
Decision Tree	92.91	93.69	92.91	92.80
KNN	87.80	89.31	87.80	87.46
Random Forest	97.69	97.74	97.69	97.69
SVM	69.71	68.96	69.71	69.12
MLP	78.54	78.29	78.54	78.37

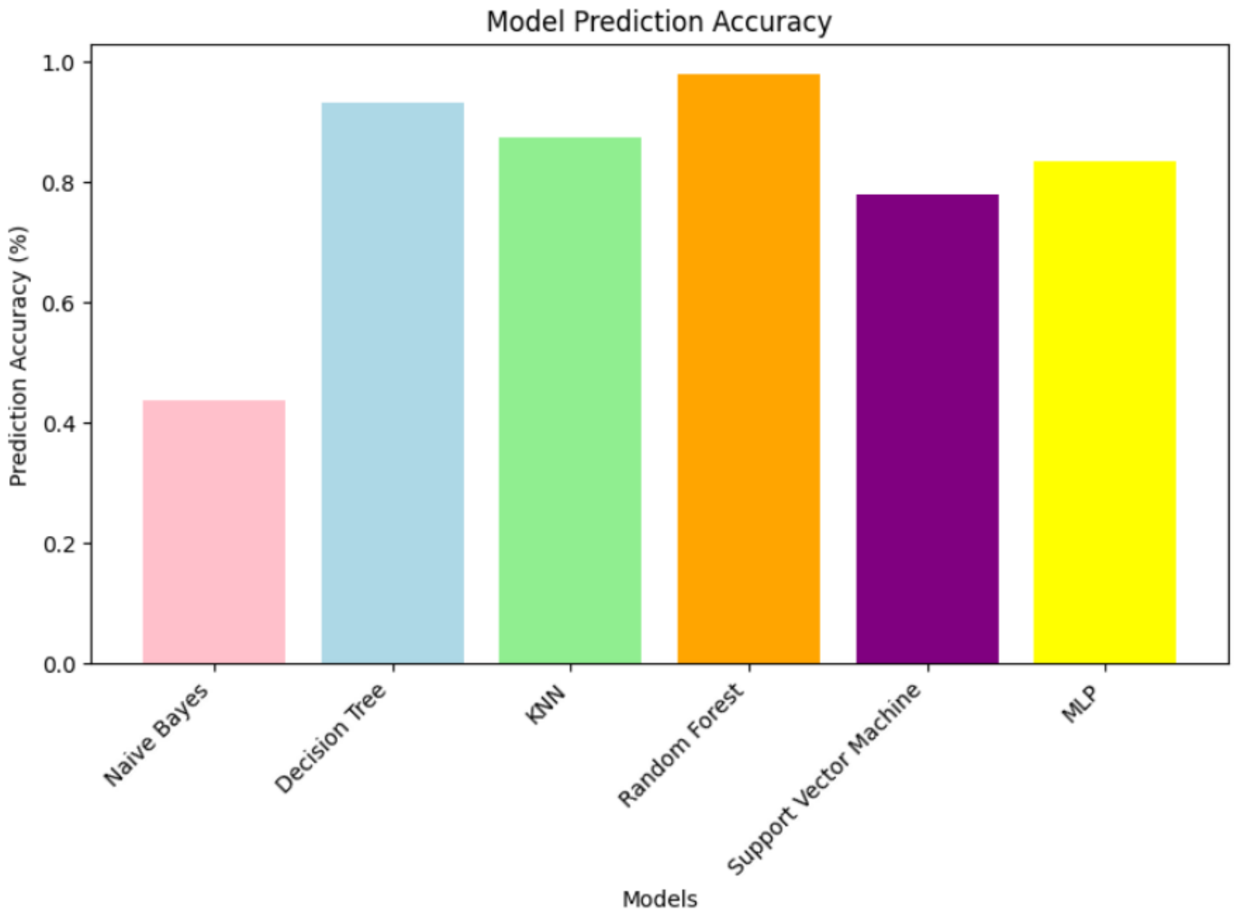
MODEL SELECTION/COMPARISON ANALYSIS

Barchart showcasing prediction accuracy of all models-

Before oversampling-



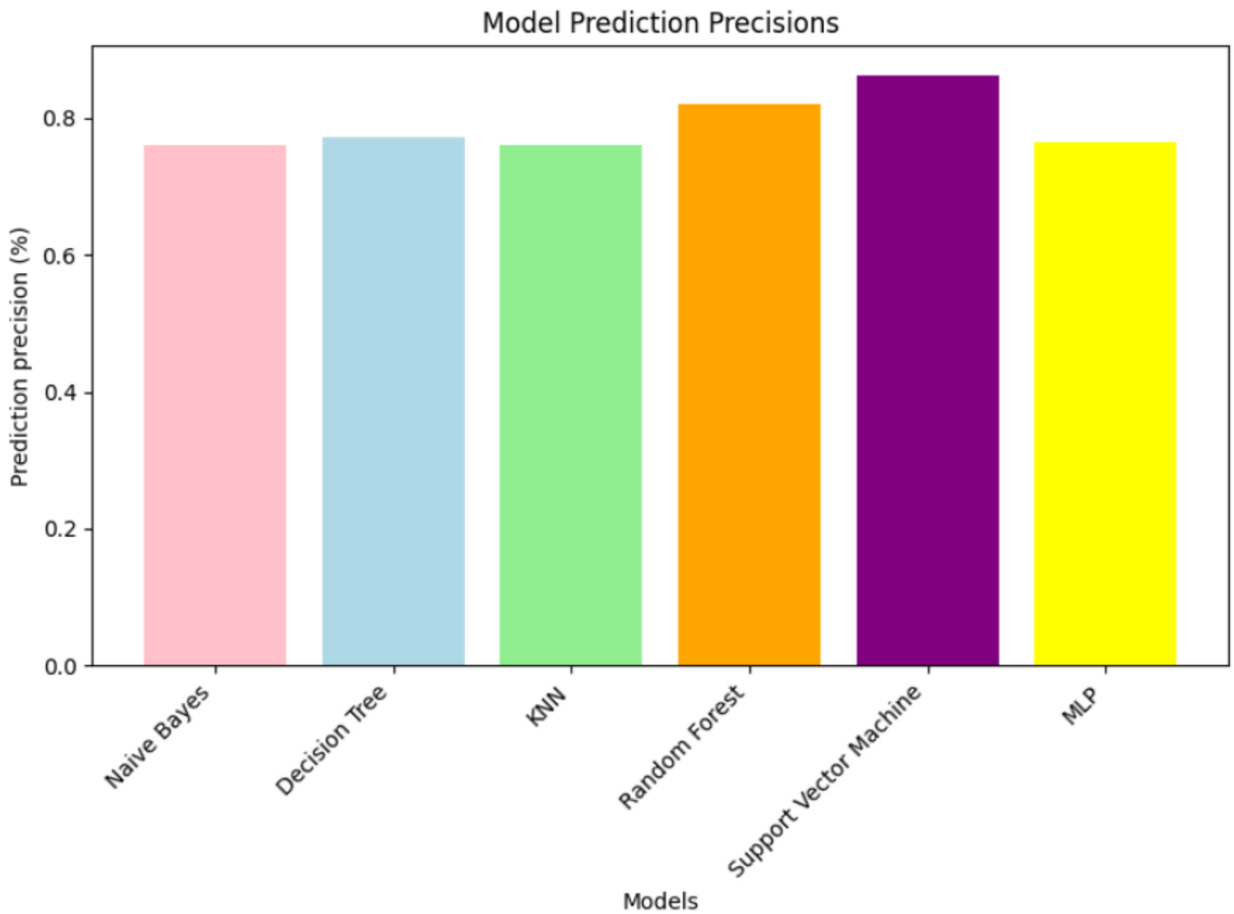
After oversampling-



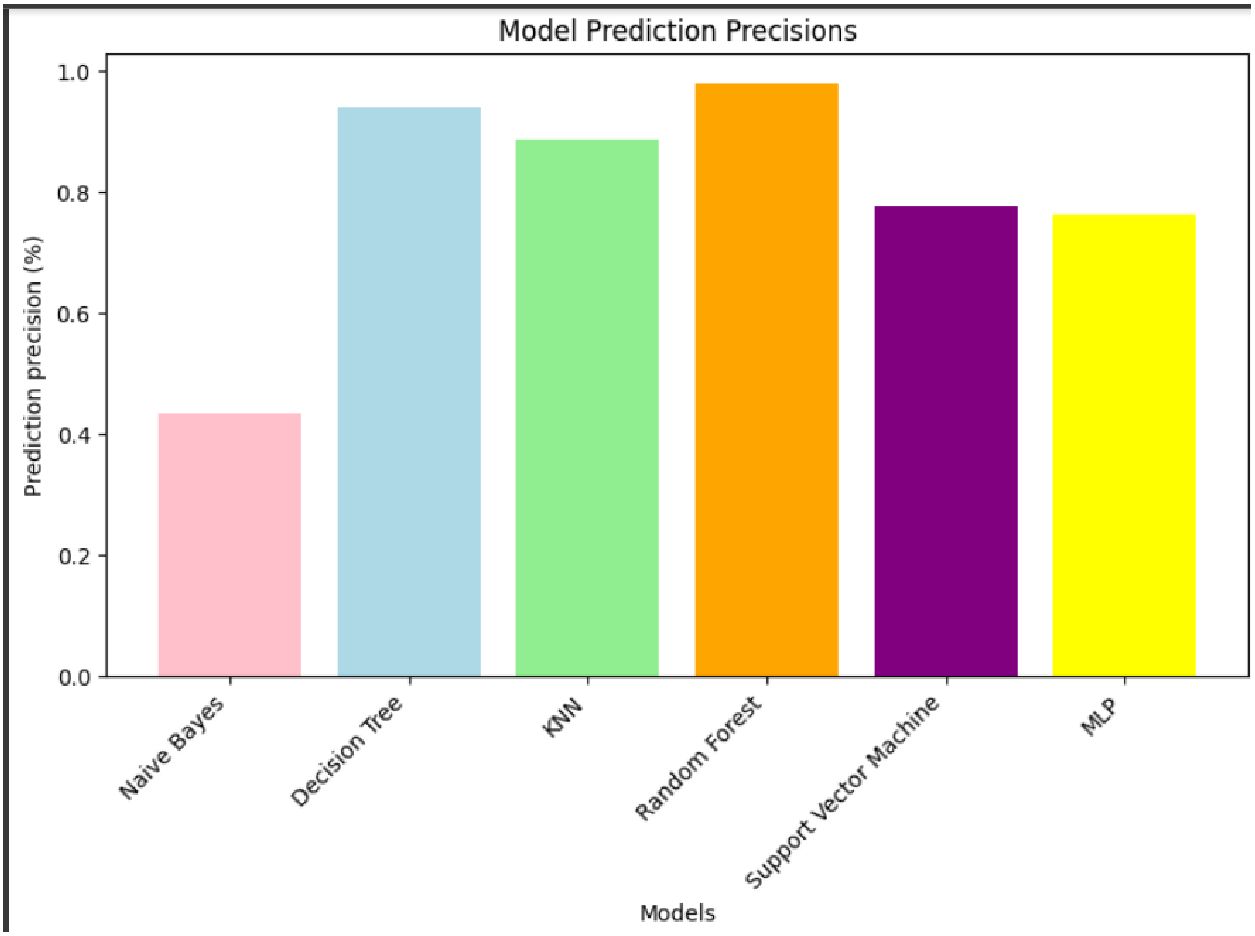
It can be seen that the accuracies before oversampling were above 0.70 for all the models. After oversampling, the accuracy of Naive Bayes, Support Vector Machine, MLP dropped to below 0.50, 0.70, 0.80 respectively, but on the other hand the accuracies of Decision Tree, Random Forest were boosted to above 0.90, and KNN jumped to 0.87.

Precision comparison of each model:

Before oversampling-



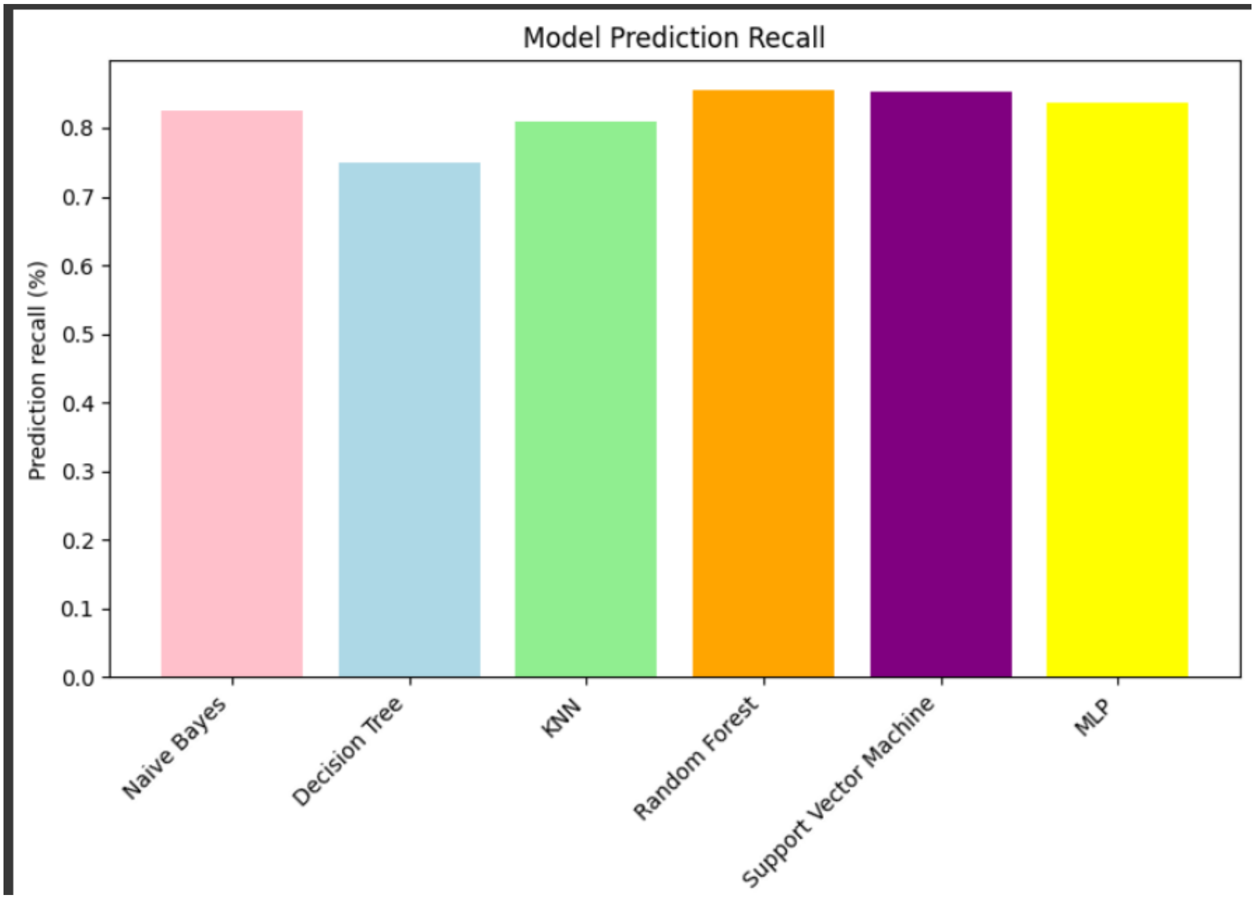
After oversampling-



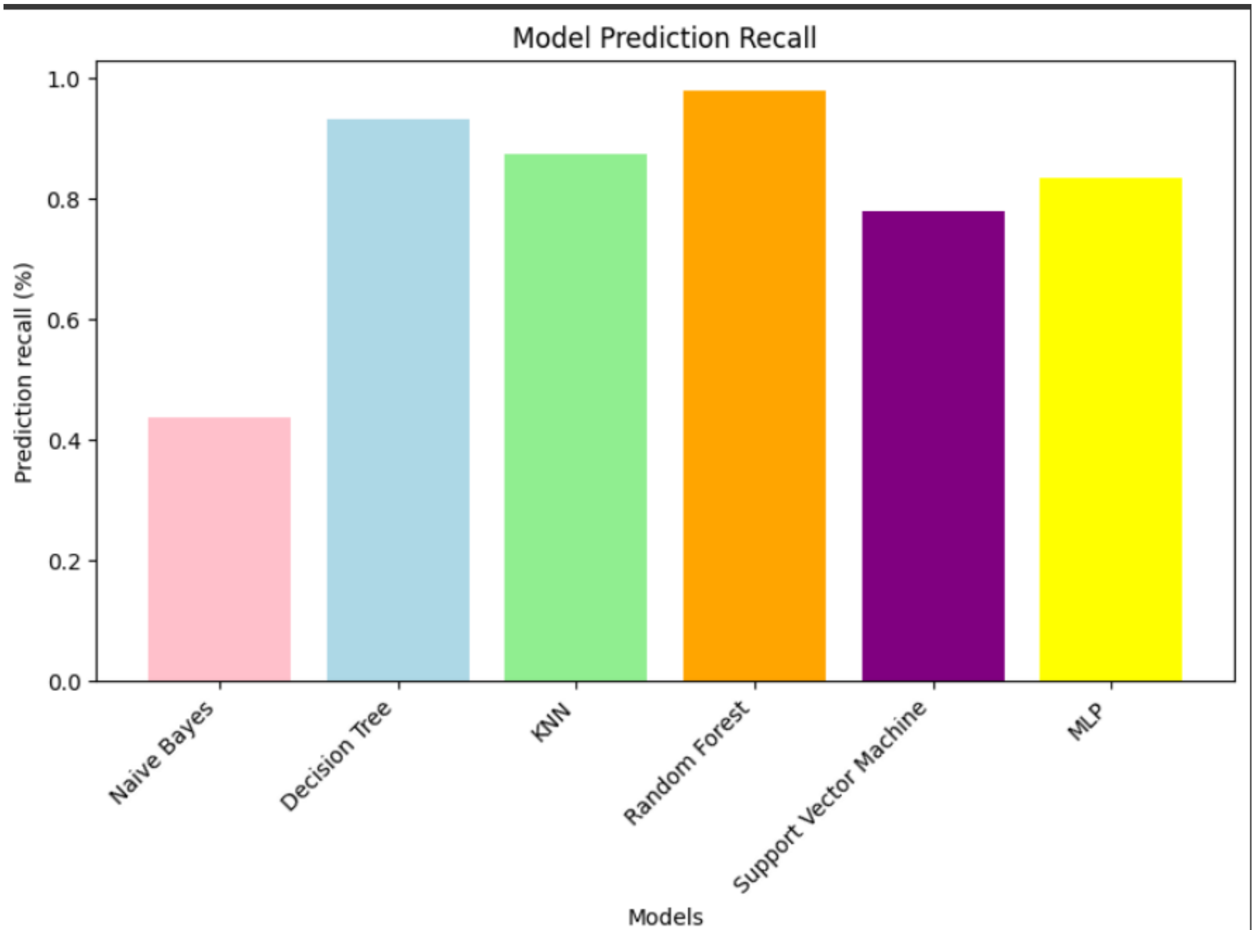
It is evident that before oversampling, the recall for each of the six models is above 0.70. But after oversampling, the precision of Naive Bayes and Support Vector Machine decreased significantly to below 0.50 and 0.70, respectively. While the other four models were able to reach precision levels of around 0.78 (MLP), 0.87 (KNN) and above 0.90 (Decision Tree, Random Forest).

Recall comparison of each model:

Before oversampling:



After oversampling:

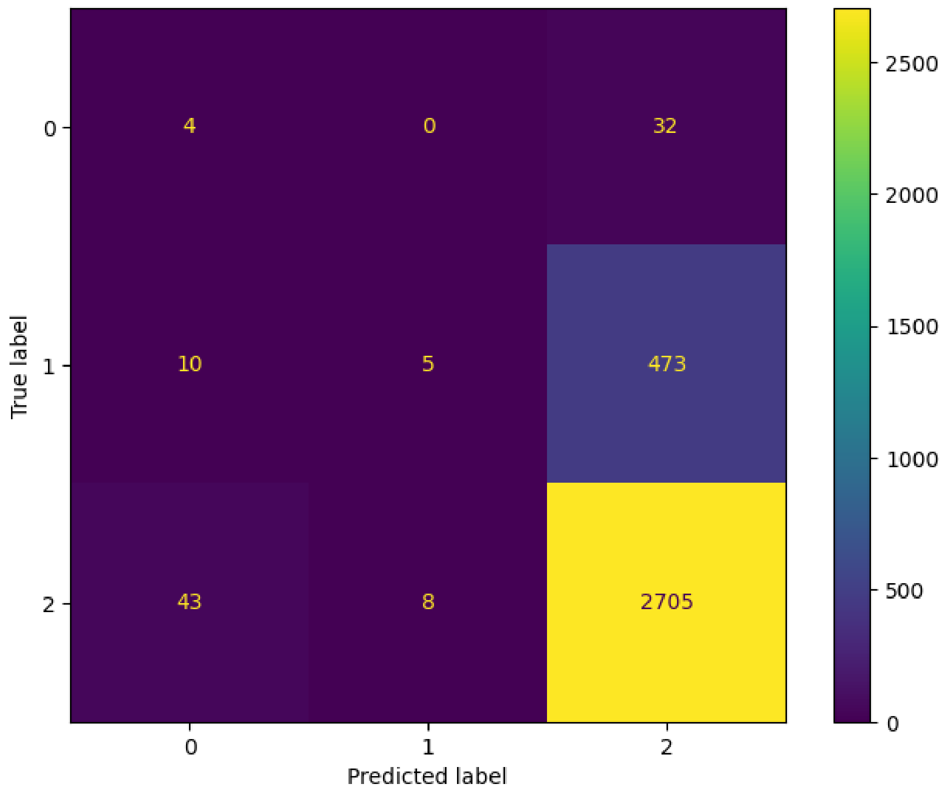


It is evident that before oversampling, the recall for each of the six models is above 0.70. But after oversampling, the recall of Naive Bayes, Support Vector Machine, MLP decreased significantly to below 0.50, 0.70, and 0.80 respectively. While the other three models were able to reach recall levels of around 0.87 (KNN) and above 0.90 (Decision Tree, Random Forest).

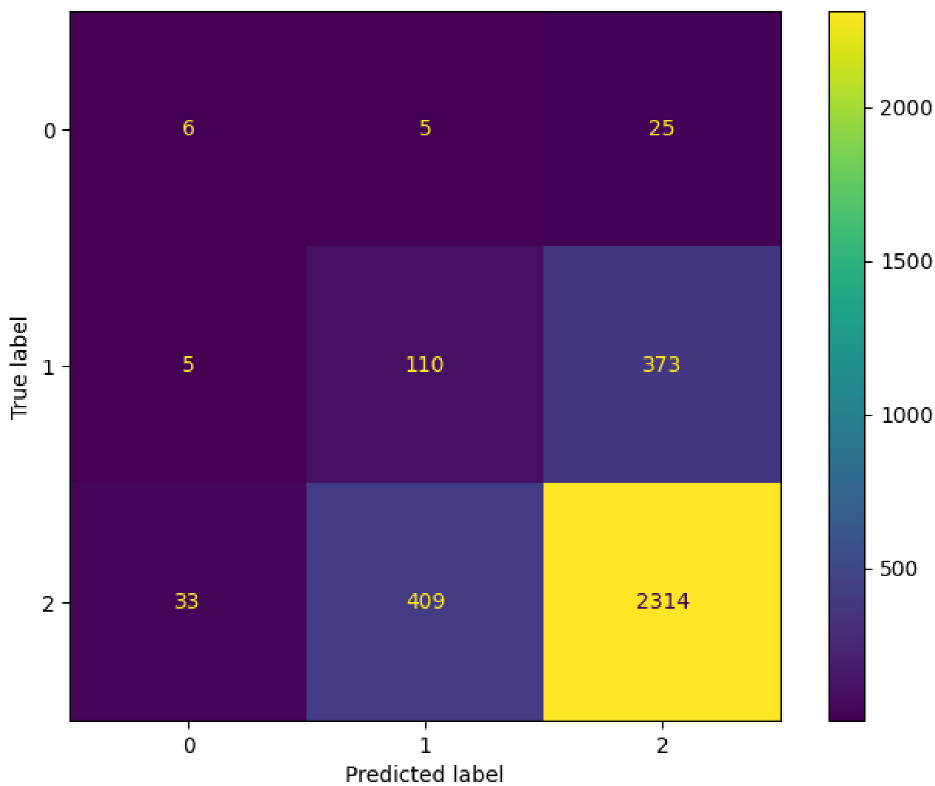
Confusion Matrix:

Before oversampling:

Naive Bayes:

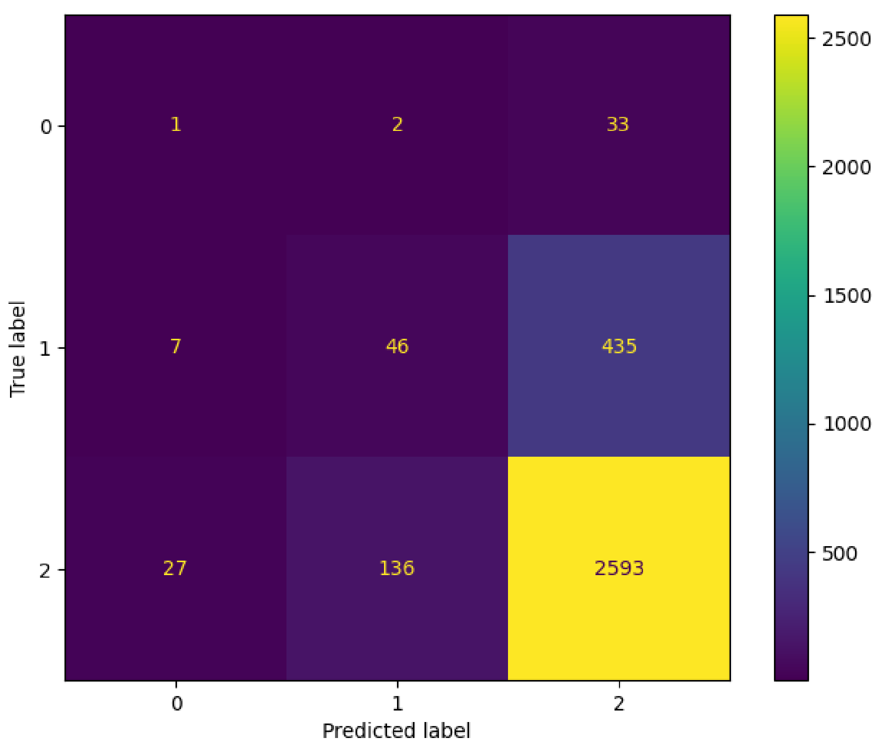


Decision Tree:



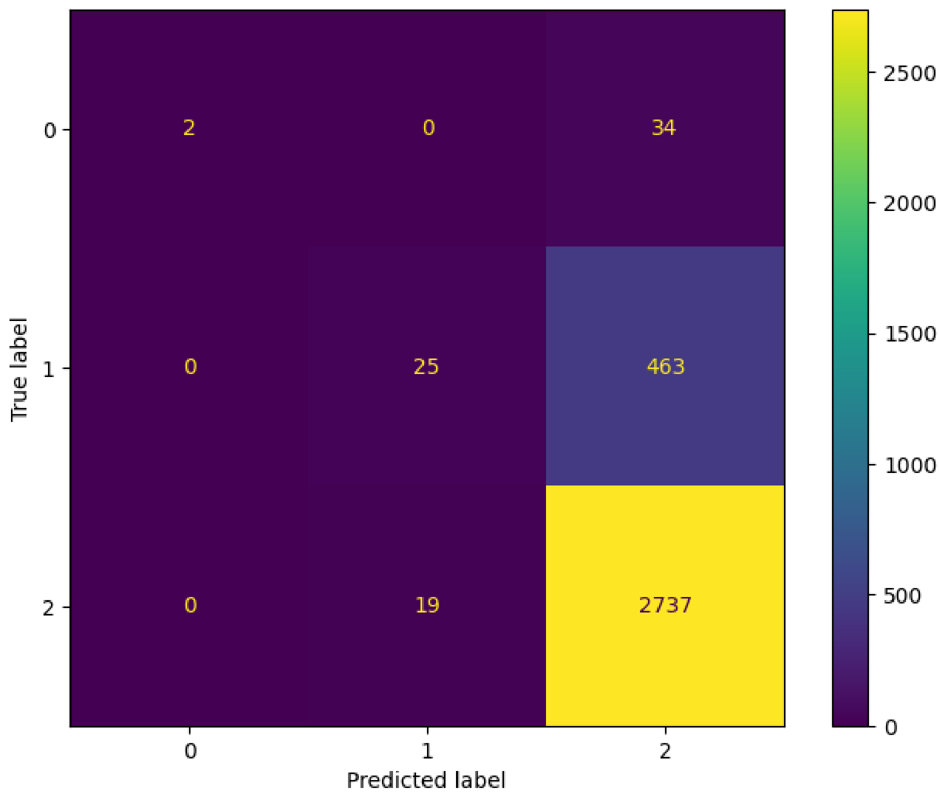
True label
Predicted label
0
1
2
6
5
25
5
110
373
33
409
2314
2000
1500
1000
500

KNN:

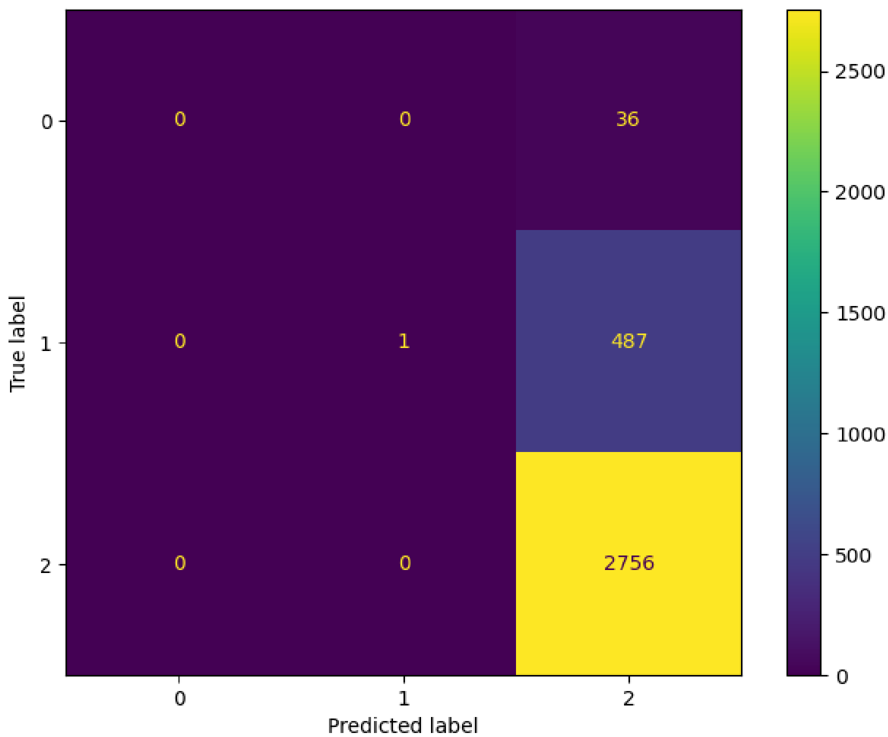


True label
Predicted label
0
1
2
1
2
33
7
46
435
27
136
2593
2500
2000
1500
1000
500

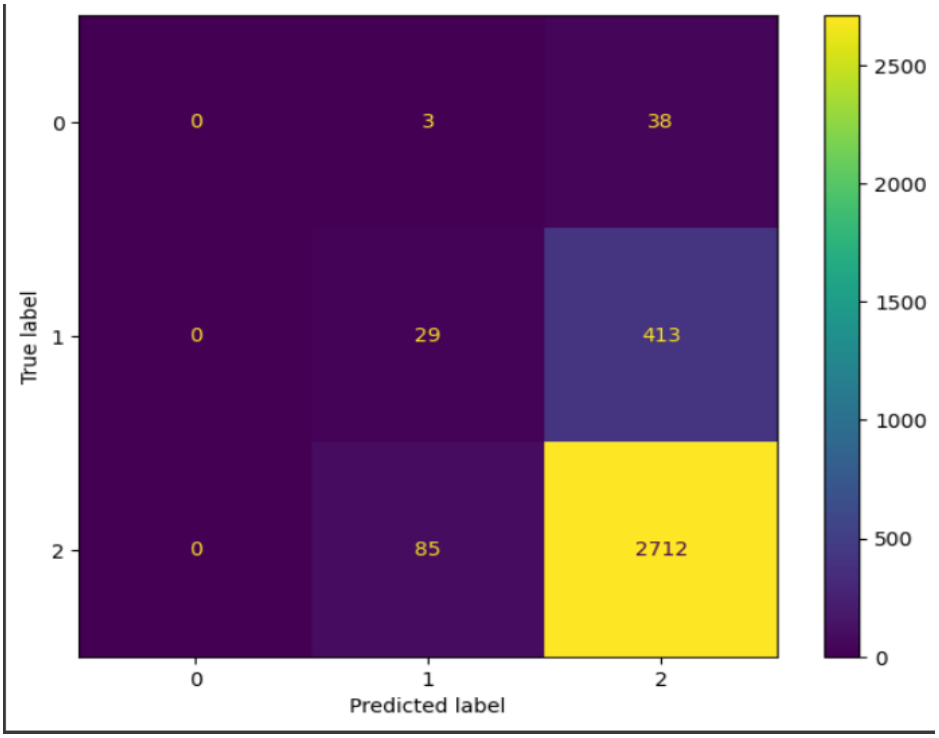
Random Forest:



SVM:

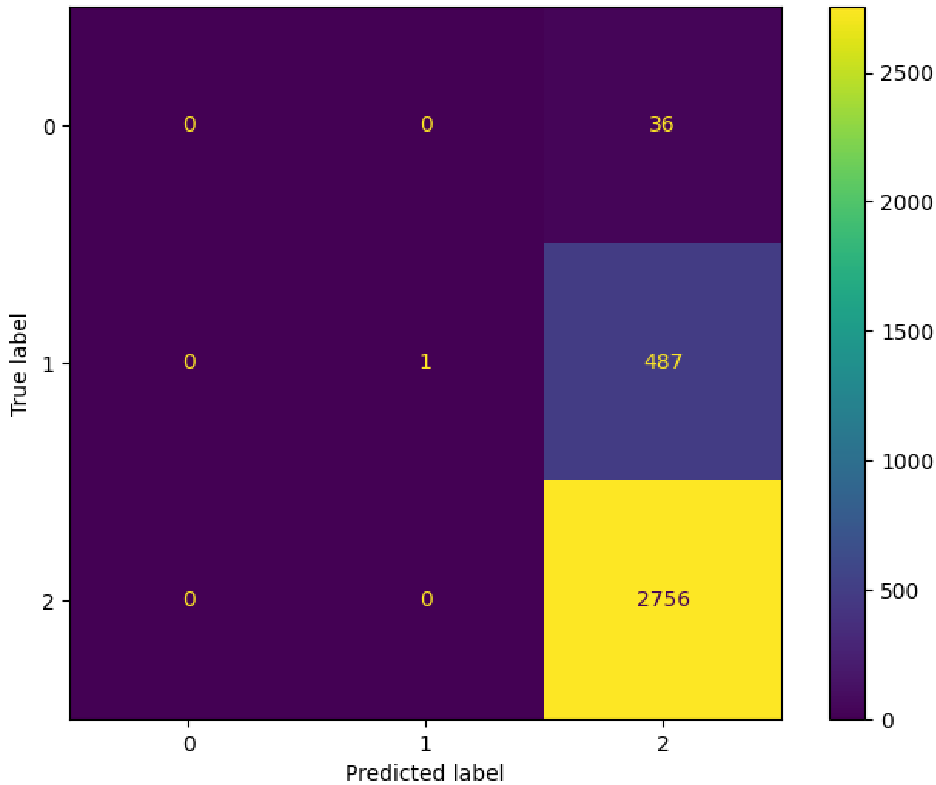

0
0
36
0
1
487
0
0
2756
True label
Predicted label
0
1
2
0
1
2
2500
2000
1500
1000
500
0

MLP:

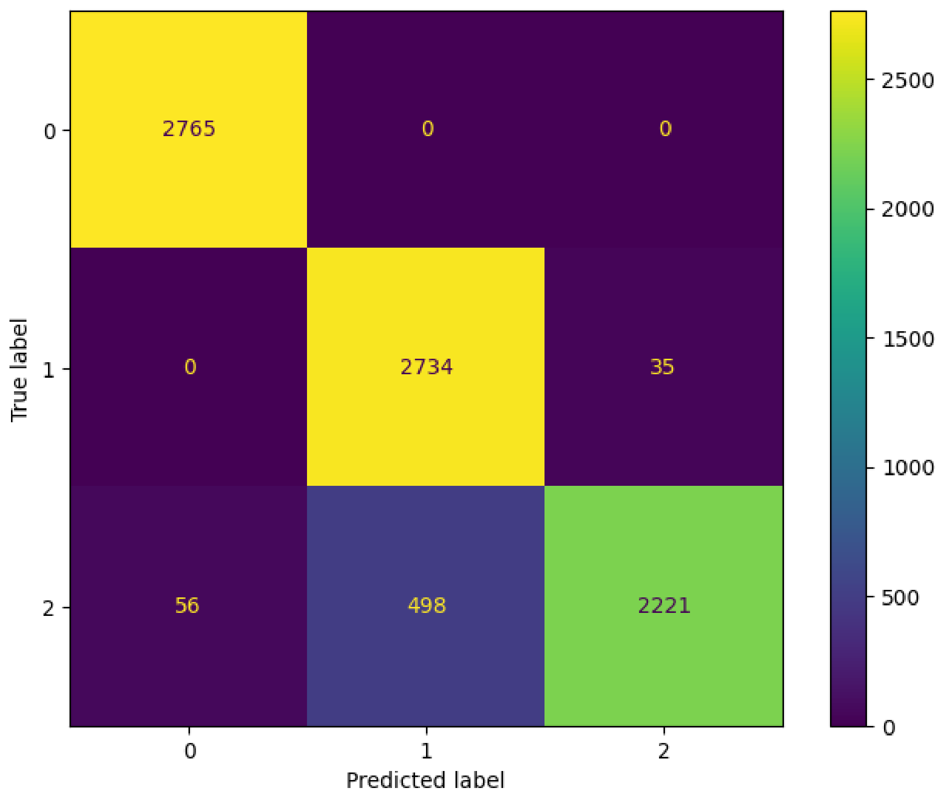


After oversampling:

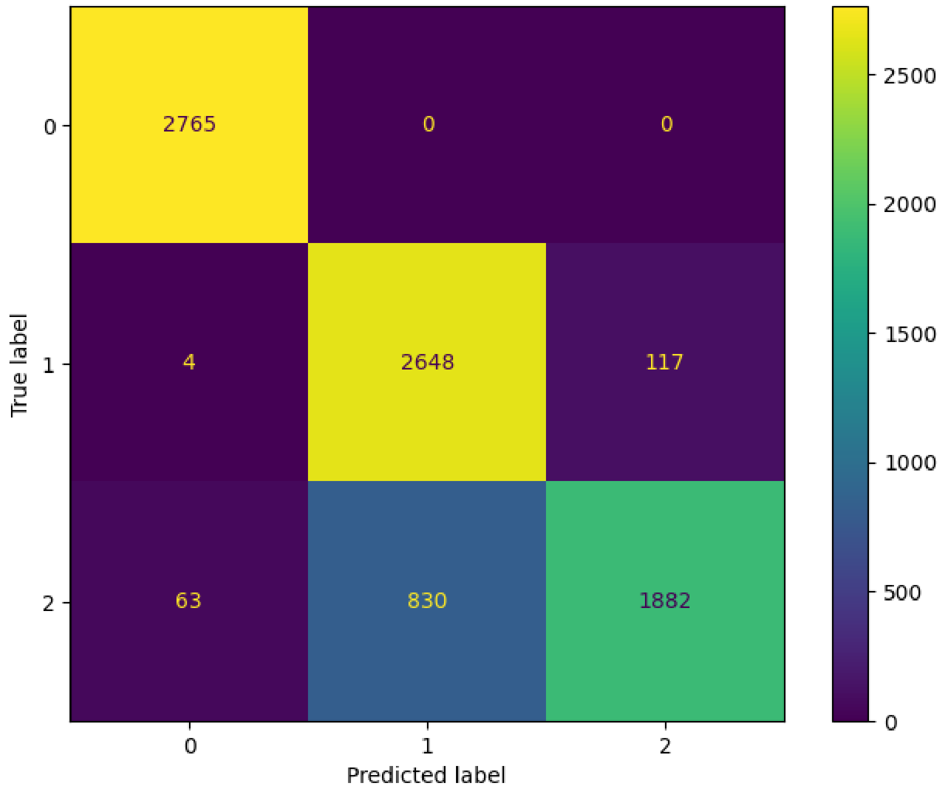
Naive Bayes:



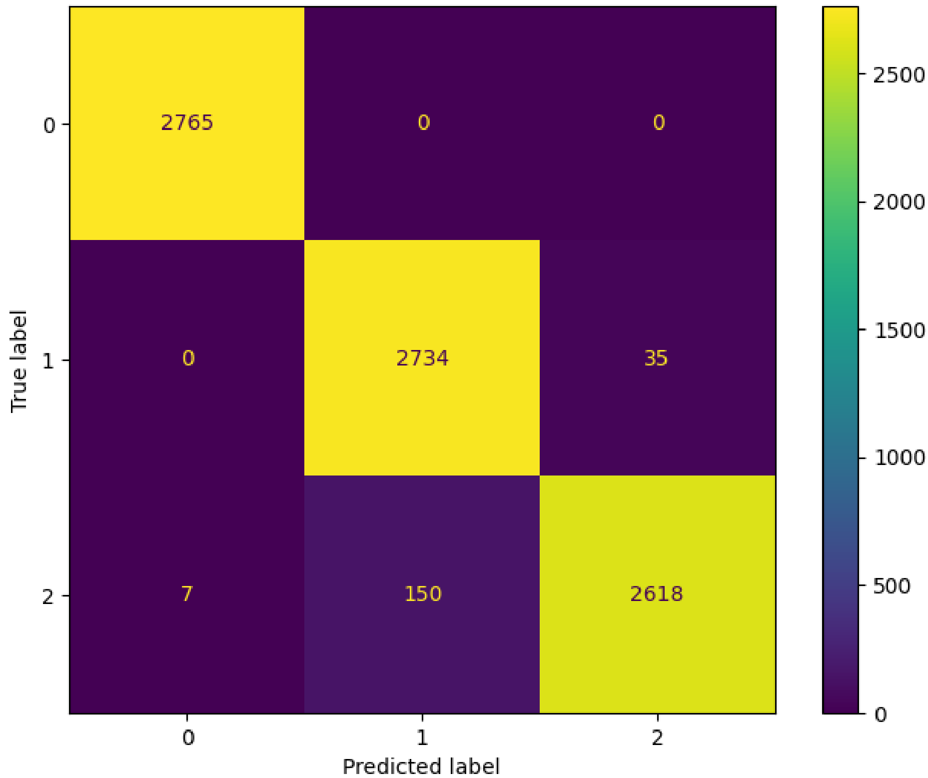
Decision Tree:


True label
0
1
2
Predicted label
0
1
2
2765
0
0
0
2734
35
56
498
2221
2500
2000
1500
1000
500
0

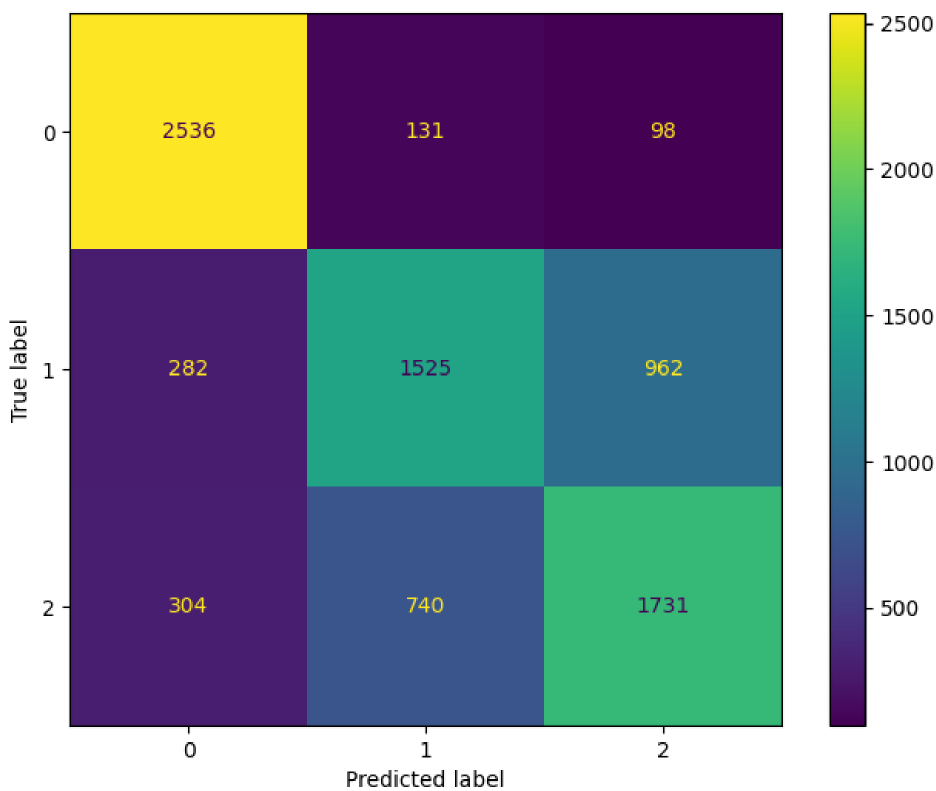
KNN:



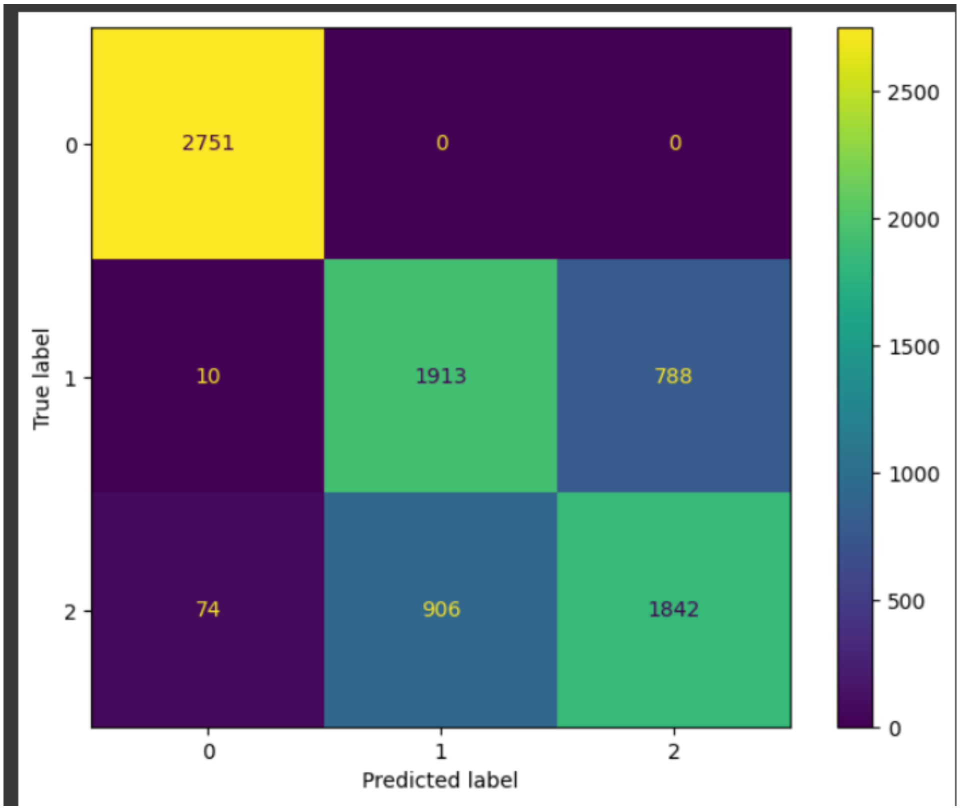
Random Forest:


2765
0
0
0
2734
35
7
150
2618
True label
Predicted label
0
1
2
2500
2000
1500
1000
500

SVM:


2536
131
98
282
1525
962
304
740
1731
True label
Predicted label
0
1
2
2500
2000
1500
1000
500

MLP:



True label
Predicted label
2751
0
0
10
1913
788
74
906
1842
2500
2000
1500
1000
500

CONCLUSION

A total of six machine learning algorithms were run, results before oversampling and after oversampling were recorded. Obtained results were accuracies, precisions, recall and F1-score of above 0.70 from scenarios of before oversampling. After the sampling the results of accuracy recorded varied significantly as three of the models were now able to reach 0.90 excluding Naive Bayes, SVM, MLP which performed worse than the other models.

As a result of the insights gained from this study, machine learning in road accident severity assessment can be implemented practically in the future. This development holds substantial ramifications in terms of enhancing global public awareness. In essence, AI holds promise in revolutionizing road safety measures and ensuring the fundamental human right to safe mobility. Through the proactive use of AI technologies, we can strive towards a future where road accidents are minimized, and every individual has the opportunity to travel safely.