CSE422 : Artificial Intelligence
Project Report

Project Title: Traffic Prediction

Name	ID
Pulak Deb Roy	**23241078**
Jaki Ahmed	**19301161**

1. Introduction

The goal of this project is to create a model for precise prediction of traffic. The goal is simple: using both historical and current data, anticipate traffic flow and congestion levels. By examining these datasets, we offer insightful information that helps commuters and urban planners make better decisions and enhance traffic control tactics in general.

2. Dataset description:

Source: Kaggle

Link: https://www.kaggle.com/datasets/hasibullahaman/traffic-prediction-dataset/data

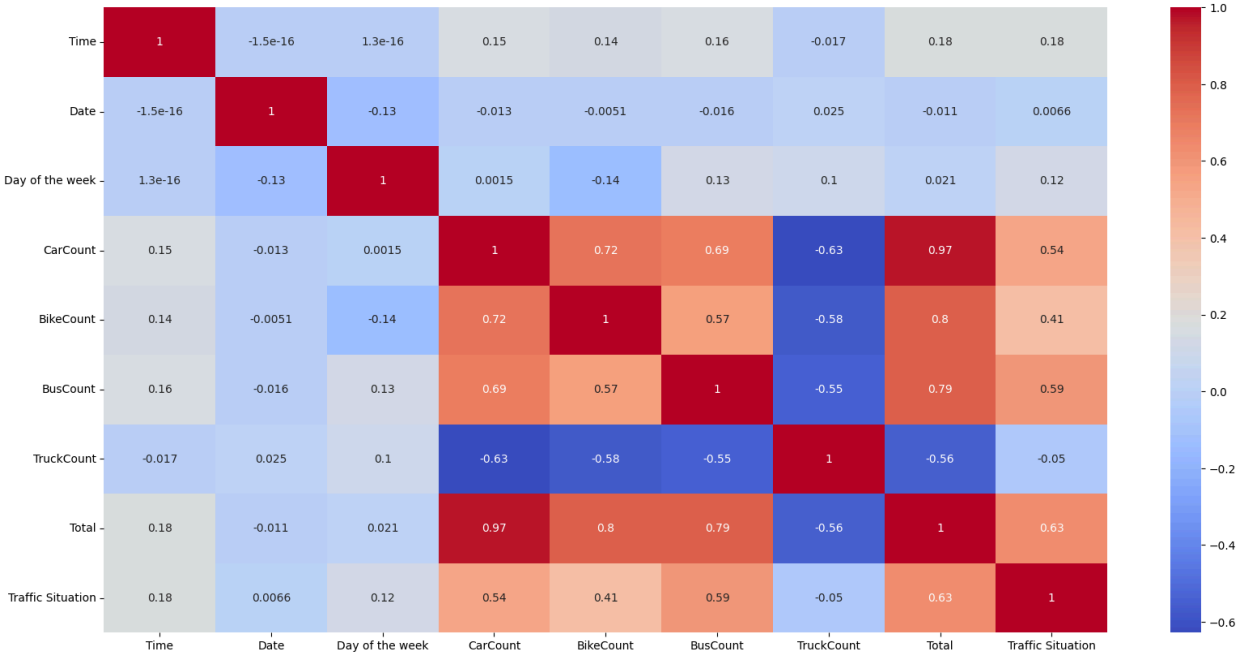
Dataset Description:

a. How many features: There are 9 features in the dataset.
b. Classification or regression problem: Classification, as the task involves classifying the output into one of four categories: Low, Normal, Heavy, High. This is evident from the need to categorize individuals based on the traffic situation, making it a classification problem.
c. How many data points: 2976 rows, column 9.
d. Features in dataset:

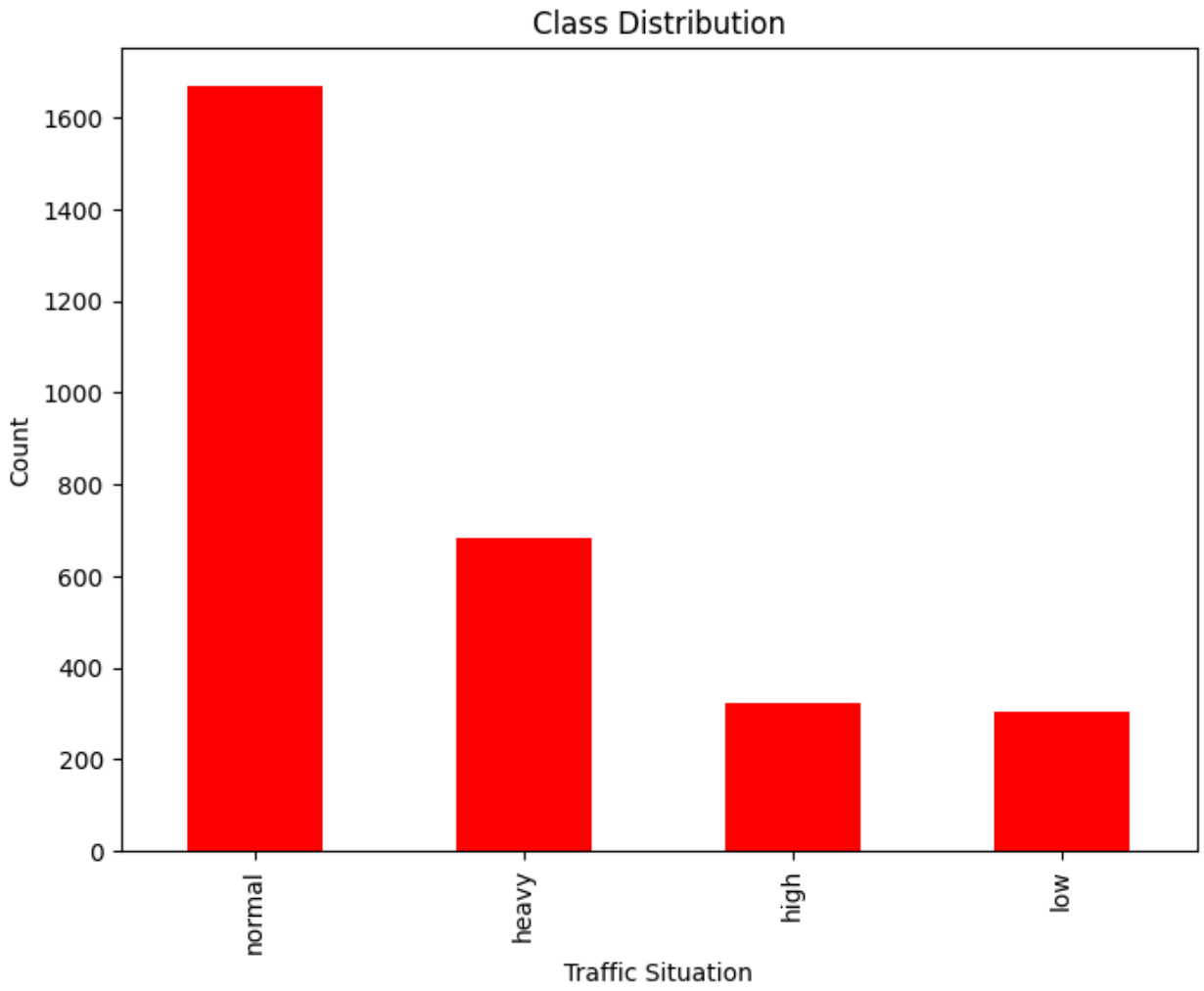
- Quantitative Features: These include numerical data which are: Date, CarCount, BikeCount, BusCount, TruckCount, Total.

- Categorical Features: These include non-numerical data which are: Time, Day of the week, Traffic Situation.

- Correlation of all the features:



Output Feature Distribution: All unique classes do not have an equal number of instances. This indicates an imbalance dataset where “Normal” class is overrepresented compared to “heavy”, “high” and “low” classes.



Output Feature Distribution: All unique classes do not have an equal number of instances.

3. Dataset pre-processing

• Faults

- NULL values: There were no none values in the dataset.
- Categorical values: "Day of the week", "Time", "Traffic Situation" columns had categorical values.
- Duplicate values: There were no duplicate values present in the dataset.

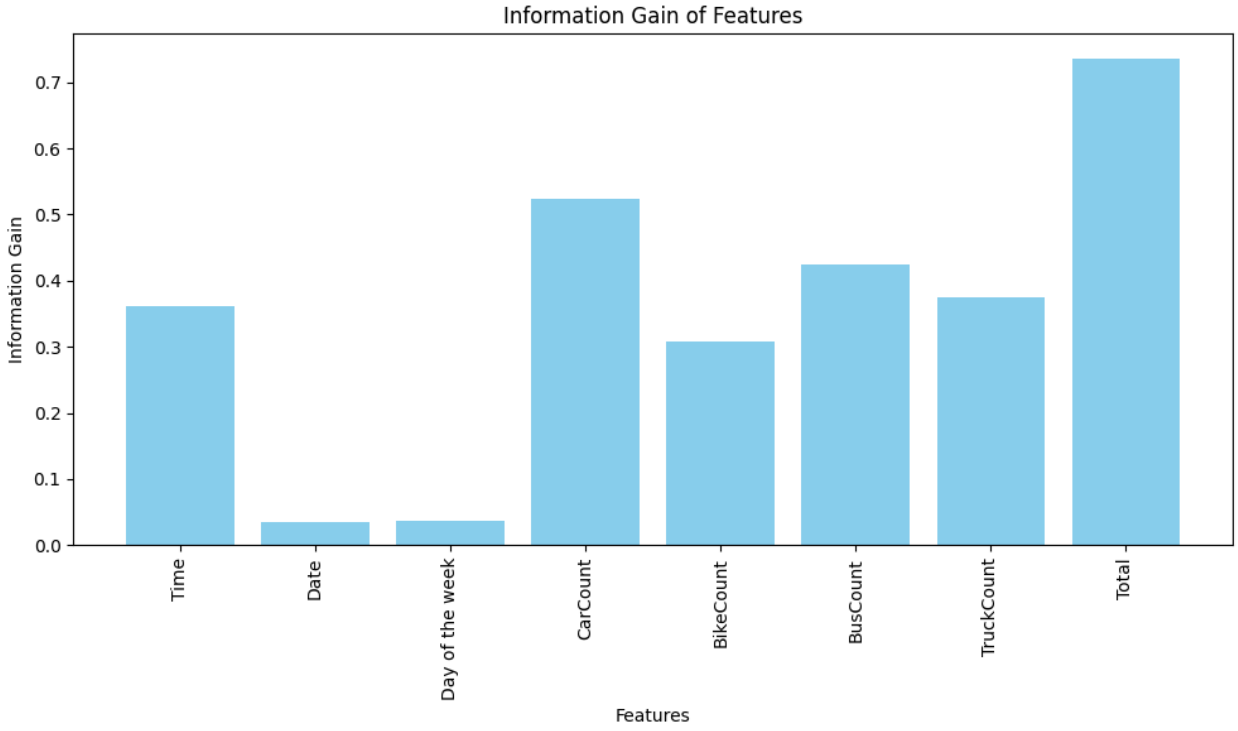
• Solutions

- Delete rows/columns, Impute values: As we didn't have any null values in any data, we didn't delete any rows. And there were no irrelevant columns that needed to be deleted.
- Encoding: The categorical features had values in different categories which were encoded into discrete values for applying machine learning algorithms.

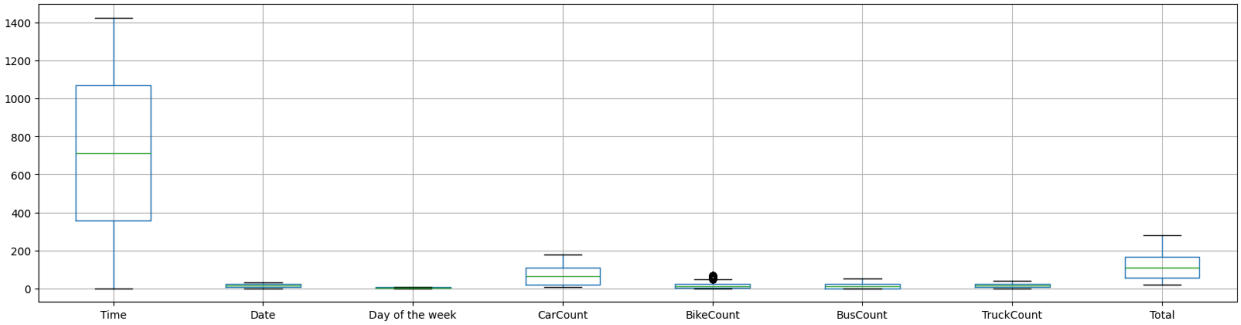
In the "Day of the week" feature, there were 7 days of the week in texts such as "Saturday", "Sunday", "Monday" etc. We assigned it in discrete values.
Similarly, "Time" and "Traffic Situation" were also encoded into necessary discrete values.

4. Feature selection and scaling (as required):

To keep relevant and useful features for our model, we plotted a graph that shows how much information we can gain from each feature, meaning how pure a feature is. However, we didn't remove any feature as our algorithms already provided a good accuracy.



We also didn't need to scale any features in our data as the outliers present were negligible. This was shown in boxplot.



5.Dataset Splitting

We stratified the label while splitting the dataset. Stratify ensures equal distribution of labels which helps the machine predict the result more accurately. We used 70% of the data for training and 30% of the data for testing.

6. Model training & testing

At the beginning of our project, we mentioned that the problem we are dealing with is a classification problem. So, it is obvious classifiers will work better on our dataset. We chose the following three classifier models:

- **KNN:** KNN is a simple and intuitive algorithm used for classification and regression taks, It works based on the idea that similar data points tend to belong to the same class or have similar target values. It's easy to understand, works well with small to moderately sized datasets. It can handle non-linear relationships in the data.

- **Decision Tree:** Decision trees are versatile algorithms used for classification and regression tasks, They partition the feature space into regions and make decisions based on the values of features. Decision trees are highly interpretable and easy to visualize.

They can handle both numerical and categorical data and can capture nonlinear relationships in the data.

- **Random Forest:** Random forest is an ensemble learning method based on decision trees. It builds multiple decision trees and combines their predictions to improve accuracy and robustness. Random forest is robust to overfitting, as it combines the prediction of multiple trees. It performs well with high-dimensional data and can handle missing values and outliers.

7. Model selection/Comparison analysis

Accuracy, precision, recall, F1 score for each model:

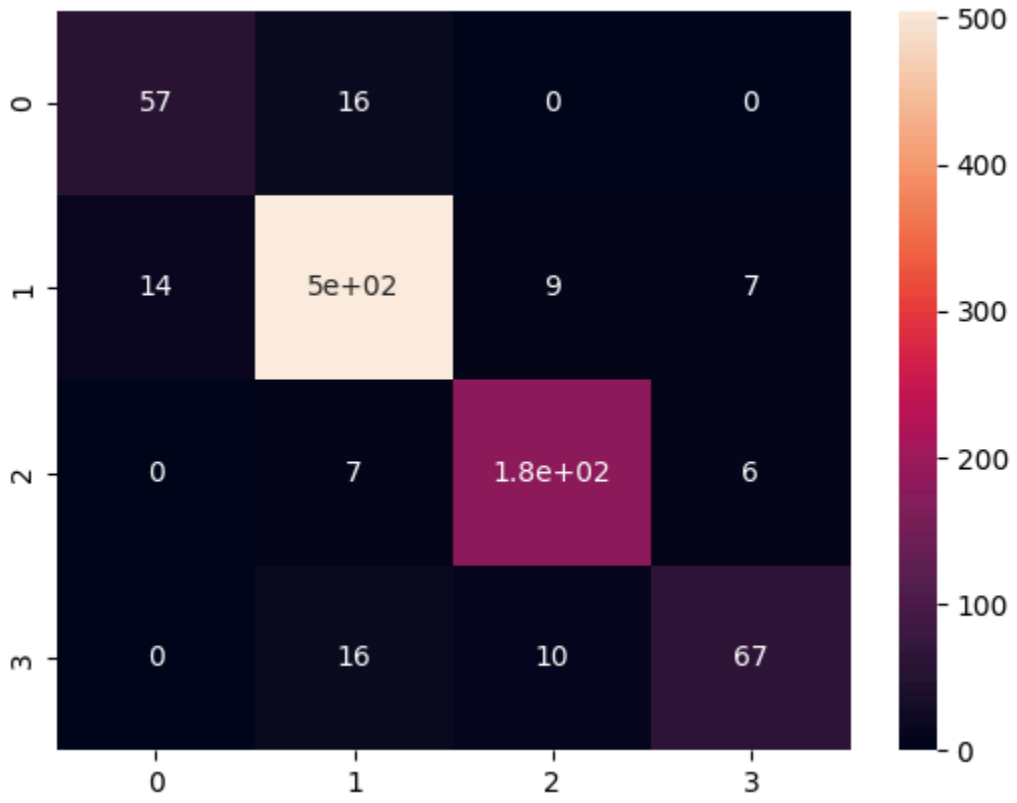
K-Nearest Neighbors:

Accuracy: 90%

Precision: 90.3%

Recall: 90.5%

F1 Score: 90.4%



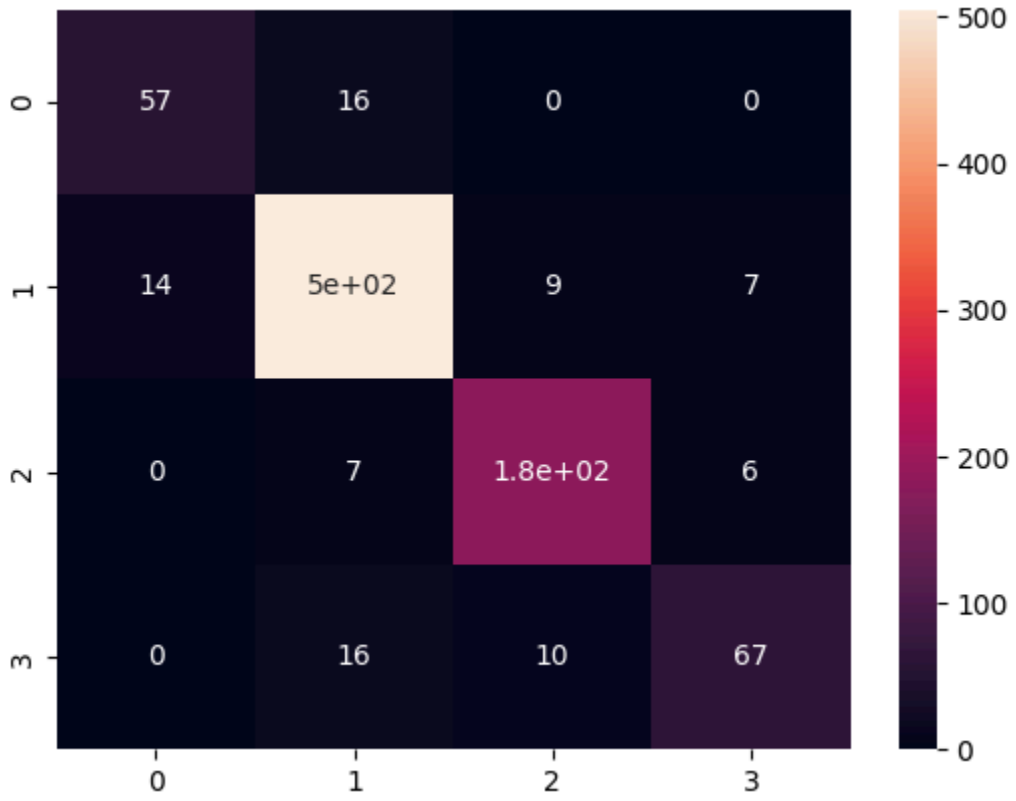
Decision Tree Classifier:

Accuracy: 100%

Precision: 100%

Recall: 100%

F1 Score: 100%



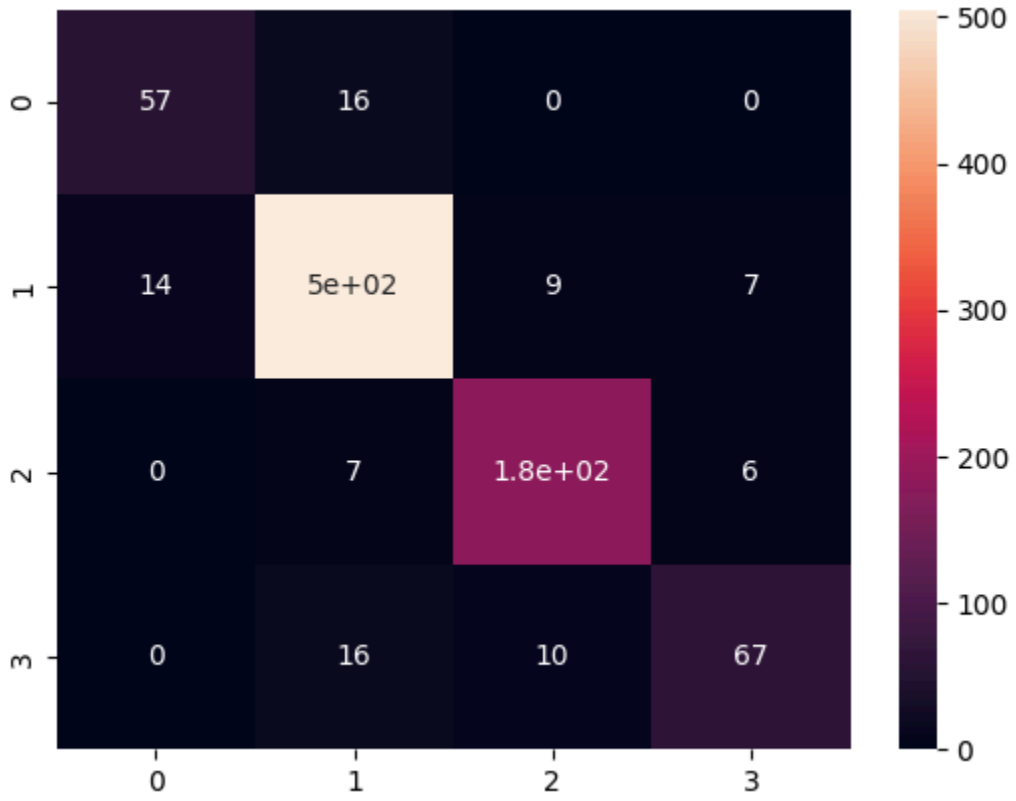
Random Forest Classifier:

Accuracy: 99%

Precision:99%

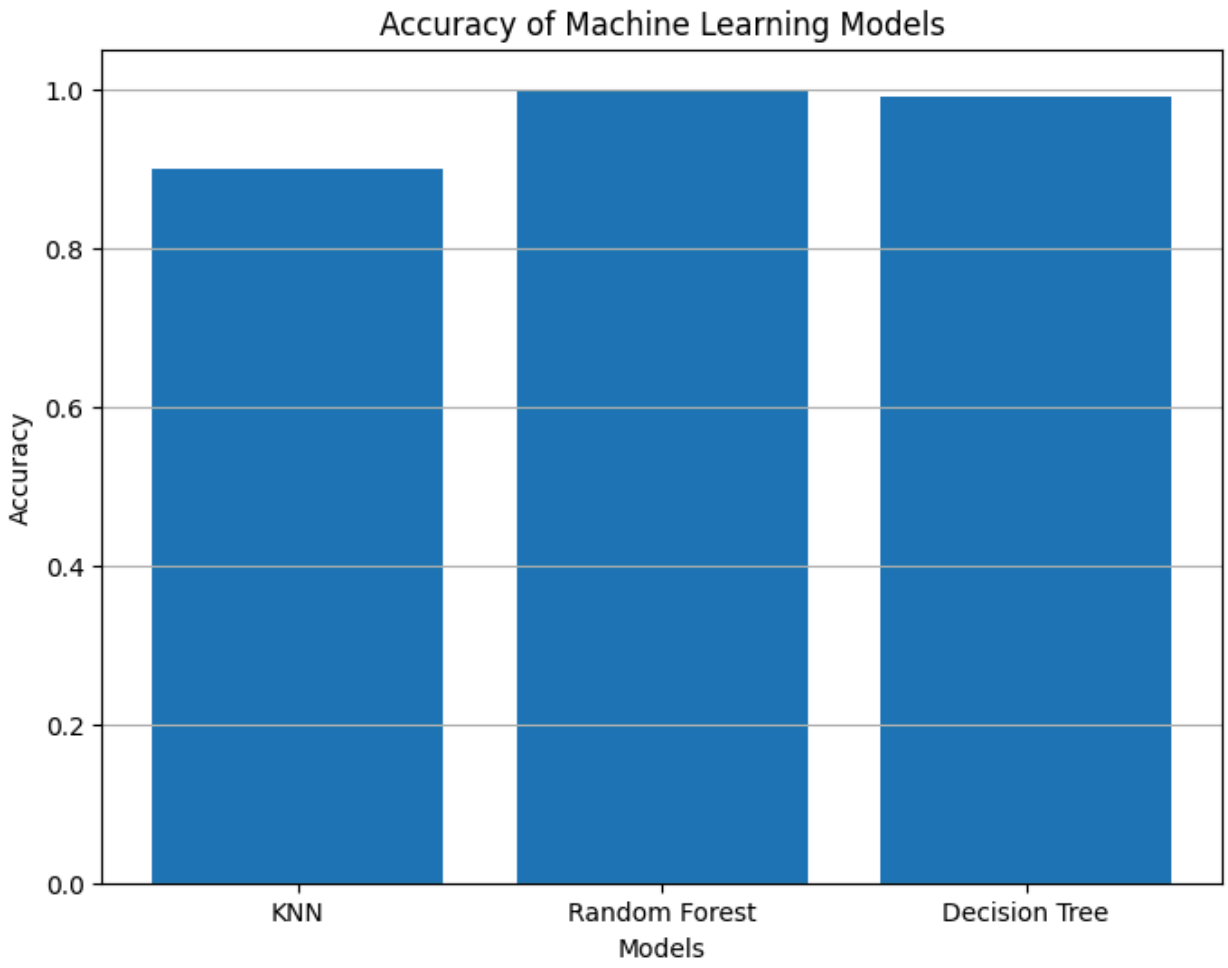
Recall: 99%

F1 Score:99%



Bar chart showcasing prediction accuracy of all models:

The barchart is showing the accuracy comparison for 3 different models.



Precision, recall comparison: Precision measures the proportion of true positive predictions among all positive predictions made by the model. A higher precision indicates fewer false positives. - Decision Tree Classifier has the highest precision. Decision Tree Classifier (100%), followed by Random Forest Classifier(99%), K-Nearest Neighbors (90%). This suggests that

Decision Tree Classifier is better at minimizing false positive predictions compared to the other models.

8. Conclusion

Decision Tree Classifier outperforms other models with the highest accuracy, precision, recall, and F1 score. Random Forest Classifier Shows moderate performance and K-Nearest Neighbors performs slightly lower than the others.