**Assessment Report**

on

# “Predict Traffic Congestion”

submitted as partial fulfillment for the award of

# BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

## CSE(AIML)

By

Name : Ananya Bharti

Roll Number : 20240110040003

Section: A

**Under the supervision of**

“BIKKI KUMAR SIR”

## KIET Group of Institutions, Ghaziabad

# May, 2025

---

1. **Introduction**

   Traffic congestion is a major urban challenge, causing delays, increased pollution, and inefficiency in transportation systems. Predicting and classifying traffic congestion levels can help cities manage traffic flow, optimize infrastructure, and reduce the negative impact on commuters. By utilizing real-time traffic sensor data, such as vehicle count, average speed, and environmental conditions, machine learning models can classify road sections into congestion levels—High, Medium, or Low. This can aid in making data-driven decisions for better traffic management and planning.

---

**2. Problem Statement**

The problem is to predict and classify the congestion levels of road sections into three categories—High, Medium, or Low—using traffic sensor data. The goal is to develop a machine learning model that analyzes various traffic-related features (such as vehicle count, speed, and environmental factors) to provide real-time insights for better traffic management and decision-making.

---

**3. Objectives**

- The objective is to build a machine learning model that accurately classifies the congestion level of road sections (High, Medium, or Low) based on real-time traffic sensor data. This model aims to assist traffic management systems in optimizing traffic flow, improving commuter experience, and reducing urban congestion.

---

4. **Methodology:**

- **Data Collection**: Gather traffic sensor data, including features like vehicle count, average speed, time of day, weather conditions, and historical traffic patterns.
- **Data Preprocessing**:
- Handle missing values, outliers, and inconsistencies in the dataset.
- Convert categorical variables (e.g., weather, time of day) into numerical values using techniques like one-hot encoding.
- Scale numerical features to ensure consistency across the dataset.
- **Model Selection**: Choose an appropriate machine learning algorithm (e.g., Decision Tree, Random Forest, SVM) to classify congestion levels based on the input features.
- **Model Training**: Split the data into training and testing sets and train the selected model on the training data.
- Evaluate the model using metrics like accuracy, precision, recall, and F1-score to assess its performance in classifying congestion levels.
- **Prediction**: Apply the trained model to new or real-time data to predict and classify congestion levels, aiding in traffic management strategies.
-

---

## 9. Conclusion

In conclusion, predicting and classifying traffic congestion levels using machine learning techniques offers significant potential for improving urban traffic management. By leveraging real-time sensor data, we can accurately classify road sections into High, Medium, or Low congestion categories, which can help cities optimize traffic flow, reduce delays, and minimize environmental impacts. The proposed model can serve as a valuable tool for traffic authorities, enabling data-driven decision-making and enhancing

commuter experiences. As traffic data becomes increasingly available, further improvements .in model accuracy and scalability can lead to smarter, more efficient urban transportation systems.

---

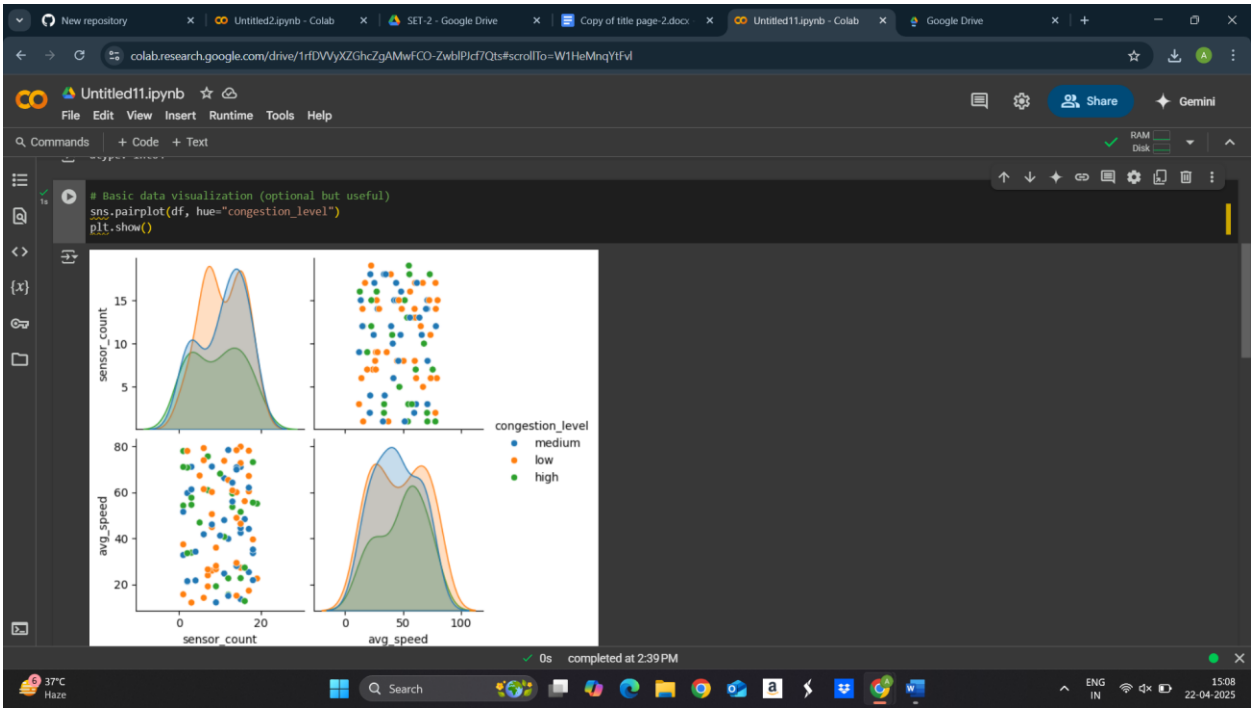
---

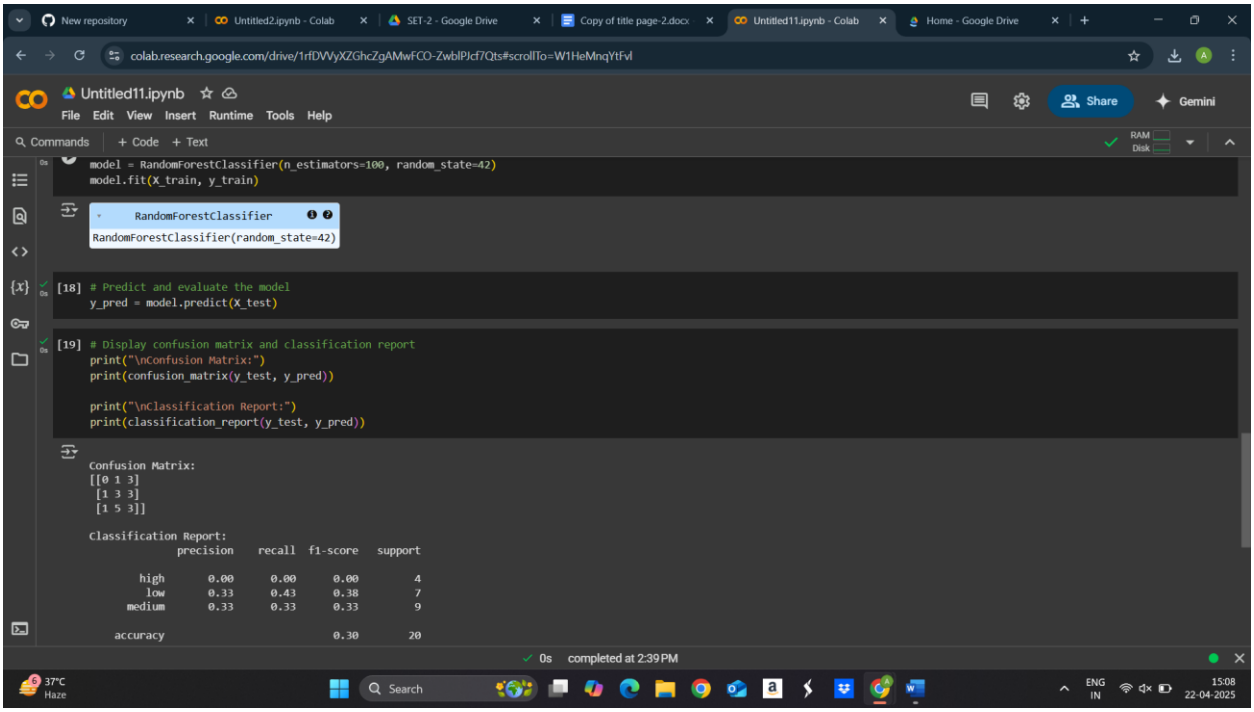
### 10. References

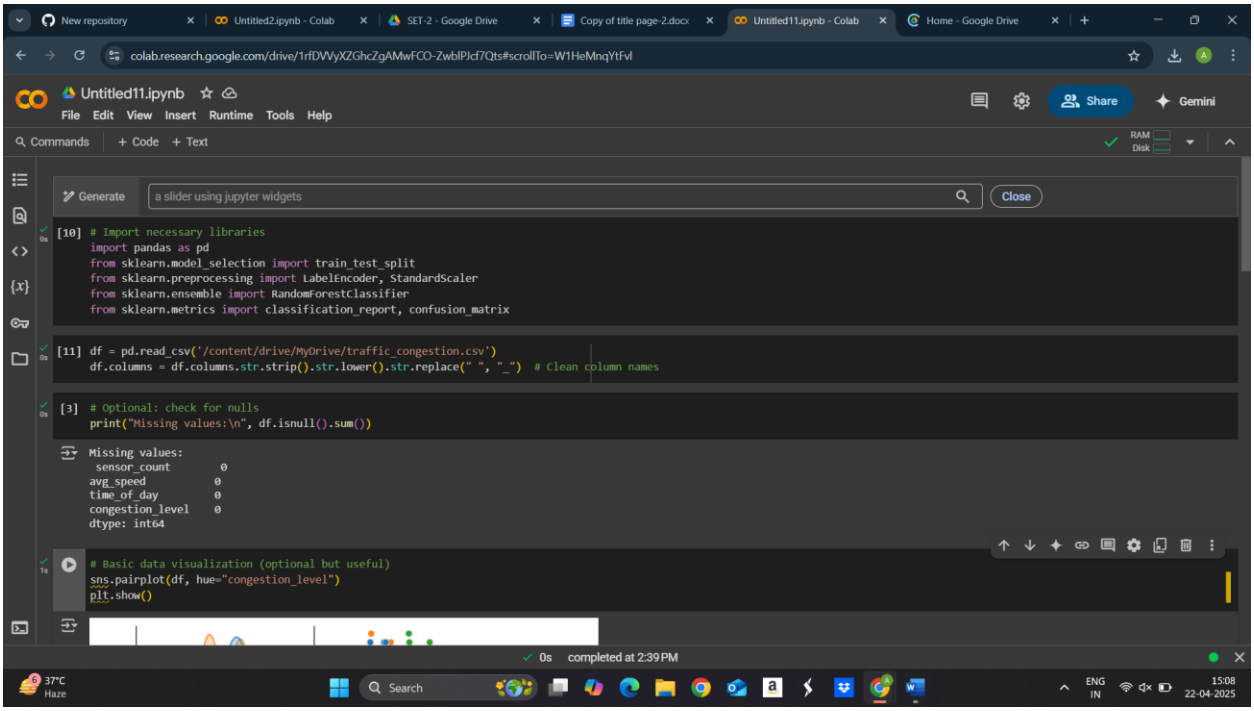

- Breiman, L. (1986). *Classification and Regression Trees*. Han, J., et al. (2011). *Data*
- Iglewicz, B., & Hoaglin, D.C. (1993). *How to Detect and Handle Outliers*.
- Sokolova, M., & Lapalme, G. (2009). *Performance Measures for Classification Tasks*.
- Zhang, Y., et al. (2010). *Urban Computing and Traffic Prediction*.


---

# CODE AND OUTPUT:




New repository
Untitled2.ipynb - Colab
SET-2 - Google Drive
Copy of title page-2.docx
Untitled11.ipynb - Colab
Home - Google Drive
colab.research.google.com/drive/1rfDVVyXZGhcZgAMwFCO-ZwblPJcf7Qts#scrollTo=W1HeMnqYtFvl
Untitled11.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text
RAM
Disk
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
RandomForestClassifier
RandomForestClassifier(random_state=42)
[18] # Predict and evaluate the model
y_pred = model.predict(X_test)
[19] # Display confusion matrix and classification report
print("\nConfusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
Confusion Matrix:
[[0 1 3]
[1 3 3]
[1 5 3]]
Classification Report:
precision recall f1-score support
high 0.00 0.00 0.00 4
low 0.33 0.43 0.38 7
medium 0.33 0.33 0.33 9
accuracy 0.30 20
0s completed at 2:39 PM
37°C
Haze
Search
ENG
IN
15:08
22-04-2025


New repository
Untitled2.ipynb - Colab
SET-2 - Google Drive
Copy of title page-2.docx
Untitled11.ipynb - Colab
Home - Google Drive
colab.research.google.com/drive/1rfDVVyXZGhcZgAMwFCO-ZwblPJcf7Qts#scrollTo=W1HeMnqYtFvl
Untitled11.ipynb
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands + Code + Text
RAM
Disk
Generate
a slider using jupyter widgets
Close
[10] # Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
[11] df = pd.read_csv('/content/drive/MyDrive/traffic_congestion.csv')
df.columns = df.columns.str.strip().str.lower().str.replace(" ", "_") # Clean column names
[3] # Optional: check for nulls
print("Missing values:\n", df.isnull().sum())
Missing values:
sensor_count 0
avg_speed 0
time_of_day 0
congestion_level 0
dtype: int64
# Basic data visualization (optional but useful)
sns.pairplot(df, hue="congestion_level")
plt.show()
0s completed at 2:39 PM
37°C
Haze
Search
ENG
IN
15:08
22-04-2025