www.kiet.edu
Delhi-NCR, Ghaziabad

KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

Assessment Report

on

“Traffic Volume Prediction”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AIML)

GROUP NO. : 9

- **Satyam Kumar(202401100400167)**
- **Shavyam chitranshi(202401100400173)**
- **Sharad Krishna Singh(202401100400170)**
- **Yash Jangid(202401100400216)**
- **Yashi Kesarwani(202401100400219)**

Introduction

In today’s fast-paced urban life, **traffic congestion** has evolved from being a minor inconvenience to a serious challenge that impacts everything — from **commute times** to **air quality**, **fuel consumption**, and even **emergency response**. With the rise of **smart cities** and **intelligent transportation systems**, there's an urgent need for **data-driven solutions** that can anticipate traffic patterns and aid in real-time traffic management.

This project focuses on leveraging **machine learning** to forecast **traffic volume** using key features such as **weather conditions** and **temporal variables** (like hour, day, and month). By analyzing historical traffic data and integrating advanced feature engineering techniques, the goal is to build a robust regression model that not only predicts traffic volume but also uncovers **insightful trends** behind urban mobility. Such predictions can empower city planners and commuters with foresight, enabling smarter decisions and more efficient transportation networks.

Methodology

Dataset Used

- **Dataset**: Metro Interstate Traffic Volume Dataset
- **Source**: Kaggle
- **Features**:
  - Time: date_time, hour, day_of_week, month, year
  - Weather: temp, rain_1h, snow_1h, clouds_all, Weather_main
    - Output Variable: traffic_volume

Workflow

- **Data Loading & Cleaning**

- **Exploratory Data Analysis (EDA)**

- **Feature Engineering** (cyclical time features, dummy variables, interaction terms)

- **Model Building** using Linear Regression, Random Forest, and Gradient Boosting
- **Evaluation** using MAE, RMSE, and R^2
- **Result Visualization**

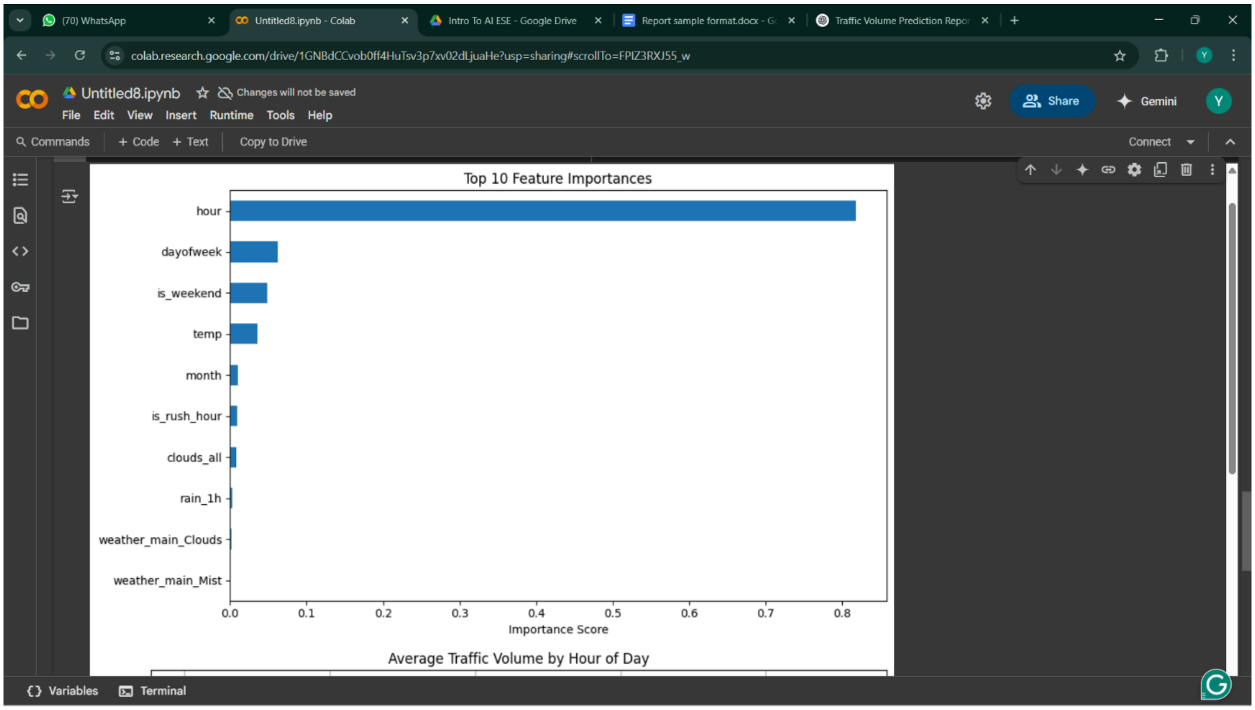
CODE

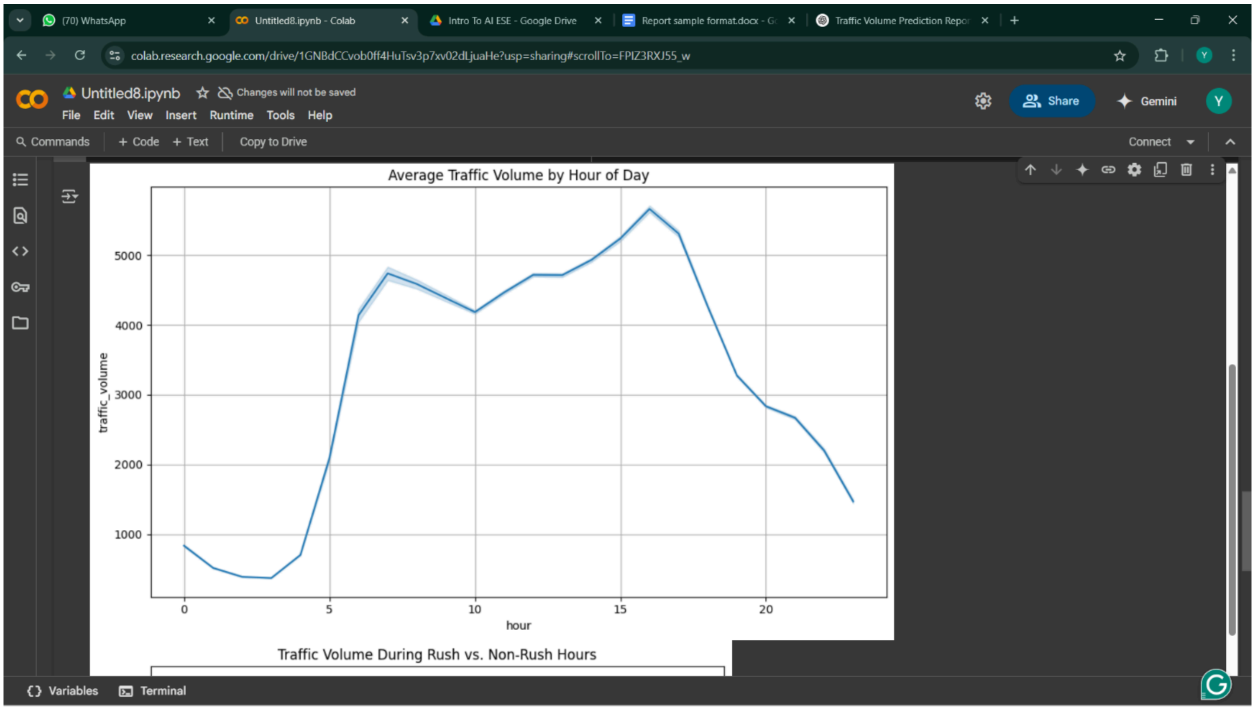
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
import kagglehub
path = kagglehub.dataset_download("rgupta12/metro-interstate-traffic-volume")
print("Path to dataset files:", path)
df = pd.read_csv("/root/.cache/kagglehub/datasets/rgupta12/metro-interstate-traffic-volume/versions/1/Metro_Interstate_Traffic_Volume.csv")
df['date_time'] = pd.to_datetime(df['date_time'])

# Time features
df['hour'] = df['date_time'].dt.hour
df['dayofweek'] = df['date_time'].dt.dayofweek
df['month'] = df['date_time'].dt.month
df['is_weekend'] = df['dayofweek'].isin([5, 6]).astype(int)
df['is_rush_hour'] = df['hour'].isin([7, 8, 16, 17, 18]).astype(int)

# Drop unnecessary columns
```

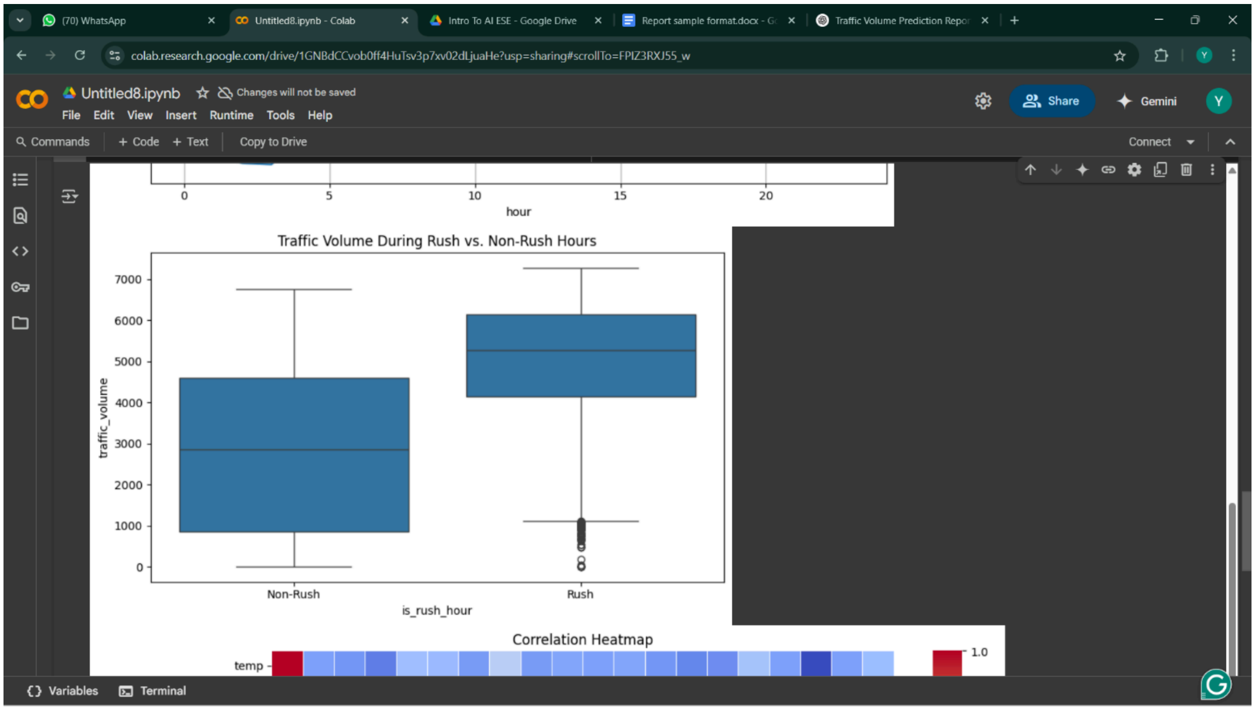
```
df.drop(columns=['date_time', 'holiday', 'weather_description'], inplace=True)

# One-hot encode categorical weather_main
df = pd.get_dummies(df, columns=['weather_main'], drop_first=True)

# Normalize continuous weather features
scaler = StandardScaler()
weather_cols = ['temp', 'rain_1h', 'snow_1h', 'clouds_all']
df[weather_cols] = scaler.fit_transform(df[weather_cols])

# ------------------------------
# Splitting Data
# ------------------------------

X = df.drop('traffic_volume', axis=1)
y = df['traffic_volume']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# ------------------------------
# Train Model
# ------------------------------

model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

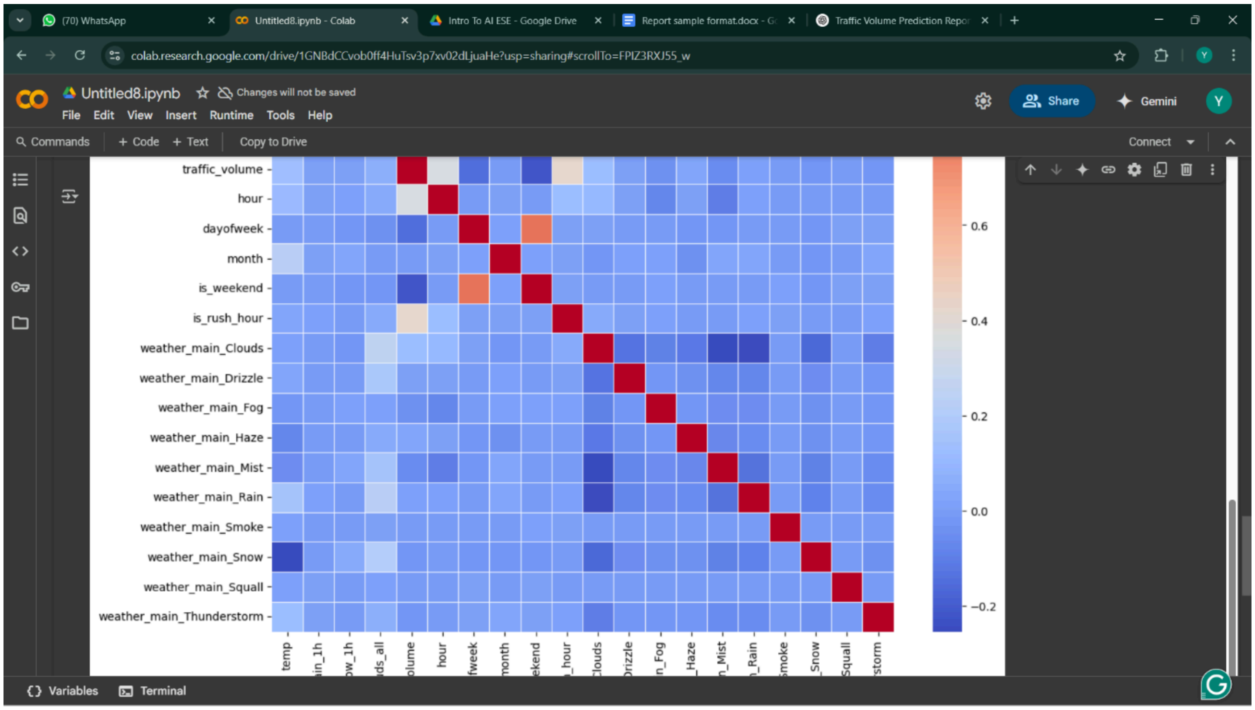
```
# ------------------------------
# Predictions
# ------------------------------

y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)

# ------------------------------
# Evaluation
# ------------------------------

train_mse = mean_squared_error(y_train, y_train_pred)
test_mse = mean_squared_error(y_test, y_test_pred)
train_r2 = r2_score(y_train, y_train_pred)
test_r2 = r2_score(y_test, y_test_pred)

print("✅ MODEL EVALUATION")
print(f"Training MSE      : {train_mse:.2f}")
print(f"Training R² Score : {train_r2:.4f}")
print(f"Testing MSE       : {test_mse:.2f}")
print(f"Testing R² Score  : {test_r2:.4f}")

# ------------------------------
# Feature Importance
# ------------------------------
```

```
importances = pd.Series(model.feature_importances_, index=X.columns)
plt.figure(figsize=(10, 6))
importances.sort_values().tail(10).plot(kind='barh')
plt.title("Top 10 Feature Importances")
plt.xlabel("Importance Score")
plt.tight_layout()
plt.show()

# ------------------------------
# Visualizations
# ------------------------------

# Traffic Volume by Hour
plt.figure(figsize=(10, 6))
sns.lineplot(x='hour', y='traffic_volume', data=df, estimator='mean')
plt.title("Average Traffic Volume by Hour of Day")
plt.grid(True)
plt.tight_layout()
plt.show()

# Rush Hour vs Non-Rush Hour
plt.figure(figsize=(8, 5))
sns.boxplot(x='is_rush_hour', y='traffic_volume', data=df)
plt.title("Traffic Volume During Rush vs. Non-Rush Hours")
plt.xticks([0, 1], ['Non-Rush', 'Rush'])
plt.tight_layout()
plt.show()


# Correlation Heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(df.corr(), cmap='coolwarm', linewidths=0.5)
plt.title("Correlation Heatmap")
plt.tight_layout()
plt.show()


# ------------------------------
# Final Summary
# ------------------------------
print("Model: Random Forest Regressor (100 trees)
print("\n📋 FINAL MODEL SUMMARY")
print(f"Training Accuracy (R²): {train_r2:.4f}")
print(f"Testing Accuracy  (R²): {test_r2:.4f}")
print(f"Feature with Highest Importance: {importances.idxmax()} ({importances.max():.4f})")


if test_r2 < 0.85:
    print("Consider changind the hyperparameters or using a more powerful model like XGBoost or LightGBM.")
else:
    print("✅ Model performs well with good generalization!")
```


(70) WhatsApp
Untitled8.ipynb - Colab
Intro To AI ESE - Google Drive
Report sample format.docx - G
Traffic Volume Prediction Repor
colab.research.google.com/drive/1GNBdCCvob0ff4HuTsv3p7xv02dLjuaHe?usp=sharing#scrollTo=FPIZ3RXJ55_w
Untitled8.ipynb
Changes will not be saved
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
Copy to Drive
Connect
Top 10 Feature Importances
hour
dayofweek
is_weekend
temp
month
is_rush_hour
clouds_all
rain_1h
weather_main_Clouds
weather_main_Mist
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
Importance Score
Average Traffic Volume by Hour of Day
Variables
Terminal


(70) WhatsApp
Untitled8.ipynb - Colab
Intro To AI ESE - Google Drive
Report sample format.docx - G
Traffic Volume Prediction Repor
colab.research.google.com/drive/1GNBdCCvob0ff4HuTsv3p7xv02dLjuaHe?usp=sharing#scrollTo=FPIZ3RXJ55_w
Untitled8.ipynb
Changes will not be saved
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
Copy to Drive
Connect
Average Traffic Volume by Hour of Day
traffic_volume
5000
4000
3000
2000
1000
0
5
10
15
20
hour
Traffic Volume During Rush vs. Non-Rush Hours
Variables
Terminal


(70) WhatsApp
Untitled8.ipynb - Colab
Intro To AI ESE - Google Drive
Report sample format.docx - G
Traffic Volume Prediction Repor
colab.research.google.com/drive/1GNBdCCvob0ff4HuTsv3p7xv02dLjuaHe?usp=sharing#scrollTo=FPIZ3RXJ55_w
Untitled8.ipynb
Changes will not be saved
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
Copy to Drive
Connect
0
5
10
15
20
hour
Traffic Volume During Rush vs. Non-Rush Hours
traffic_volume
0
1000
2000
3000
4000
5000
6000
7000
Non-Rush
Rush
is_rush_hour
Correlation Heatmap
temp
1.0
Variables
Terminal


(70) WhatsApp
Untitled8.ipynb - Colab
Intro To AI ESE - Google Drive
Report sample format.docx - G
Traffic Volume Prediction Repor
colab.research.google.com/drive/1GNBdCCvob0ff4HuTsv3p7xv02dLjuaHe?usp=sharing#scrollTo=FPIZ3RXJ55_w
Untitled8.ipynb
Changes will not be saved
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code
+ Text
Copy to Drive
Connect
traffic_volume
hour
dayofweek
month
is_weekend
is_rush_hour
weather_main_Clouds
weather_main_Drizzle
weather_main_Fog
weather_main_Haze
weather_main_Mist
weather_main_Rain
weather_main_Smoke
weather_main_Snow
weather_main_Squall
weather_main_Thunderstorm
temp
ain_1h
ow_1h
ds_all
olume
hour
fweek
month
ekend
_hour
Clouds
Drizzle
n_Fog
_Haze
n_Mist
_Rain
moke
Snow
Squall
storm
0.6
0.4
0.2
0.0
−0.2
Variables
Terminal

References/Credits

- Dataset Source: [Traffic Dataset from Kaggle]
- Image: Wikimedia Commons - Traffic in Los Angeles
- Libraries: Pandas, Scikit-learn, Matplotlib, Seaborn
- Code inspired by public examples and tutorials from Medium and Towards Data Science