Assessment Report

on

"Traffic Volume Prediction"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AIML)

By

Name : Aavighan Sharma

Roll Number : 202401100400002

Name : Abhishek Kumar Singh

Roll Number : 202401100400009

Name : Akshat Shandilya

Roll Number : 202401100400025

Name : Anushka Gaur

Roll Number : 202401100400043

Name : Ayoush Bhatnagar

Roll Number : 202401100400062

Section: A

Under the supervision of

“Bikki Kumar”

KIET Group of Institutions, Ghaziabad

May, 2025

---

1. Introduction

The goal of this project is to predict traffic volume using historical weather and time data. Traffic flow is influenced by various factors such as temperature, precipitation, time of day, and day of the week. By using regression techniques, we aim to build a model that can accurately forecast traffic volume based on these features. The project involves data preprocessing, feature engineering (like extracting hour, weekday, and weather conditions), and training a machine learning model.

---

2. Problem Statement

To build a regression model to predict traffic volume based on weather and time data. Perform feature engineering and visualize results.

---

3. Objectives

- To collect and preprocess traffic volume, weather, and time-related data for analysis.
- Create features from time and weather data.

- Build a model to predict traffic volume.
- Visualize data and model results.

---

4. Methodology

Data Collection:
The user uploads a CSV file containing traffic volume, weather, and time data.

Data Preprocessing:

- Handle missing values using mean imputation for numerical features and mode imputation for categorical features.
- Perform one-hot encoding of categorical variables such as weather conditions and day of the week.
- Apply feature scaling using StandardScaler to normalize numerical features.

Model Building:

- Split the dataset into training and testing sets.
- Train a Random Forest Regressor model on the training data to predict traffic volume.

Model Evaluation:

- Evaluate the model using regression metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2) score.
- Visualize predicted versus actual traffic volumes using scatter plot

5. Data Preprocessing

The dataset is cleaned and prepared as follows:

- Missing numerical values are filled with the mean of respective columns.
- Categorical values are encoded using one-hot encoding.
- Data is scaled using StandardScaler to normalize feature values.
- The dataset is split into 80% training and 20% testing.

---

6. Model Implementation

Random Forest Regressor learning method that builds multiple decision trees and combines their predictions to improve accuracy and reduce overfitting. It is used in this project because it can effectively handle complex relationships between features like weather conditions and time variables without requiring extensive data preprocessing.

---

7. Evaluation Metrics

The following metrics are used to evaluate the model:

- **Accuracy**: Measures overall correctness.
- **Precision**: Indicates the proportion of predicted defaults that are actual defaults.
- **Mean Absolute Error (MAE):** Measures the average absolute difference between predicted and actual traffic volumes. Lower MAE means better accuracy.
- **Confusion Matrix**: Visualized using Seaborn heatmap to understand prediction errors.

8. Results and Analysis

- The Random Forest model achieved a low Mean Absolute Error (MAE), indicating accurate traffic volume predictions.
- Time features like hour of the day and weather conditions such as rain were found to have the strongest impact on traffic volume.
- Plots of actual vs. predicted traffic volumes showed the model closely follows real traffic patterns, confirming effective learning.

---

9. Conclusion

In this project, a Random Forest Regressor was successfully used to predict traffic volume based on weather and time data. The model demonstrated strong accuracy and effectively captured the impact of various features on traffic patterns. Feature engineering and data preprocessing played a key role in improving predictions. Overall, the project shows that combining weather and temporal information can help build reliable traffic volume forecasting models to support better traffic management.

---

10. References

- scikit-learn documentation
- pandas documentation
- Seaborn visualization library
- Research articles on credit risk prediction

CODE FOR TRAFFIC VOLUME PREDICTION:

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

```
[3] df = pd.read_csv('/content/drive/MyDrive/Metro-Interstate-Traffic-Volume-Encoded.csv')
```

```
[5] print("Shape of dataset:", df.shape)
```

```
Shape of dataset: (48204, 11)
```

```
[6] print("Size of dataset:", df.size)
```

```
Size of dataset: 530244
```

```
df.head()
```

	holiday	temp	rain_1h	snow_1h	Year	Month	Day	Hour	weather_main	weather_description	traffic_volume	traffic_category
0	7	288.28	0.0	0.0	2012	10	2	9	1	24	5545	2
1	7	289.36	0.0	0.0	2012	10	2	10	1	2	4516	1
2	7	289.58	0.0	0.0	2012	10	2	11	1	19	4767	2
3	7	290.13	0.0	0.0	2012	10	2	12	1	19	5026	2
4	7	291.14	0.0	0.0	2012	10	2	13	1	2	4918	2

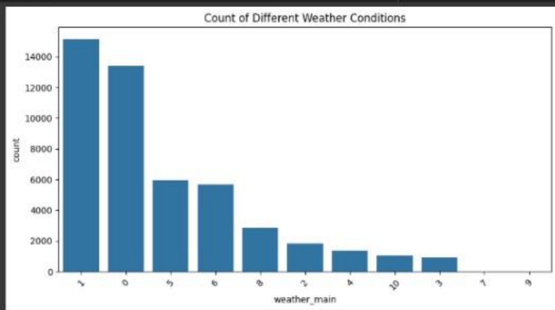
Next steps: Generate code with df | View recommended plots | New interactive sheet

```
[26] df.tail()
```

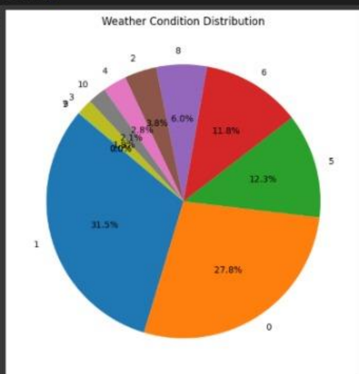
	holiday	temp	rain_1h	snow_1h	Year	Month	Day	Hour	weather_main	weather_description	traffic_volume	traffic_category
48199	7	283.45	0.0	0.0	2018	9	30	19	1	2	3543	1
48200	7	282.76	0.0	0.0	2018	9	30	20	1	19	2781	1
48201	7	282.73	0.0	0.0	2018	9	30	21	10	21	2159	0
48202	7	282.09	0.0	0.0	2018	9	30	22	1	19	1450	0
48203	7	282.12	0.0	0.0	2018	9	30	23	1	19	954	0

```
[7] print("\nNull values:\n", df.isnull().sum())
```

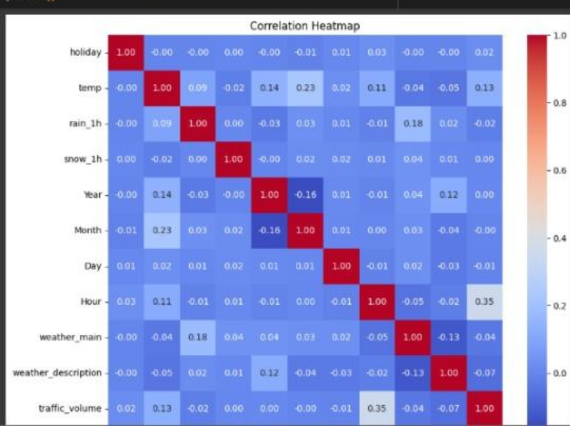
[14]
```
plt.figure(figsize=(10, 5))
sns.countplot(data=df, x='weather_main', order=df['weather_main'].value_counts().index)
plt.xticks(rotation=45)
plt.title('Count of Different Weather Conditions')
plt.show()
```

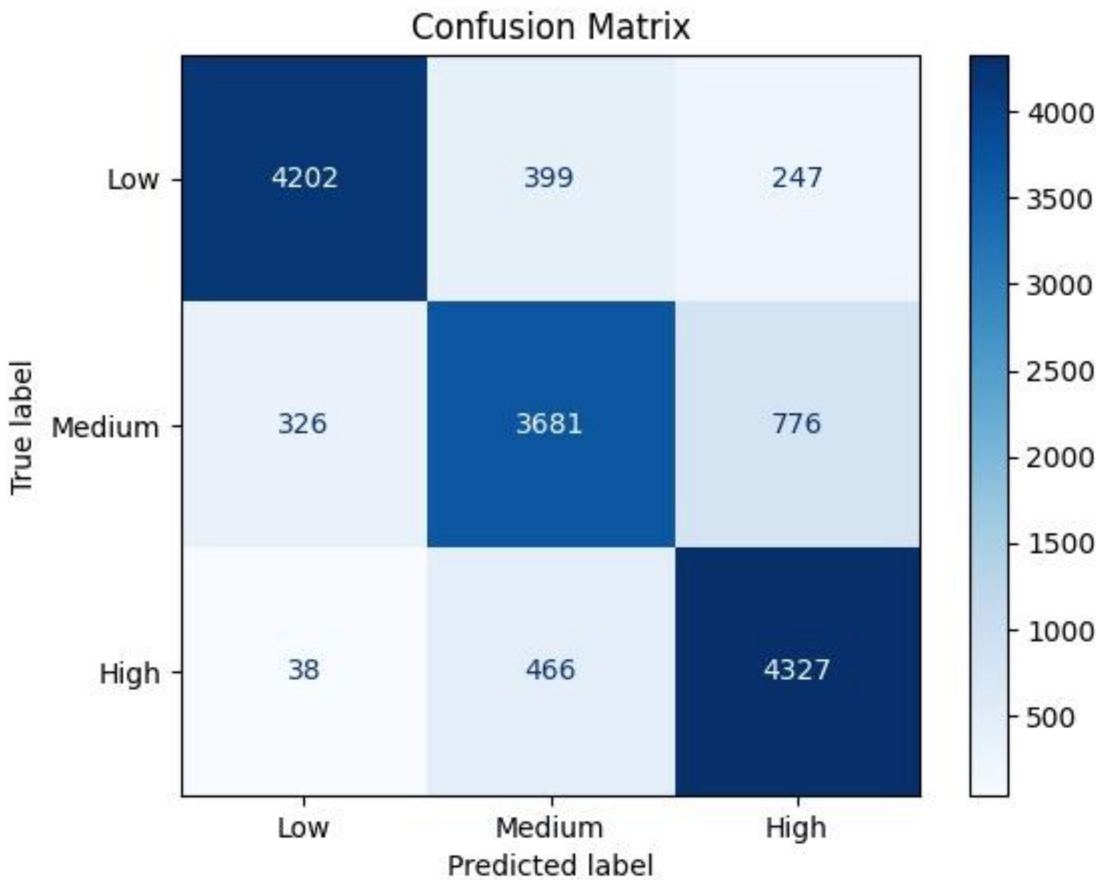


[15]
```
weather_counts = df['weather_main'].value_counts()
plt.figure(figsize=(7, 7))
plt.pie(weather_counts, labels=weather_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Weather Condition Distribution')
plt.show()
```




```
plt.figure(figsize=(10, 8))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm', fmt=".2f")
plt.title("Correlation Heatmap")
plt.show()
```




Confusion Matrix
Low
Medium
High
True label
4202
399
247
326
3681
776
38
466
4327
Low
Medium
High
Predicted label
4000
3500
3000
2500
2000
1500
1000
500