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 : Abhay Pratap Singh

Roll Number : 202401100400005

Section: A

Under the supervision of

“BIKKI KUMAR SIR”

KIET Group of Institutions, Ghaziabad

May, 2025

1. Introduction

With the rapid urbanization and rise in vehicle usage, traffic congestion has become a significant concern in metropolitan cities. Predicting traffic congestion using data-driven machine learning methods can help city planners, traffic authorities, and commuters to take timely decisions. This project aims to predict traffic congestion levels using supervised learning techniques based on features like time of day, day of the week, weather conditions, and historical traffic volume.

2. Problem Statement

To develop a machine learning model that predicts the level of traffic congestion based on real-time and historical traffic-related data. Accurate predictions can facilitate better traffic management and route planning to reduce delays and improve commuter experience.

3. Objectives

- Preprocess the dataset for training a machine learning model.
- Train a Logistic Regression model to classify traffic congestion levels.
- Evaluate the model performance using classification metrics.
- Visualize the confusion matrix to interpret the model's accuracy and error types.

4. Methodology

- **Data Collection**: The user uploads a CSV file containing the dataset.
- **Data Preprocessing**:
  - Handling missing values using mean/mode imputation.
  - Encoding categorical variables like day of the week and weather condition.

  - Scaling numerical features such as vehicle count and temperature.
- **Model Building**:
  - Dataset is split into training (80%) and testing (20%) sets.
  - Logistic Regression model is trained to classify congestion levels (e.g., low, medium, high).
- **Model Evaluation**:
- Evaluate using accuracy, precision, recall, and F1-score.
- Use a confusion matrix heatmap for visual interpretation of model results.

5. Data Preprocessing

The dataset is cleaned and prepared as follows:

- Missing numerical values (e.g., temperature, vehicle count) are imputed using mean.
- Categorical variables like weather and day are converted using one-hot encoding.
- StandardScaler is applied to normalize the features.
- Dataset split ensures the model is validated on unseen data.

6. Model Implementation

Logistic Regression is chosen for its interpretability and efficiency in classification problems. The model is trained using the preprocessed dataset to predict congestion levels during various conditions.

7. Evaluation Metrics

The following metrics are used to evaluate the model:

- **Accuracy:** Overall correctness of the model.
- **Precision:** Ratio of correctly predicted congestion events.
- **Recall:** Ability to identify actual congested periods.
- **F1 Score:** Balances precision and recall.
- **Confusion Matrix:** Visual representation using Seaborn to analyze prediction distribution.

8. Results and Analysis

- The model demonstrates reliable performance in predicting congestion levels.
- The confusion matrix highlights correct vs. incorrect predictions across congestion classes.
- Metrics reveal how well the model identifies high congestion scenarios without many false positives.

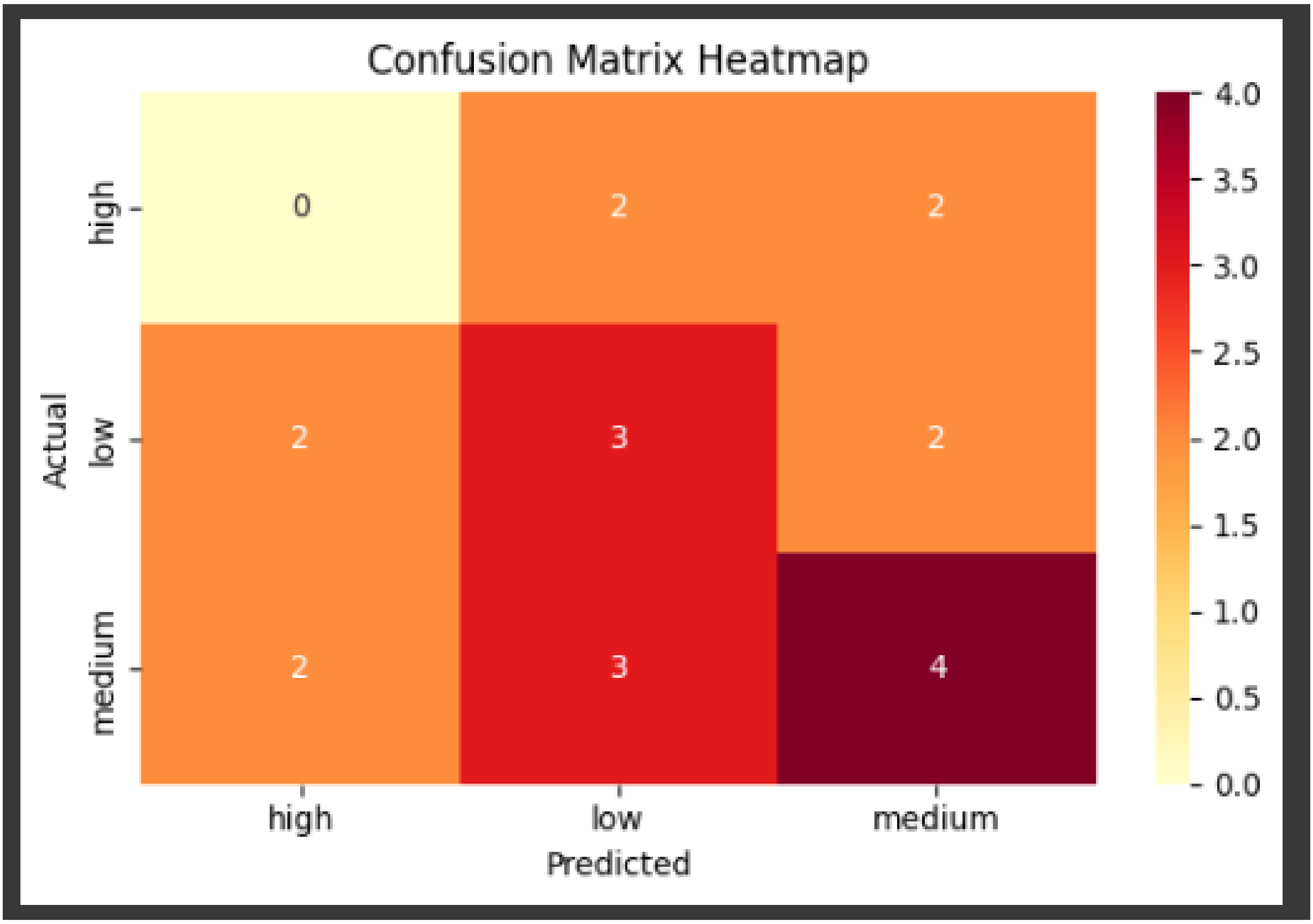
9. Conclusion

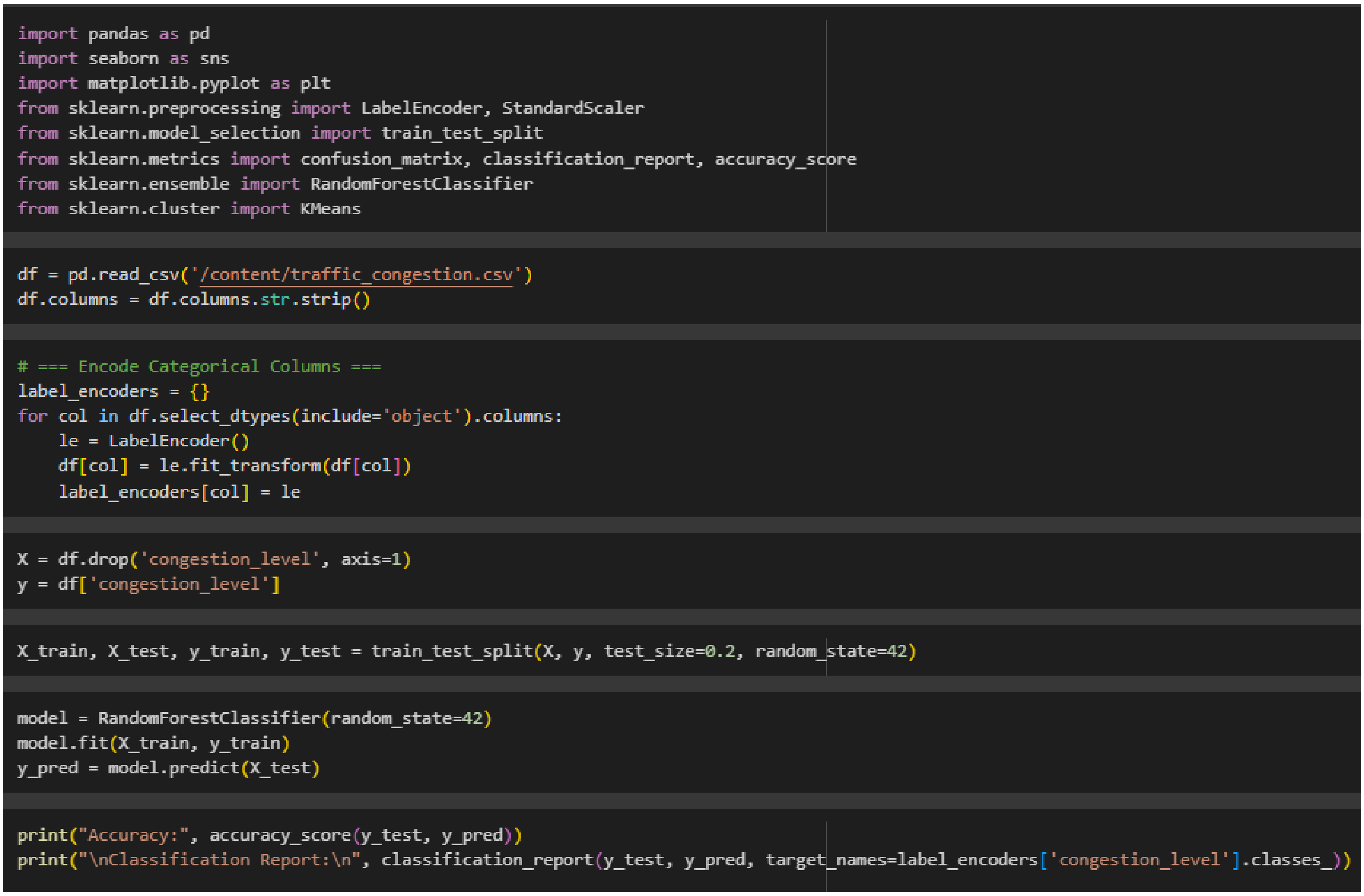
This project shows how machine learning can be applied to traffic data to forecast congestion. Logistic Regression proved effective in modeling congestion patterns, though performance can be improved with advanced models like Random Forest or XGBoost, and by incorporating more granular data (e.g., GPS, real-time feeds).

10. References

- scikit-learn documentation
- pandas documentation

- Seaborn visualization library
- Research articles on traffic flow and congestion prediction





```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.cluster import KMeans
```

```
df = pd.read_csv('/content/traffic_congestion.csv')
df.columns = df.columns.str.strip()
```

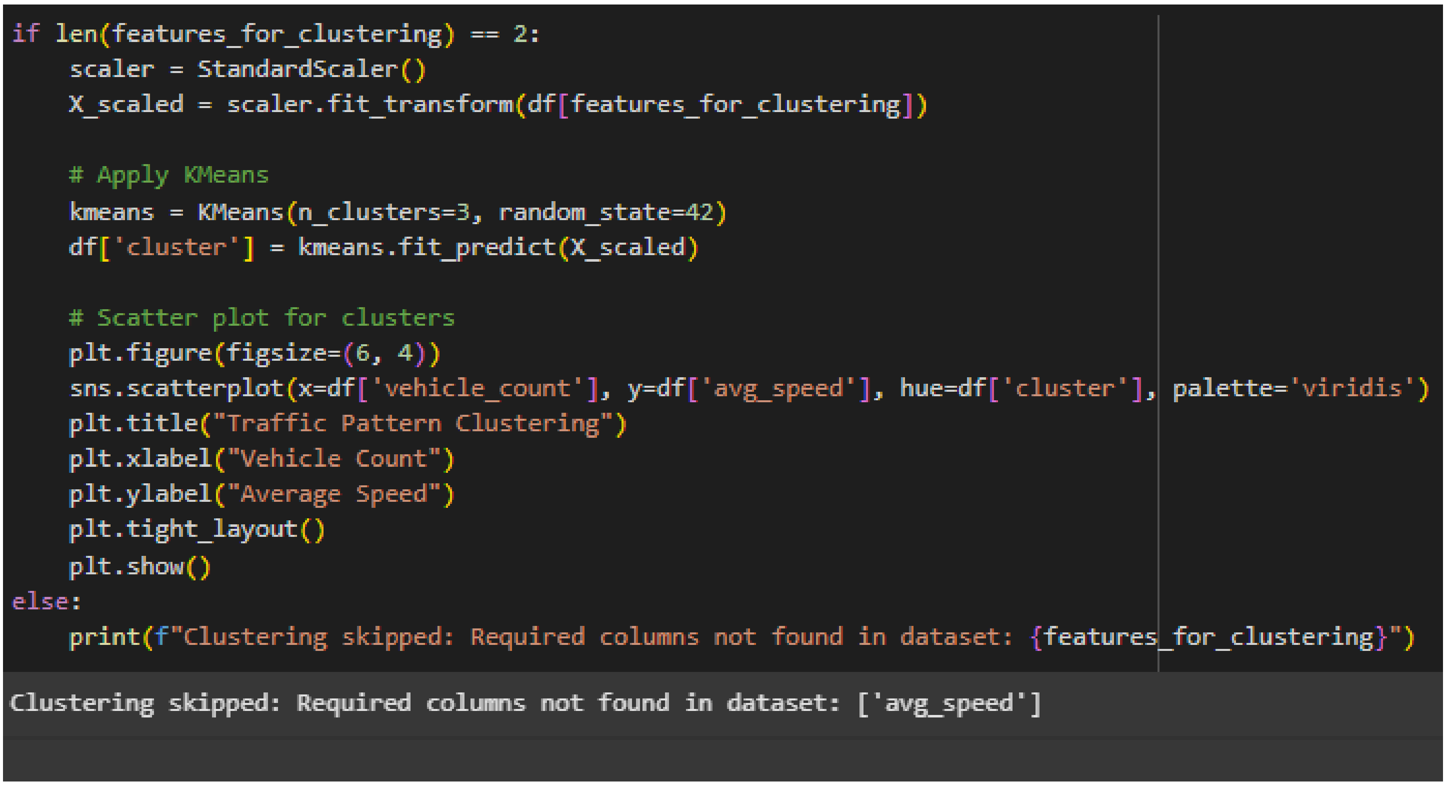
```
# === Encode Categorical Columns ===
label_encoders = {}
for col in df.select_dtypes(include='object').columns:
    le = LabelEncoder()
    df[col] = le.fit_transform(df[col])
    label_encoders[col] = le
```

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

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

```
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

```
print("Accuracy:", accuracy_score(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred, target_names=label_encoders['congestion_level'].classes_))
```



```
if len(features_for_clustering) == 2:
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(df[features_for_clustering])

    # Apply KMeans
    kmeans = KMeans(n_clusters=3, random_state=42)
    df['cluster'] = kmeans.fit_predict(X_scaled)

    # Scatter plot for clusters
    plt.figure(figsize=(6, 4))
    sns.scatterplot(x=df['vehicle_count'], y=df['avg_speed'], hue=df['cluster'], palette='viridis')
    plt.title("Traffic Pattern Clustering")
    plt.xlabel("Vehicle Count")
    plt.ylabel("Average Speed")
    plt.tight_layout()
    plt.show()
else:
    print(f"Clustering skipped: Required columns not found in dataset: {features_for_clustering}")
```

```
Clustering skipped: Required columns not found in dataset: ['avg_speed']
```