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 : Aryan Tomar

Roll Number : 202401100400055

Section: A

Under the supervision of

“Bikki Kumar”

KIET Group of Institutions, Ghaziabad

May, 2025

---

1. Introduction

Urban traffic congestion has become a significant issue in modern cities, resulting in lost productivity, increased pollution, and driver frustration. Predicting traffic congestion levels and understanding traffic behavior can help city planners and traffic management authorities make informed decisions. This project focuses on classifying road sections into different congestion levels using machine learning and performing segmentation to identify patterns in traffic behavior using clustering techniques.

---

2. Problem Statement

Urban traffic congestion has become a significant issue in modern cities, resulting in lost productivity, increased pollution, and driver frustration. Predicting traffic congestion levels and understanding traffic behavior can help city planners and traffic management authorities make informed decisions.

---

3. Objectives

- Preprocess the dataset for training a machine learning model.
- Train a Logistic Regression model to classify loan defaults.
- Evaluate model performance using standard classification metrics.
- Visualize the confusion matrix using a heatmap for interpretability.

---

4. Methodology

1. Classification Task

- Objective: Classify road sections into 'High', 'Medium', or 'Low' congestion levels.
- Approach:
  - Preprocess the dataset by encoding categorical variables.
  - Split the dataset into training and test sets.
  - Train a Random Forest Classifier.
  - Evaluate model performance using Accuracy, Precision, Recall, and Classification Report.
  - Visualize results using a Confusion Matrix, Pie Chart, and Feature Importance bar chart.

2. Clustering Task

- Objective: Segment road sections based on traffic patterns without using labels.
- Approach:
  - Remove the target variable to perform unsupervised learning.
  - Use KMeans clustering to group road segments into clusters.
  - Apply PCA to reduce dimensions for 2D visualization.
  - Visualize the clusters using a 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

Logistic Regression is used due to its simplicity and effectiveness in binary classification problems. The model is trained on the processed dataset and used to predict the traffic congestion level.

---

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.
- **Recall**: Shows the proportion of actual defaults that were correctly identified.
- **F1 Score**: Harmonic mean of precision and recall.
- **Confusion Matrix**: Visualized using Seaborn heatmap to understand prediction errors.

---

8. Results and Analysis

- The model provided reasonable performance on the test set.
- Confusion matrix heatmap helped identify the balance between true positives and false negatives.
- Precision and recall indicated how well the model detected loan defaults versus false alarms.

---

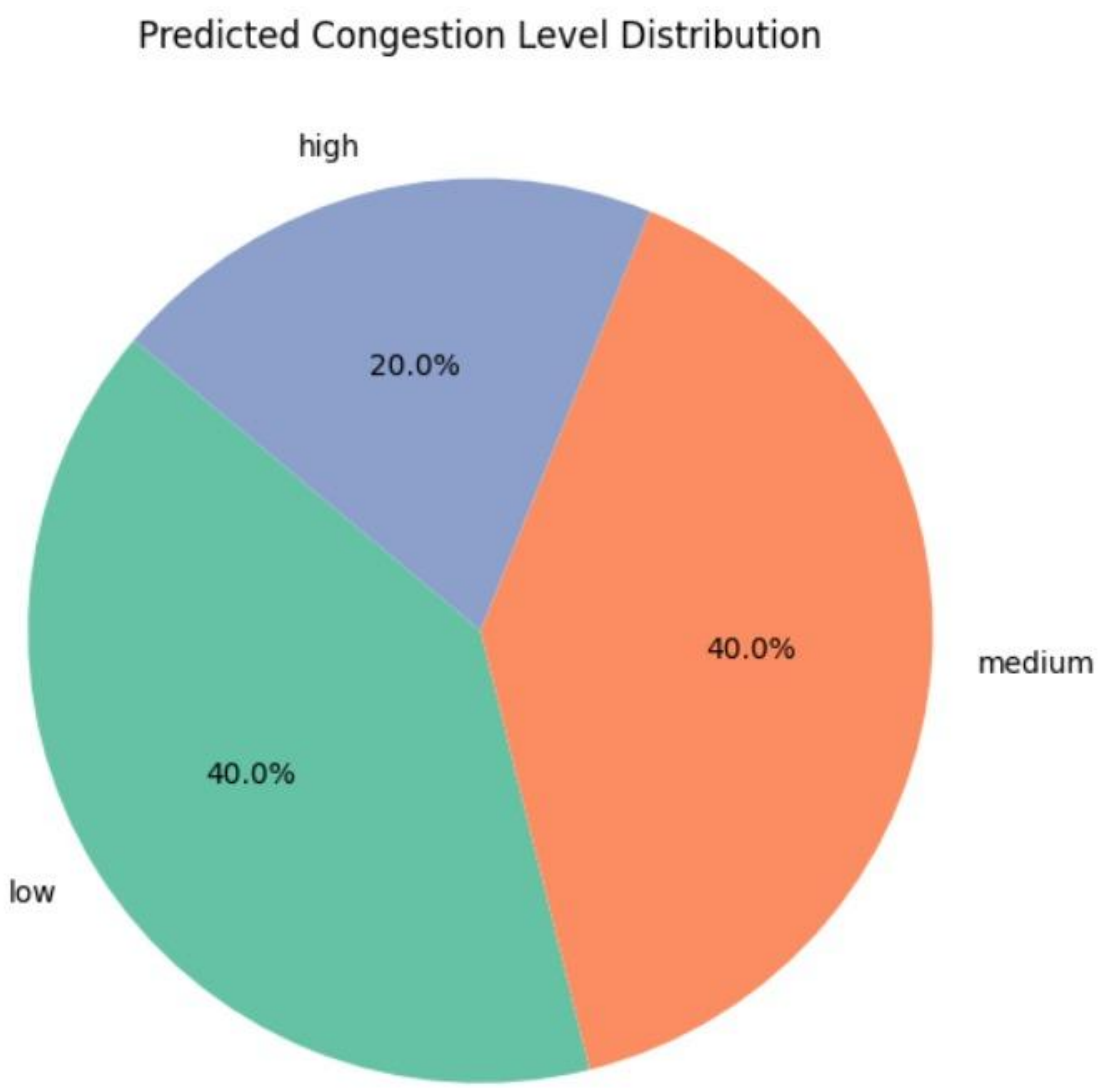
9. Conclusion

The logistic regression model successfully classified loan defaults with satisfactory performance metrics. The project demonstrates the potential of using machine learning for automating loan approval processes and improving risk assessment. However, improvements can be made by exploring more advanced models and handling imbalanced data.

Code

The following code performs both classification and clustering tasks:

```
# Import necessary libraries
import pandas as pd
```




```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score, precision_score,
recall_score
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA

# Load dataset
df = pd.read_csv("traffic_congestion.csv")

# Encode categorical features
```

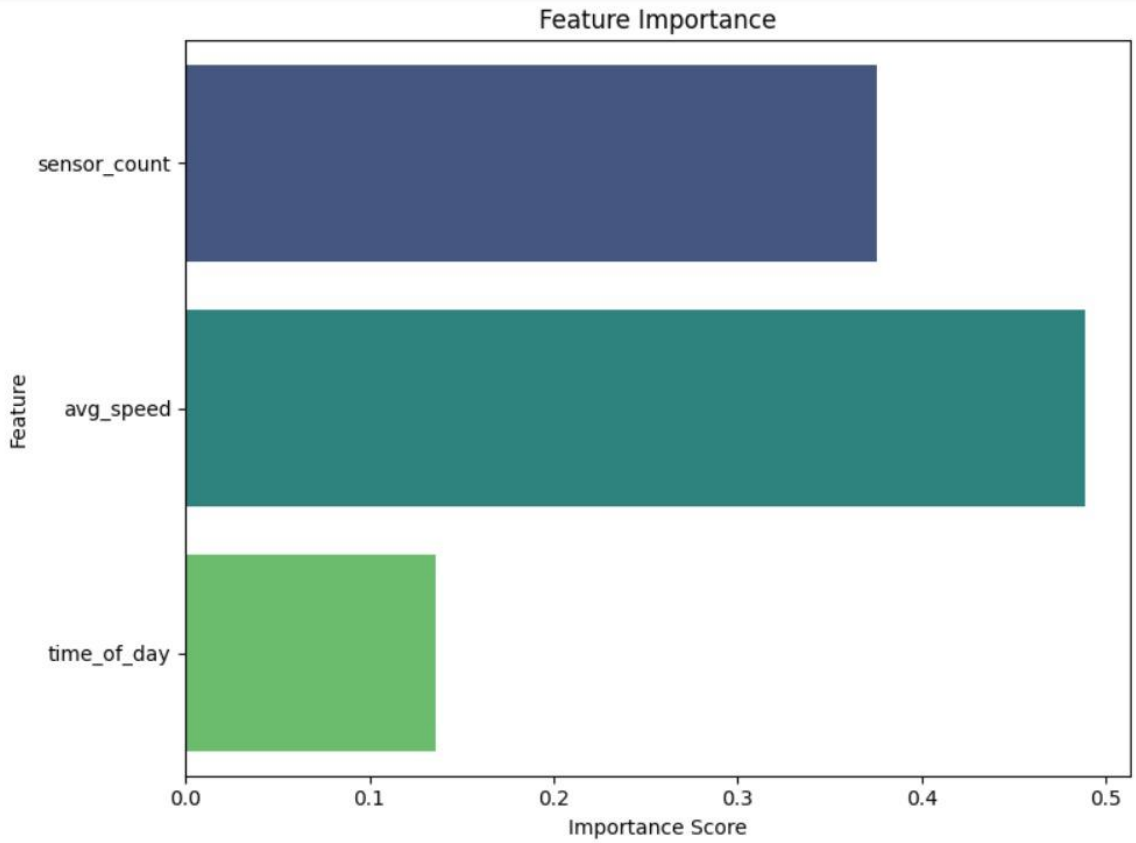
```
df_encoded = df.copy()
le_time = LabelEncoder()
df_encoded['time_of_day'] = le_time.fit_transform(df_encoded['time_of_day'])

le_target = LabelEncoder()
df_encoded['congestion_level'] = le_target.fit_transform(df_encoded['congestion_level'])

# Classification
X = df_encoded.drop(columns='congestion_level')
y = df_encoded['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)

# Evaluation
conf_matrix = confusion_matrix(y_test, y_pred)
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
report = classification_report(y_test, y_pred, target_names=le_target.classes_)

# Clustering
X_unsupervised = df_encoded.drop(columns='congestion_level')
kmeans = KMeans(n_clusters=3, random_state=42)
clusters = kmeans.fit_predict(X_unsupervised)

pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_unsupervised)
```

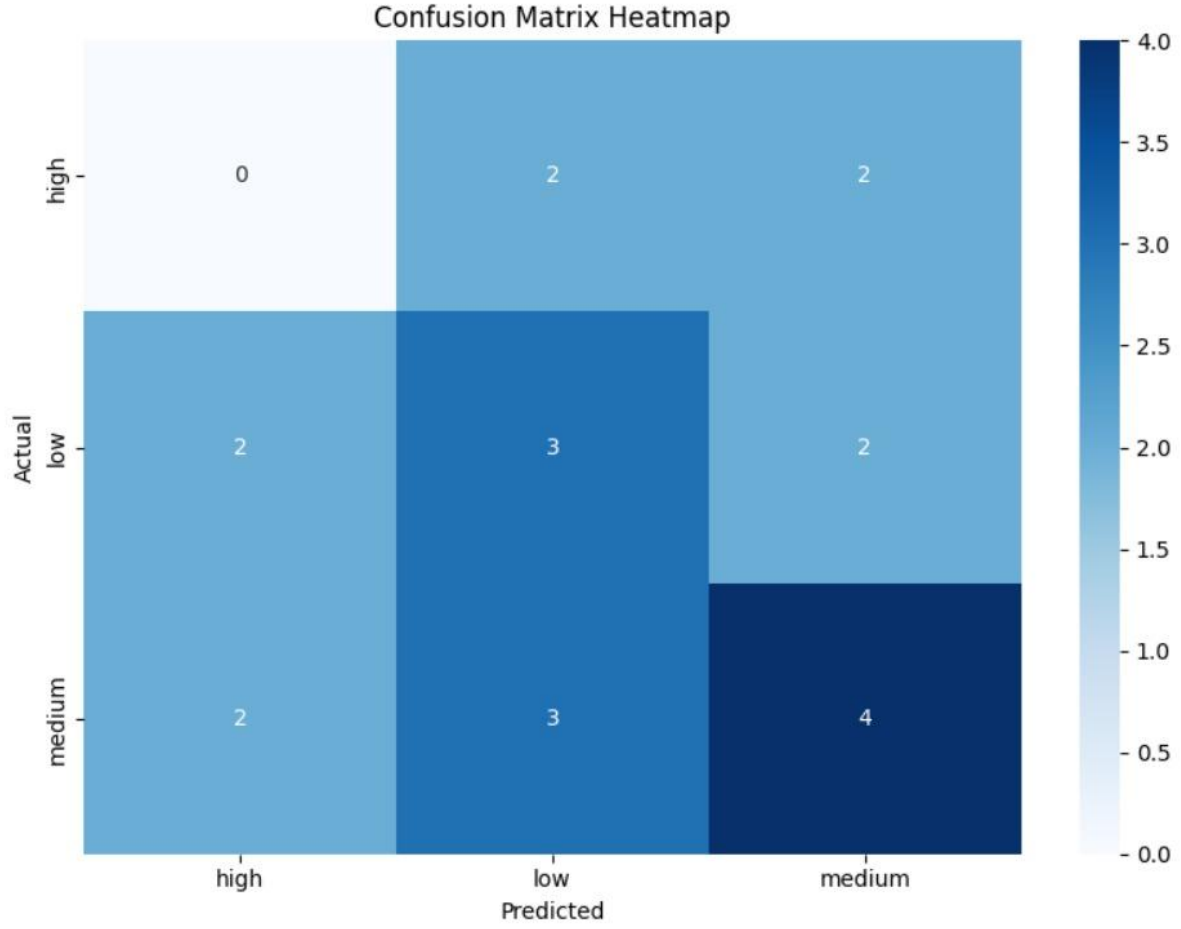
---

10. References

- scikit-learn documentation
- pandas documentation
- Seaborn visualization library

- Research articles on credit risk prediction


Feature Importance
sensor_count
avg_speed
time_of_day
Feature
0.0
0.1
0.2
0.3
0.4
0.5
Importance Score




```
Classification Report:
              precision    recall  f1-score   support

        high       0.00      0.00      0.00         4
         low       0.38      0.43      0.40         7
      medium       0.50      0.44      0.47         9

    accuracy                           0.35        20
   macro avg       0.29      0.29      0.29        20
weighted avg       0.36      0.35      0.35        20

Accuracy: 0.35
Precision: 0.36
Recall: 0.35
```