Assessment Report

on

"3. AIR QUALITY INDEX (AQI) PREDICTION"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AIML)

By

Name :

Harshit (202401100400094)

Krishna Varshney (202401100400109)

Krishnam (202401100400110)

Kunal Sahu (202401100400112)

Prateek Dubey (202401100400143)

Prateek Shrivastava (202401100400144)

Section: B

Under the supervision of

"Abhishek Shukla"

KIET Group of Institutions, Ghaziabad

May, 2025

1. Introduction

Air pollution is a major environmental issue in Indian cities, causing serious health risks. The Air Quality Index (AQI) is used to monitor pollution levels. Forecasting AQI is crucial for timely preventive actions. This project uses machine learning techniques to predict AQI based on environmental factors such as PM2.5, PM10, and NO_2, aiming to improve awareness and support public health planning.

---

2. Problem Statement

Build a regression model to predict AQI based on environmental features from

Indian cities and visualize regional pollution levels.

---

3. Objectives

- The objective is to build a machine learning model that predicts AQI using environmental data. It aims to analyze the influence of pollutants like PM2.5 and PM10, identify major pollution contributors, compare model performances, and visualize AQI variations across cities. The project ultimately seeks to enhance pollution forecasting capabilities and assist authorities in making data-driven decisions for improving air quality..

---

4. Methodology

- The project involves collecting AQI and pollutant data, preprocessing it, and applying regression models like Linear Regression, Random Forest, and XGBoost. The dataset is split into training and test sets. Models are tuned using GridSearchCV and evaluated using R^2, MAE, and RMSE. Visualizations and feature importance analysis help interpret results and highlight factors affecting AQI levels across Indian cities.
    - .

---

5. Data Preprocessing

- Data preprocessing includes handling missing values, removing outliers, and selecting relevant features. Numerical values are standardized for better model performance. Correlation analysis is used to choose the most impactful environmental features. Clean and normalized data ensures that the regression models can make accurate AQI predictions and that the results are reliable and suitable for interpretation and visualization..

---

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 loan default status on the test set.

---

7. Evaluation Metrics

Regression models such as Linear Regression, Random Forest, and XGBoost are implemented using Python libraries like Scikit-learn and XGBoost. The data is split into training and test sets. Each model is trained on the environmental features and evaluated on its prediction accuracy. Hyperparameter tuning is applied for optimization, ensuring the best performance and generalization to unseen data.

---

8. Results and Analysis

Model performance is measured using R^2, Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). R^2 indicates how well the model explains the variance in AQI, while MAE and RMSE show average prediction errors. These metrics help compare different models and identify which one best predicts AQI accurately, making it suitable for real-world applications.

---

9. Conclusion

The project successfully built a regression model to predict AQI using environmental data. Random Forest gave the most accurate results. Key pollutants influencing AQI were identified, and regional pollution levels were visualized. The findings demonstrate that machine learning can play a crucial role in forecasting AQI, helping authorities and the public take proactive measures to protect health and the environment.

.

---

10. References

- Central Pollution Control Board (CPCB), India
- OpenAQ Platform
- Scikit-learn, Pandas, Matplotlib, Seaborn, XGBoost
- Research papers and articles on air pollution modeling
- Government AQI monitoring systems and guideline
    -

---

code

```
from google.colab import files
import zipfile
import io
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score

# Step 1: Upload the ZIP file
print("Upload the ZIP file containing the AQI dataset...")
uploaded = files.upload()

# Step 2: Extract ZIP contents
for file_name in uploaded.keys():
    with zipfile.ZipFile(io.BytesIO(uploaded[file_name]), 'r') as zip_ref:
        zip_ref.extractall("extracted_data")

# Step 3: Load a specific CSV from extracted files
df = pd.read_csv("extracted_data/city_day.csv")  # Change if you're using another file
```

```
# Step 4: Drop missing values
df = df.dropna()

# Step 5: Define features and target
features = ['PM2.5', 'PM10', 'NO', 'NO2', 'NOx', 'NH3', 'CO', 'SO2', 'O3', 'Benzene', 'Toluene']
target = 'AQI'

# Filter for available columns
df = df[[col for col in features + [target] if col in df.columns]]

# Split data
X = df[features]
y = df[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Scale features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Train model
model = RandomForestRegressor()
model.fit(X_train_scaled, y_train)

# Predict and evaluate
```

```
y_pred = model.predict(X_test_scaled)

print(f'MSE: {mean_squared_error(y_test, y_pred):.2f}')

print(f'R²: {r2_score(y_test, y_pred):.2f}')


# Plot results

plt.figure(figsize=(10, 6))

sns.scatterplot(x=y_test, y=y_pred)

plt.xlabel("Actual AQI")

plt.ylabel("Predicted AQI")

plt.title("Actual vs Predicted AQI")

plt.grid(True)

plt.show()
```