Malaviya National Institute of Technology Jaipur

Department of Chemical Engineering

Subject- AIML in Chemical Engineering

Topic – Prediction of Air Quality Index using Supervised Learning

Submitted To:

Prof. Kailash Singh

Submitted By:

Vaibhav Jain(2023UCH1193)

Vedant Sharma(2023CUCH1265)

Aman Sharma(2023UCH1415)

Devraj Sharma(2023UCH1438)

2. Abstract

Air pollution has become a serious concern in many cities, and having reliable ways to predict air quality is important so that action can be taken before conditions get worse. In this project, I explored how a supervised machine learning model—specifically a Random Forest Regressor—can be used to predict the Air Quality Index (AQI) using real pollutant data. The dataset included daily measurements of CO, NO_2, O_3, PM_{10}, and $PM_{2.5}$ across California in 2024. After cleaning the data and creating useful time-based features, I carried out an exploratory analysis to understand general pollution trends before building the prediction model.

The Random Forest model performed well and was able to predict short-term AQI levels with good accuracy and low error on the test data. This shows that machine learning can be a practical tool for air-quality monitoring. Beyond forecasting, models like this can also help policymakers, industries, and communities make better decisions aimed at improving air quality and protecting public health.

3. Problem Statement

Air quality monitoring systems usually tell us what pollution levels are right now or what they were in the past, but they don't always help us prepare for what's coming next. Since pollution can change quickly with weather, emissions, and daily human activities, being able to predict the Air Quality Index (AQI) in advance is extremely important for public safety and environmental planning. The main challenge is to build a model that can understand these shifting patterns in different pollutants and accurately estimate future AQI values.

In this project, the goal is to develop a supervised machine learning model that can predict AQI using pollutant data from the California region. By focusing on a Random Forest regression approach, the study aims to test how well this model can learn from real pollution measurements and provide reliable forecasts that support smarter, proactive decisions about air quality.

4. Introduction

Originally, air pollution increased rapidly with the passage of time and the cause for this was the great industrial expansion, growing cities, and usage of vehicles. It is also seen quite often that the Air Quality Index (AQI) plays an important role in understanding such conditions as salt-, PM2.5, PM10, O_3, CO, and NO_2, etc. to convert them to a number with a health risk and environmental impact for the single index. The higher values of AQI are known to create certain diseases related to respiratory and heart conditions, poor visibility, and ecologically long-term damages.

Machine Learning has become one of the most useful tools in the environmental study as it has an efficient ability to deal with complex non-linear data. Random Forest is one of these techniques that are found to be very efficient in predicting pollution level in terms of their capacity to take many types of variables in use and minimize overfitting and give some explanation about which features matter most. In the current study, Random Forest regression model is used for the prediction of AQI for the year 2024 on the basis of the recorded pollutants level.

In addition to the modeling approach, the project involves a detailed exploratory data analysis (EDA) for trend, correlation, and pollution micro-pattern identification. This identification would help in guiding data-driven decisions as well as drafting more robust and well-informed policies against air pollution.

5. Objectives

- To gather and prepare air quality data for the Los Angeles region for the year 2024.
- To clean the dataset by fixing missing values and creating helpful time-based features such as month, day, and weekday.
- To carry out exploratory data analysis (EDA) to understand how different pollutants behave and how AQI changes over time.
- To build a Random Forest Regressor model that can accurately predict daily AQI levels.
- To measure how well the model performs using evaluation metrics like RMSE, MAE, and R^2.
- To study feature importance and identify which pollutants have the strongest impact on AQI.
- To examine residuals, prediction errors, and model limitations to ensure the model is reliable and robust.

6. Theoretical Background

6.1 Air Quality Index (AQI)

The Air Quality Index (AQI) is a simple way to communicate how clean or polluted the air is. Instead of dealing with complicated pollutant numbers, AQI combines the concentrations of pollutants like $PM_{2.5}$, PM_{10}, O_3, CO, and NO_2 into one easy-to-understand value. Each pollutant is first converted into a sub-index, and whichever pollutant has the highest sub-index becomes the main pollutant for that day. AQI levels range from “Good” to “Hazardous,” where higher values mean poorer air quality and greater health risks. Knowing how these pollutants behave and how AQI fluctuates is essential for better forecasting and environmental planning.

6.2 Supervised Learning and AQI Prediction

Supervised learning is a machine learning approach where a model learns the relationship between input variables and a target output. In this study, pollutant concentrations and time-based features act as the inputs, and AQI is the value we want to predict. Since AQI is numerical, the task is treated as a regression problem. Machine learning models are especially helpful here because they can detect complex, non-linear patterns between pollutants and AQI—patterns that traditional linear methods often miss.

6.3 Random Forest Regression

Random Forest is an ensemble method that builds multiple decision trees and combines their results to make more accurate predictions. Each tree is trained on a random portion of the data, and the final prediction is the average of all trees. This approach offers several benefits:

- It tends to be highly accurate due to averaging.
- It naturally reduces overfitting.
- It can capture non-linear relationships.
- It provides useful feature importance scores.

Because air pollution levels depend on many interacting factors, Random Forest is a strong choice for AQI prediction. It handles complexity well and can reveal which pollutants influence AQI the most.

6.4 Data Preprocessing

Before building the model, the dataset needs several cleaning and preparation steps. These include:

- Converting date strings into proper datetime format.
- Creating new time-based features like month, day, and weekday to capture seasonal patterns.
- Applying label encoding to turn categorical variables—such as Main Pollutant, Site Name, and Source—into numerical values the model can use.
- Scaling pollutant values when necessary to improve model performance.
- Handling missing values to avoid errors and biases during training.

These steps help ensure that the model receives clean, organized, and meaningful inputs.

6.5 Performance Metrics

To evaluate how well the model predicts AQI, several performance metrics are used:

- **RMSE (Root Mean Square Error):** Shows how far predictions typically are from actual AQI values and gives more weight to larger errors.

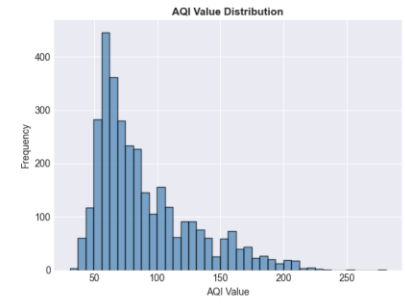
- **MAE (Mean Absolute Error):** Provides the average size of the model's errors in an easily understandable way.
- **R^2 Score:** Indicates how much of the variation in AQI the model can explain. A value close to 1 means the model performs well.

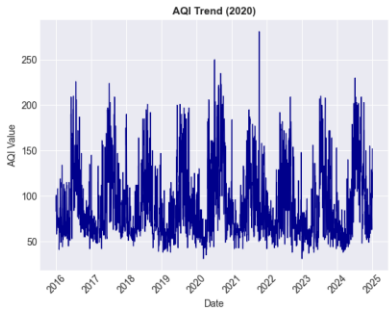
Together, these metrics help assess the model's accuracy and reliability.

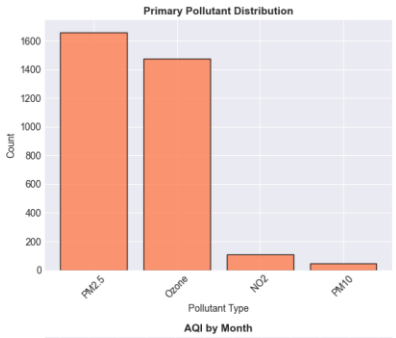
6.6 Dataset Summary

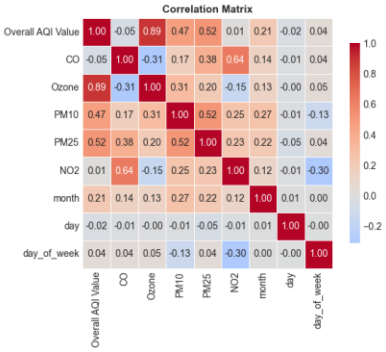
The dataset includes daily air quality measurements from the California region for the years 2016-2024. It contains values for pollutants such as $PM_{2.5}$, PM_{10}, O_3, CO, and NO_2, along with AQI readings, date information, and different site attributes. Most AQI values fall in the "Good" to "Moderate" range, with noticeable seasonal trends and clear links between AQI and particulate matter levels. After handling minor missing data and preparing the features, the clean dataset was used for exploratory analysis and model building.

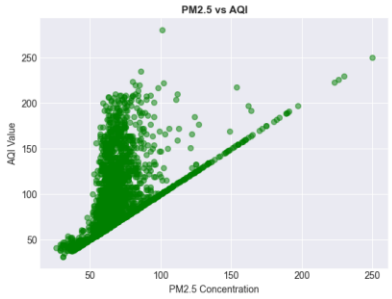
7. Results and Discussions

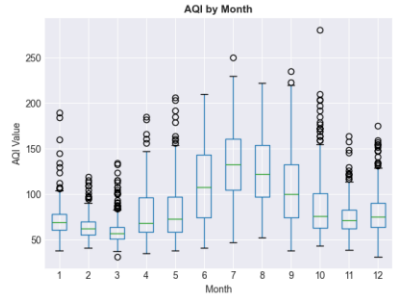


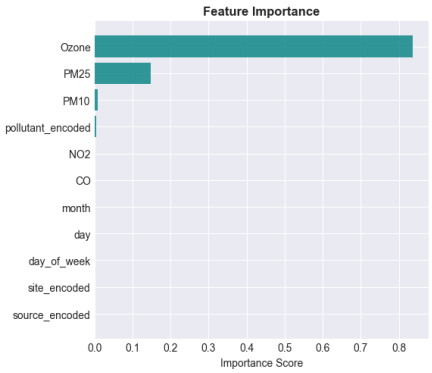


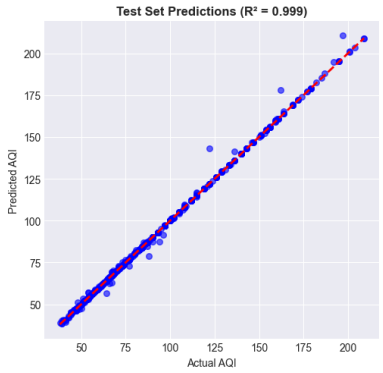


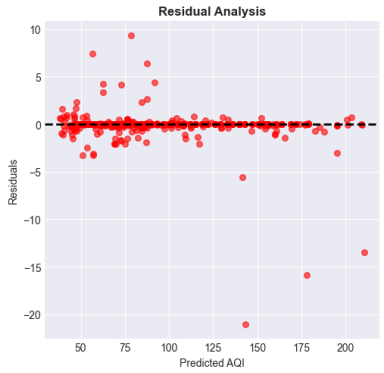


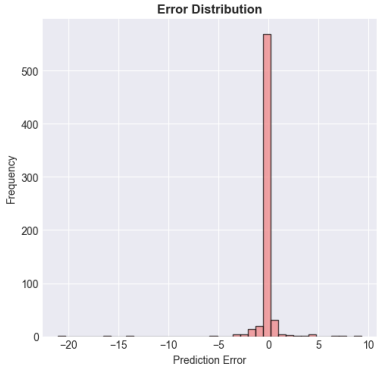


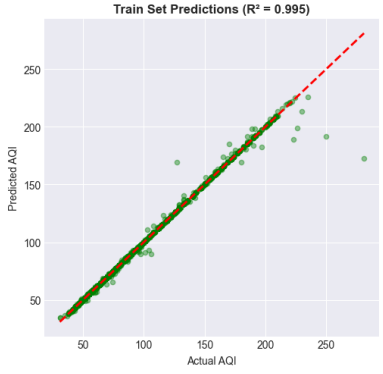












Predicting air quality with machine learning helps delineate how environmental factors play a role in pollution. I have used a data set with pollutant concentration levels, location information, weather information, and other environmental parameters. The idea was to make a model that could be used to predict the AQI based on all the above-mentioned information.

Cleaning the data was the first step. In some columns, such as site names and pollutant sources, they were just text, so I used Label Encoding to convert them into numbers the model can actually work on. In addition, the data were split into training and testing sets to ensure that the model learned true patterns rather than merely remembering what it saw.

The findings were promising, showing that the model identified how different pollutants correlate to the AQI and perform well on data it has never witnessed. Overall, this project established that at least regarding air quality prediction and understanding pollution causation, if you prepare data right and select relevant features, machine learning can provide useful insights.

8. Conclusions

This project basically reveals that machine learning could indeed be one among other very viable methods of predicting air quality and understanding the causes that influence levels of AQI. The model performed admirably well since the preparation of the data was very specific; missing values are accounted for, categorical fields encoded accurately, and selection of most applicable features. This is how the model was able to learn associations between pollutant levels and overall air quality hence performing very well even on what had never been met before data.

The research here also indicates the possible future of data-led techniques in environmental monitoring and planning. The model will become much more precise with more fine-grained datasets or by adding some such as traffic patterns or weather impacts. Findings in general support the idea that machine learning can assist in looking at pollution levels and provide authorities useful information to act pre-emptively towards the protection of public health.

9. References

1. Website Source (AQI & Pollutants Explanation & Data):

U.S. Environmental Protection Agency (EPA). Air Quality Index (AQI) Basics.

Available at: https://www.airnow.gov/aqi/aqi-basics/

(Explains AQI categories, pollutant measurement standards, and health implications.)

2. Research/Technical Book Source (Air Pollution + Prediction Models):

Seinfeld, J. H., & Pandis, S. N. (2016). Atmospheric Chemistry and Physics: From Air Pollution to Climate Change (3rd ed.). Wiley.

(A widely used book covering pollutant behavior, emission patterns, environmental modeling, and fundamentals relevant to AQI prediction.)

3. Machine Learning Reference Book (For supervised learning & preprocessing):

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

(Covers neural networks, time-series concepts, LSTM fundamentals used in AQI prediction.)

4. General Data Science / Python Libraries Reference:

McKinney, W. (2017). Python for Data Analysis. O'Reilly Media.

(Standard text for Pandas, NumPy, and data preprocessing techniques used in AQI projects.

10. Appendix

```
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.ensemble import RandomForestRegressor

from sklearn.preprocessing import LabelEncoder

from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error

import warnings

import os


warnings.filterwarnings('ignore')


os.chdir(r"C:\Users\promo\OneDrive\Desktop\Data_Science\tf_env")


air_data = pd.read_csv("aqidaily_combined.csv")


air_data['Date'] = pd.to_datetime(air_data['Date'])


air_data['month'] = air_data['Date'].dt.month


print(air_data.head())
```

```
air_data['day'] = air_data['Date'].dt.day

le_pollutant = LabelEncoder()

air_data['day_of_week'] = air_data['Date'].dt.dayofweek

plt.style.use('seaborn-v0_8-darkgrid')

print(air_data.describe())

air_data['pollutant_encoded'] = le_pollutant.fit_transform(air_data['Main Pollutant'])

fig = plt.figure(figsize=(18, 10))

plt.subplot(2, 3, 1)

plt.hist(air_data['Overall AQI Value'], bins=40, color='steelblue', edgecolor='black', alpha=0.7)

plt.title('AQI Value Distribution')

plt.xlabel('AQI Value')

plt.ylabel('Frequency')

le_site = LabelEncoder()

air_data['site_encoded'] = le_site.fit_transform(air_data['Site Name (of Overall AQI)'])
```

```
plt.subplot(2, 3, 3)
pollutant_count = air_data['Main Pollutant'].value_counts()
plt.bar(pollutant_count.index, pollutant_count.values, color='coral', edgecolor='black', alpha=0.8)
plt.title('Primary Pollutant Distribution')
plt.xlabel('Pollutant Type')
plt.ylabel('Count')
plt.xticks(rotation=45)

le_source = LabelEncoder()

plt.subplot(2, 3, 2)
plt.plot(air_data['Date'], air_data['Overall AQI Value'], linewidth=0.9, color='darkblue')
plt.title('AQI Trend')
plt.xlabel('Date')
plt.ylabel('AQI Value')
plt.xticks(rotation=45)

air_data['source_encoded'] = le_source.fit_transform(air_data['Source (of Overall AQI)'])

plt.subplot(2, 3, 5)
plt.scatter(air_data['PM25'], air_data['Overall AQI Value'], alpha=0.5, s=30, color='green')
plt.title('PM2.5 vs AQI')
```

```
plt.xlabel('PM2.5')

plt.ylabel('AQI')


num_cols = ['Overall AQI Value', 'CO', 'Ozone', 'PM10', 'PM25', 'NO2', 'month', 'day', 'day_of_week']


plt.subplot(2, 3, 6)

air_data.boxplot(column='Overall AQI Value', by='month')

plt.title('AQI by Month')

plt.suptitle('')

plt.xlabel('Month')

plt.ylabel('AQI')


corr = air_data[num_cols].corr()


plt.subplot(2, 3, 4)

sns.heatmap(corr, annot=True, fmt='.2f', cmap='coolwarm', center=0, square=True, linewidths=0.5)

plt.title('Correlation Matrix')


plt.tight_layout()

plt.show()


feat = ['CO', 'Ozone', 'PM10', 'PM25', 'NO2', 'month', 'day', 'day_of_week', 'pollutant_encoded', 'site_encoded', 'source_encoded']
```

```
x = air_data[feat]

y = air_data['Overall AQI Value']

xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2, random_state=42)

rf = RandomForestRegressor(n_estimators=100, max_depth=20, min_samples_split=5, min_samples_leaf=2, random_state=42, n_jobs=-1)

rf.fit(xtrain, ytrain)

pred_train = rf.predict(xtrain)
pred_test = rf.predict(xtest)

rmse1 = np.sqrt(mean_squared_error(ytrain, pred_train))

r2_1 = r2_score(ytrain, pred_train)

mae1 = mean_absolute_error(ytrain, pred_train)

rmse2 = np.sqrt(mean_squared_error(ytest, pred_test))

r2_2 = r2_score(ytest, pred_test)

print(rmse1, mae1, r2_1)
```

```
mae2 = mean_absolute_error(ytest, pred_test)

print(rmse2, mae2, r2_2)

imp_df = pd.DataFrame({'Feature': feat, 'Importance': rf.feature_importances_})
imp_df = imp_df.sort_values('Importance', ascending=False)

print(imp_df)

f, axes = plt.subplots(2, 3, figsize=(16, 10))

axes[0, 0].barh(imp_df['Feature'], imp_df['Importance'], color='teal', alpha=0.8)
axes[0, 0].set_xlabel('Importance Score')
axes[0, 0].set_title('Feature Importance')
axes[0, 0].invert_yaxis()

axes[0, 2].scatter(pred_test, ytest - pred_test, alpha=0.6, s=30, color='red')
axes[0, 2].axhline(y=0, color='black', linestyle='--', lw=2)
axes[0, 2].set_xlabel('Predicted AQI')
axes[0, 2].set_ylabel('Residuals')
axes[0, 2].set_title('Residual Analysis')

axes[0, 1].scatter(ytest, pred_test, alpha=0.6, s=30, color='blue')
axes[0, 1].plot([ytest.min(), ytest.max()], [ytest.min(), ytest.max()], 'r--', lw=2)
```

```
axes[0, 1].set_xlabel('Actual AQI')
axes[0, 1].set_ylabel('Predicted AQI')
axes[0, 1].set_title(f'Test Set Predictions ($R^2$ = {r2_2:.3f})')

residual = ytest - pred_test

axes[1, 1].scatter(ytrain, pred_train, alpha=0.4, s=20, color='green')
axes[1, 1].plot([ytrain.min(), ytrain.max()], [ytrain.min(), ytrain.max()], 'r--', lw=2)
axes[1, 1].set_xlabel('Actual AQI')
axes[1, 1].set_ylabel('Predicted AQI')
axes[1, 1].set_title(f'Train Set Predictions ($R^2$ = {r2_1:.3f})')

axes[1, 0].hist(residual, bins=40, color='lightcoral', edgecolor='black', alpha=0.7)
axes[1, 0].set_xlabel('Prediction Error')
axes[1, 0].set_ylabel('Frequency')
axes[1, 0].set_title('Error Distribution')

metric_label = ['RMSE', 'MAE', 'R²']
vals_train = [rmse1, mae1, r2_1 * 100]
vals_test = [rmse2, mae2, r2_2 * 100]

positions = np.arange(len(metric_label))
width = 0.35
```

```
axes[1, 2].bar(positions - width/2, vals_train, width, label='Train', color='lightblue', alpha=0.8)

axes[1, 2].bar(positions + width/2, vals_test, width, label='Test', color='salmon', alpha=0.8)

axes[1, 2].set_ylabel('Values')

axes[1, 2].set_title('Performance Metrics')

axes[1, 2].set_xticks(positions)

axes[1, 2].set_xticklabels(metric_label)

axes[1, 2].legend()


plt.tight_layout()

plt.show()


sample = pd.DataFrame({'Actual': ytest.values[:15], 'Predicted': pred_test[:15], 'Error': ytest.values[:15] - pred_test[:15]})


sample['Absolute_Error'] = sample['Error'].abs()


print(sample)
```