**AQI Prediction System: Forecasting Air Quality Using Machine Learning**

"An Automated Machine Learning-Based System for Predicting Air Quality Index (AQI) Across Multiple Cities"

**Name**: Mahnoor Safeer Abbasi

**Position**: Data Science

**Duration**: December 2024 - January 2025

**Mentor**: Abdullah Farooqi

**Organization**: 10Pearls

**AQI Prediction System: Detailed Report**

**Introduction**

The Air Quality Index (AQI) prediction system aims to forecast air quality for various cities by collecting real-time weather and air pollution data, preprocessing the information, training machine learning models, and automating the prediction process. The system provides daily forecasts, ensuring that relevant stakeholders have access to the necessary information for managing air quality issues.

**1) Data Acquisition:**

The first step in the system involves collecting data from external sources. Specifically, two key sources were used:

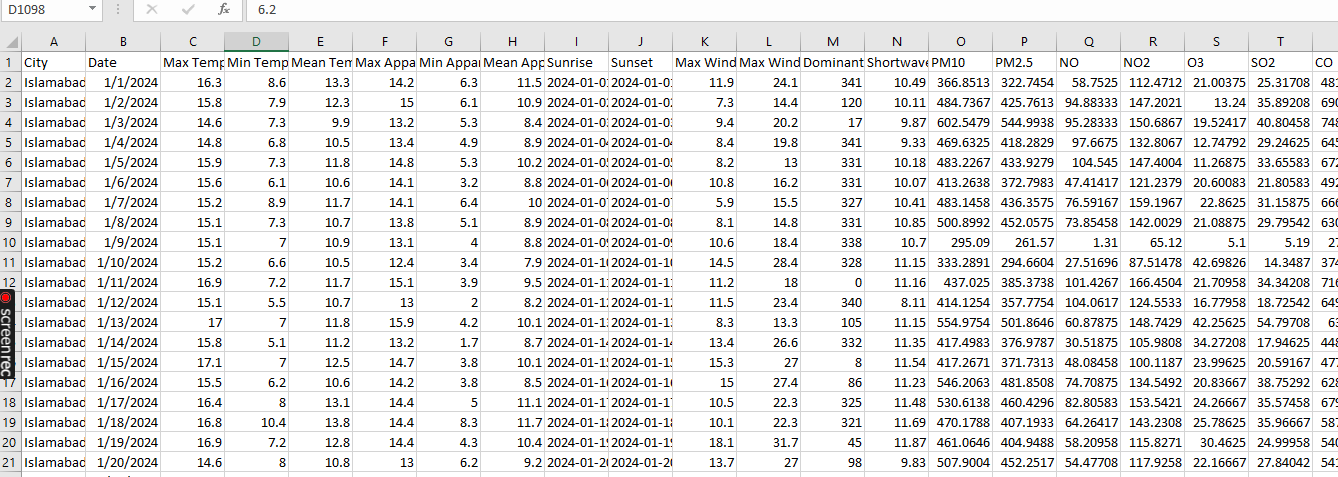
* **Open Meteo API**: This API was used to fetch weather data, including parameters like temperature, wind speed, radiation, and dominant wind direction. This data provides essential context for AQI predictions, as weather conditions can influence air quality.
* **OpenWeather API**: The air pollution data was retrieved from this API, providing crucial metrics such as concentrations of PM10, PM2.5, NO2, SO2, CO, and O3, along with the AQI value for the respective cities.

These APIs were queried to fetch daily forecasts, ensuring that data was available for cities like Karachi, Lahore, and Islamabad. This real-time data was essential to ensure accurate predictions and forecasting.

**2) Data Preprocessing and Merging:**

Once the data was fetched, it was merged and preprocessed:

* **Weather Data**: Weather data from Open Meteo was cleaned and structured to ensure uniformity. Features such as temperature, wind speed, and radiation were extracted and cleaned to be used for model training.
* **Air Pollution Data**: The pollution data from OpenWeather was processed similarly. Key pollutants (PM10, PM2.5, NO2, SO2, CO, and O3) were extracted, and missing values were handled using imputation techniques.
* **Merging**: Both datasets were merged on the City and Date columns to combine weather and pollution information for each city on a given day. This combined dataset served as the input for model training.

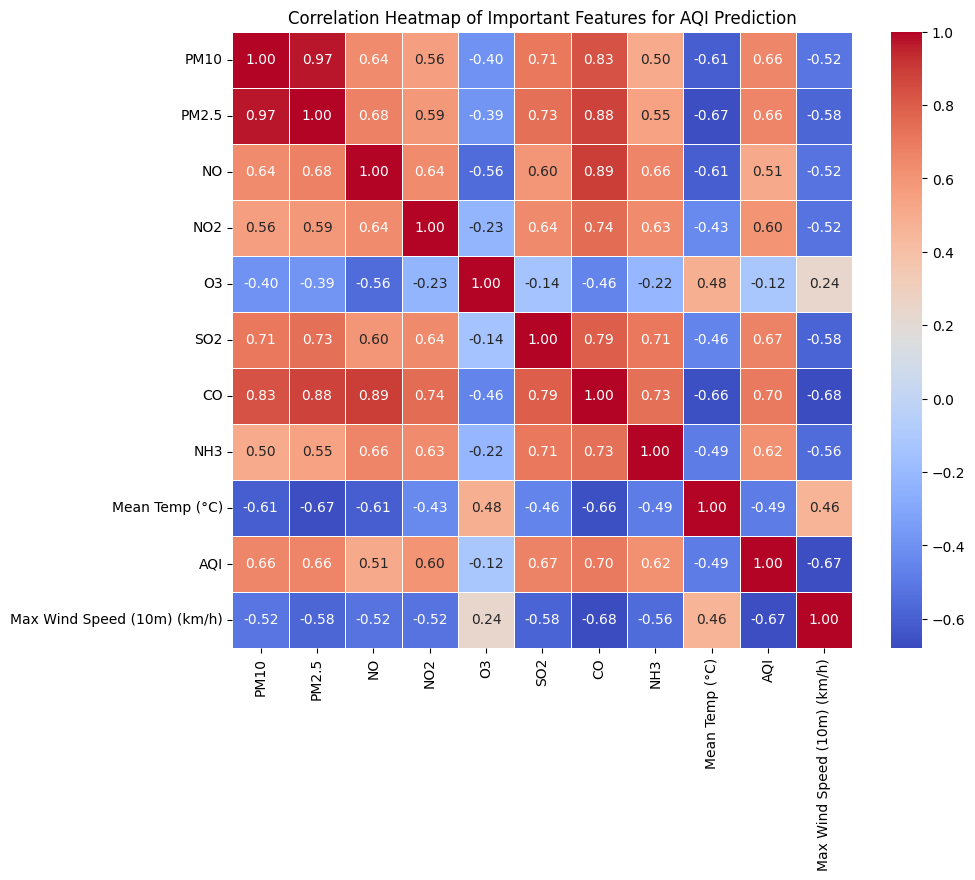


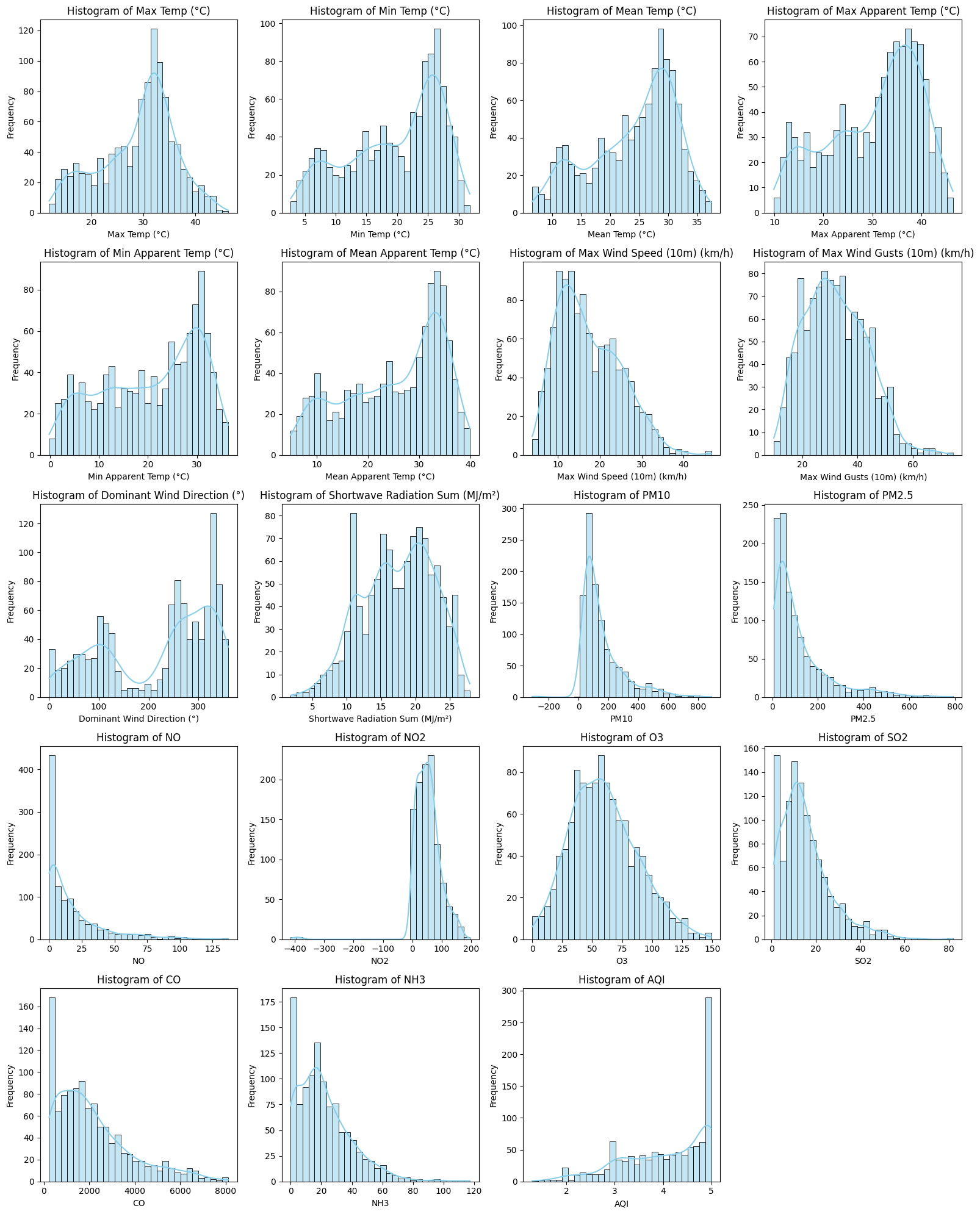
Merged data after combining APIs

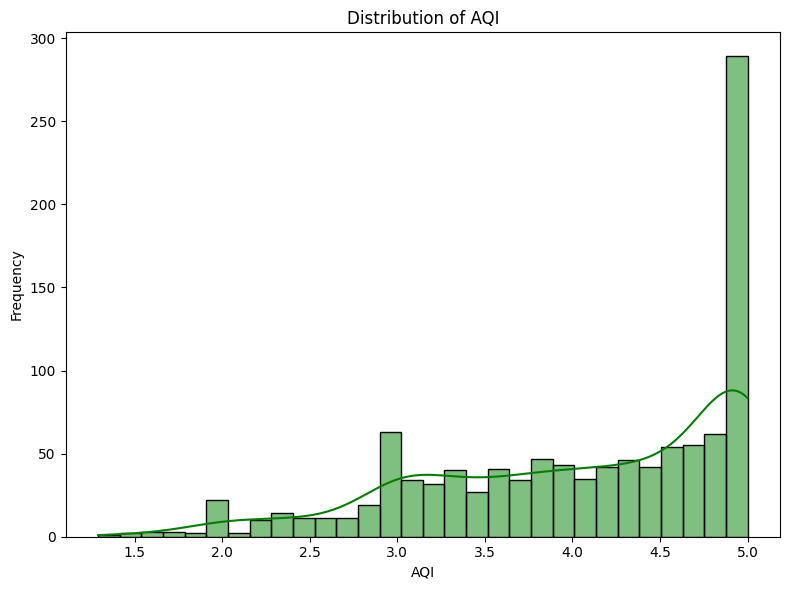
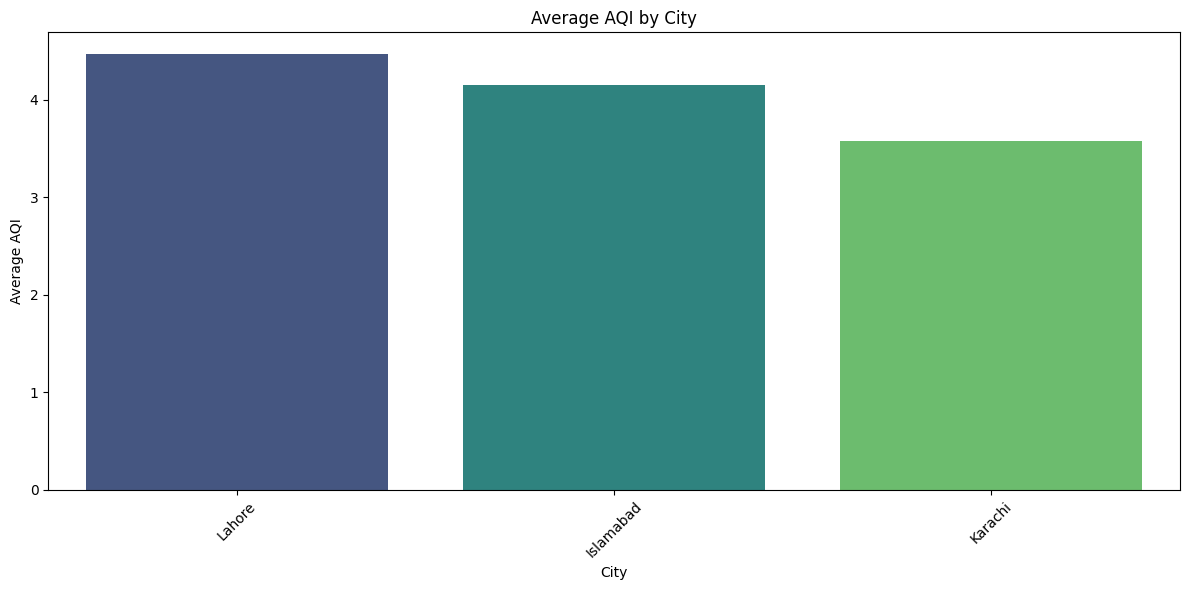
**3) Exploratory Data Analysis (EDA):**

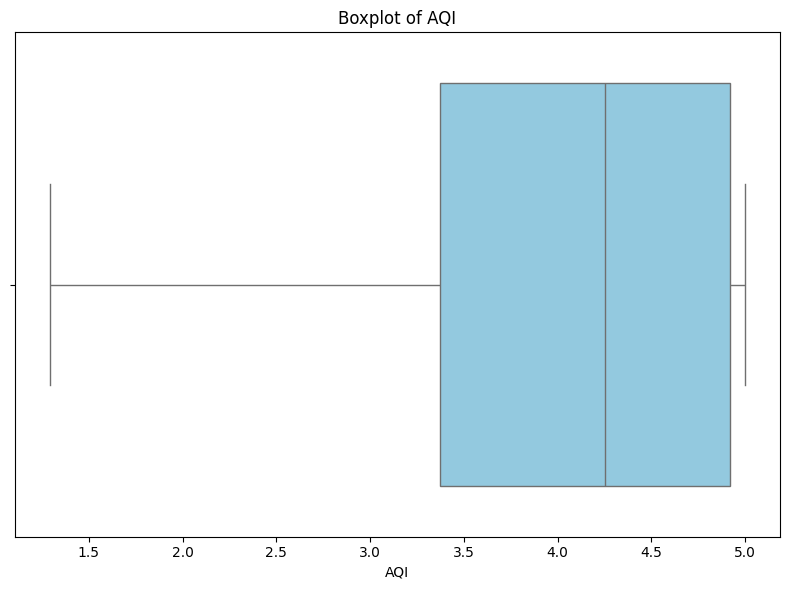
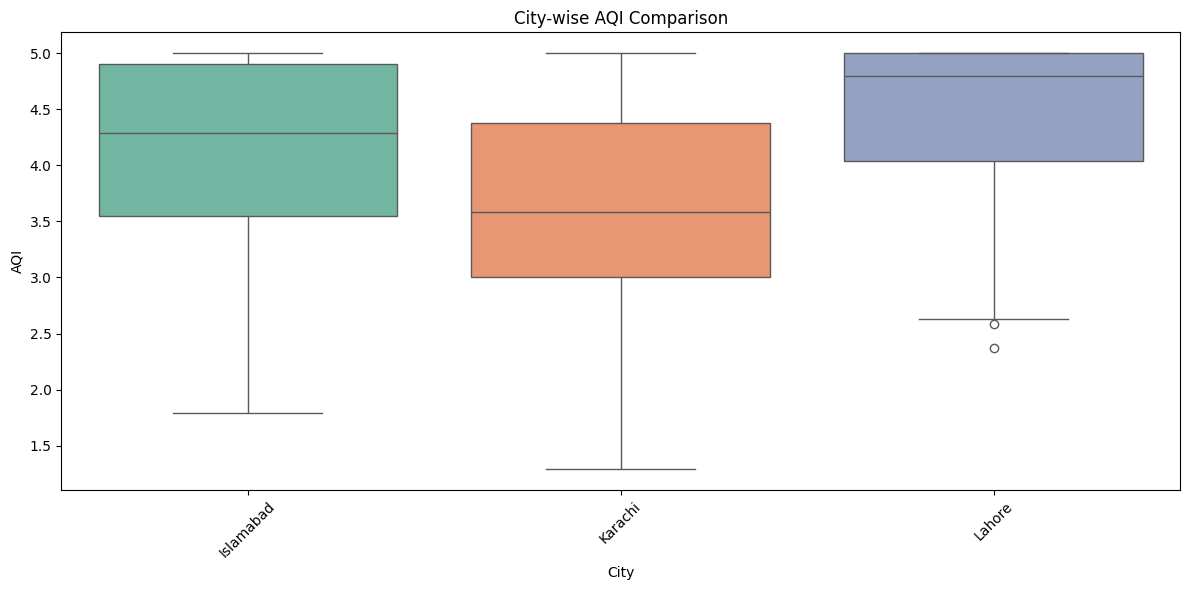
EDA was an essential step in understanding the structure of the data, identifying patterns, and detecting outliers. Several key steps were performed during the EDA process:

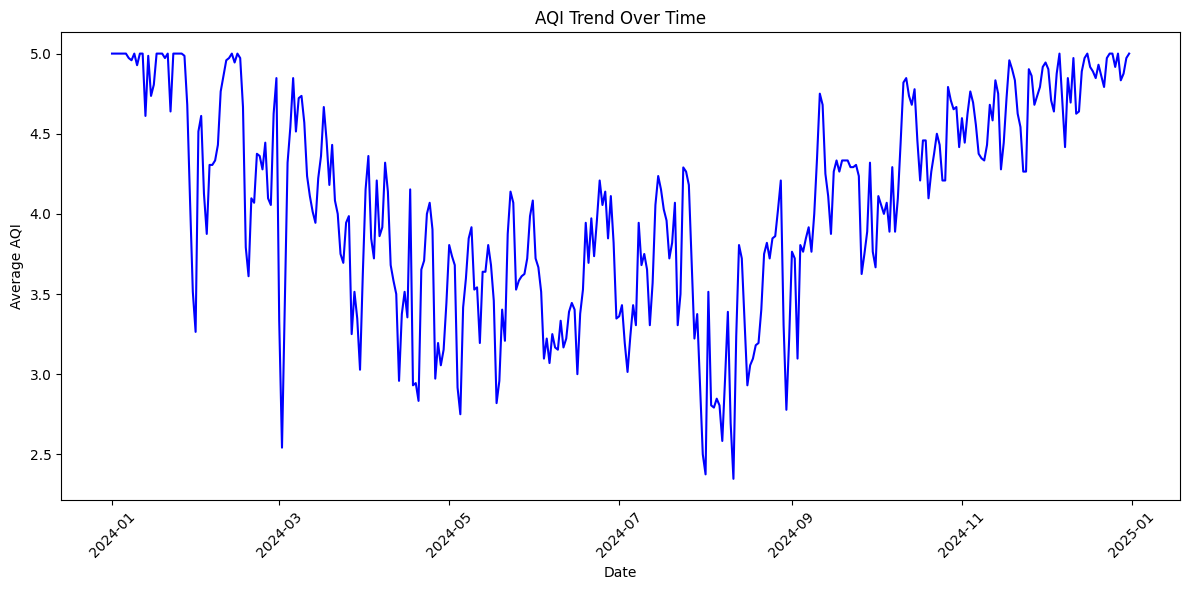
* **Visualizing Correlations**: The relationships between various features such as temperature, wind speed, and pollutant concentrations were explored using correlation matrices and pair plots. These helped identify which variables were most influential in predicting AQI.





* **Distribution of AQI**: The distribution of AQI values across different cities was plotted to understand its spread and variance. This step helped identify whether AQI was consistently high or low in certain areas, which could inform further analysis. 
* **Weather and Pollution Impact**: The impact of weather conditions on air pollution was analyzed to identify if higher temperatures, for instance, correlated with worse air quality. Scatter plots and box plots were used to understand these relationships. City wise Comparison plot





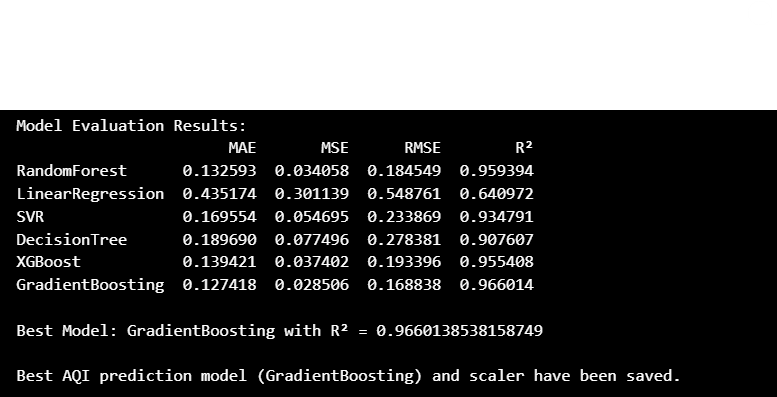
* **Missing Data Analysis**: The dataset was checked for any missing or null values, and appropriate imputation techniques (e.g., mean imputation) were applied to handle them.

This thorough exploration laid the foundation for model selection and provided insights into the underlying patterns within the data.

**4) Machine Learning Model Selection and Training:**

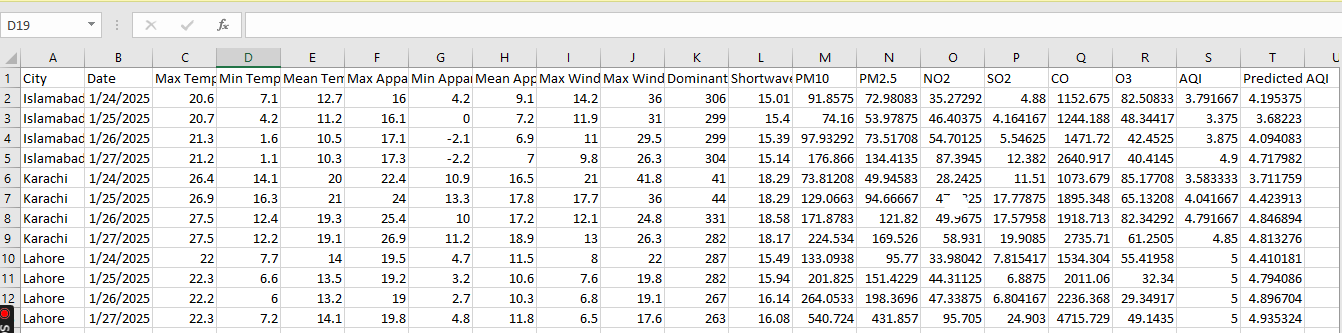
Various machine learning models were applied to the preprocessed data to predict AQI values. The models tested included:

* **Linear Regression**
* **Random Forest**
* **Gradient Boosting**

After comparing the models based on performance metrics such as Mean Squared Error (MSE) and R² score, **Gradient Boosting** emerged as the best-performing model, achieving an accuracy of 96%. This model was chosen for its ability to handle complex, nonlinear relationships and for its superior performance in predicting AQI values compared to the other models. 

**5) Forecasting and Model Evaluation:**

The trained Gradient Boosting model was tested on a forecast dataset, and the predictions closely matched the actual AQI values. The model demonstrated its ability to generalize and predict AQI effectively for unseen data, further validating its potential for deployment.



**6) Automation with Apache Airflow:**

To automate the entire pipeline, Apache Airflow was used as a scheduler. This allowed for a fully automated process, running daily to fetch new data from the APIs, preprocess it, merge the datasets, and make predictions for the next 3 days. This was an essential step in ensuring that forecasts were updated automatically without manual intervention.

A screenshot of a computer

Description generated with very high confidence

**7) Integration with Hopsworks:**

To improve the scalability and management of the system, **Hopsworks** was used to store the processed features and the trained model. Key steps in this process included:

* **Feature Store**: The preprocessed data (weather, pollution, and AQI predictions) was stored in a **Hopsworks Feature Store** for easier access and management. This also facilitated consistent feature versioning and ensured that the latest data was always available for retraining and model updates.
* **Model Registry**: The trained **Gradient Boosting model** and the **scaler model** were registered in Hopsworks' **Model Registry**, enabling easy tracking of model versions and facilitating deployment into production environments.
* **Training Pipeline**: A training pipeline was created within Hopsworks, allowing for continuous retraining of the model with the latest data from the feature store. This helped ensure that the predictions remained up to date and improved over time.

However, a challenge was encountered during the integration with Hopsworks: **Enterprise-level features** were required to establish a seamless connection between Airflow and Hopsworks. Due to this limitation, the current workflow involves running Airflow locally, saving the data on a local system, and then manually updating the Feature Store.

A screenshot of a computer

Description generated with very high confidence

**8) Streamlit UI:**

A **Streamlit** dashboard was created to provide an intuitive and interactive user interface for the AQI prediction system. The dashboard included:

* **City Selection**: Users could select a city (Islamabad, Karachi, Lahore, or All Cities) to view the AQI prediction for that city.
* **Date-wise Predictions**: The dashboard displayed the AQI predictions for different dates, giving users the ability to track changes in air quality over time.
* **Visualizations**: Various visualizations such as AQI prediction trends, distribution of AQI values, and detailed city-wise data were incorporated to help users understand the predictions better.

A screenshot of a computer

Description generated with very high confidence

A screenshot of a computer

Description generated with very high confidence

A screenshot of a computer

Description generated with very high confidence

**9) Challenges Faced:**

* **Hopsworks Connection Limitation**: The biggest challenge encountered was the inability to connect **Apache Airflow** directly with **Hopsworks** due to the enterprise-level feature limitation. This restricted the ability to fully automate the process on the cloud, forcing us to handle data updates and model retraining manually.
* **Data Quality and Integration**: Handling missing data and integrating multiple sources of weather and pollution data proved challenging, requiring multiple preprocessing steps to ensure consistency and completeness.

**10) Future Work:**

* **Improved Scheduling and Automation**: The goal is to overcome the Hopsworks connection limitation and integrate Apache Airflow with Hopsworks, enabling a fully automated pipeline that updates the feature store and retrains the model without manual intervention.
* **Deployment on Serverless Infrastructure**: The next step will involve deploying the AQI prediction model on a serverless infrastructure, reducing operational overhead and making the system more scalable. This would also enable more flexible resource allocation based on demand.
* **Expanding to More Cities and Features**: The system could be expanded to include more cities and additional weather and pollution features for more granular predictions.

**Conclusion**

The AQI prediction system provides an automated and scalable solution for forecasting air quality based on real-time weather and pollution data. By combining robust data preprocessing, machine learning modeling, and visualization, the system delivers accurate predictions to help users track and manage air quality. Although there were challenges with integrating Apache Airflow and Hopsworks, the system is operational and has a clear roadmap for future enhancements, making it a valuable tool for environmental management.