**Air Quality Index (AQI) Prediction and Web Application Deployment**

**1. Introduction**

The objective of this project is to predict the Air Quality Index (AQI) for the next three days based on historical and environmental data, and to deploy this prediction model on a web application. The web app will allow users to visualize historical AQI trends and receive real-time AQI predictions for the upcoming days. The data utilized for prediction includes environmental features such as temperature, humidity, and levels of different pollutants like CO, NO2, PM2.5, and others.

**2. Data Collection and Feature Store Integration**

Data for this project is gathered and managed using the Hopsworks platform, which provides a centralized Feature Store. This allows for efficient versioning and retrieval of features, and is essential for model training and prediction. The collected dataset includes not only the historical AQI levels but also several environmental variables, such as pollutants and weather data (temperature and humidity).

The primary features used for prediction include:

* Components of various pollutants (e.g., SO2, PM2.5, CO, NO2).
* Temperature and humidity readings.
* Lag features, capturing past AQI values to understand trends.

This data is stored in a Feature Group within Hopsworks, which is continuously updated to ensure the model has access to the most recent data for real-time predictions.

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

Before building the predictive model, an essential step was performed to understand the dataset thoroughly through **Exploratory Data Analysis (EDA)**. EDA helps to uncover patterns, detect outliers, and understand relationships between variables. Here's a summary of the key EDA steps conducted:

1. **Data Inspection**: The dataset was first inspected to identify any missing values or anomalies. Missing values were handled either through imputation or removal based on the nature of the data and the significance of the missing entries.
2. **Descriptive Statistics**: Descriptive statistics, such as mean, median, and standard deviation, were calculated for key features like AQI, temperature, and pollutants. This helped to summarize the data and identify the central tendency and spread of values.
3. **Correlation Analysis**: A correlation matrix was generated to explore how different features relate to the target variable (AQI). Strong correlations between certain pollutants and AQI were observed, confirming the relevance of these features in predicting air quality levels.
4. **Visualizations**:
   * **Histograms**: Distribution of features like AQI, PM2.5, CO, NO2, and temperature were plotted to understand their spread and detect any skewness.
   * **Pair Plots**: Scatter plots were created between AQI and its associated features to examine relationships.
   * **Time-Series Plots**: Time series visualizations were used to observe how AQI changes over time and to identify seasonal patterns or trends.
5. **Outlier Detection**: Outliers in features such as pollutants were identified using box plots, which could be indicative of errors or rare events in the data.

The insights from EDA helped to refine the feature engineering process, including creating lag features for AQI and deciding which pollutant concentrations would be included in the model.

**4. Model Development**

For predicting AQI levels, two machine learning models—Random Forest Regressor and XGBoost Regressor—were trained using the pre-processed data. After evaluating both models, the XGBoost model was selected based on its superior performance in terms of prediction accuracy.

The trained model was then registered in the Hopsworks Model Registry, ensuring it can be easily accessed and versioned for future use. This also enables the model to be downloaded for deployment in the web application.

The model was evaluated using standard performance metrics such as the R² score, Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The XGBoost model performed better in terms of error metrics, making it the preferred choice for predicting AQI levels.

**5. Web Application Development Using Streamlit**

The web application was developed using **Streamlit**, a Python library that allows for quick and easy development of interactive web applications. The app is designed to provide the following features:

1. **Historical AQI Trends**: The app visualizes historical AQI trends, giving users insights into past air quality levels. This is done using interactive line charts, allowing users to explore AQI trends over time.
2. **Current AQI**: The app displays the most recent AQI value, with a color-coded indicator that suggests the quality of the air (Good, Moderate, or High). This provides real-time information to users, helping them assess the air quality in their region.
3. **Predicted AQI for the Next Three Days**: The app provides predictions for the next three days based on the current AQI and other environmental factors. These predictions are displayed alongside their corresponding AQI levels (Good, Moderate, or High), helping users plan their activities accordingly.
4. **Optimization**: The data-fetching process is optimized using caching, which reduces the overhead of repeatedly querying the Feature Store. This makes the app more responsive and efficient, especially when dealing with large datasets.

The Streamlit app was further enhanced with interactive visualizations using **Plotly**, which allows for clear, easy-to-understand graphs for both historical data and predictions.

**6. Model Interpretability with SHAP**

To provide users with insights into the model’s decision-making process, SHAP (SHapley Additive exPlanations) was integrated. SHAP values explain how much each feature contributes to a particular prediction, providing transparency into the model’s behavior. By visualizing the SHAP values, users can understand which features (e.g., temperature, PM2.5 levels) are most influential in determining AQI predictions.

Two types of SHAP visualizations were used:

* **Waterfall Plot**: This plot shows how individual feature values contribute to the final prediction for a specific instance (e.g., a day’s AQI prediction).
* **Summary Plot**: This plot provides a summary of feature importance across multiple predictions, highlighting which features have the most significant impact on the model.

**7. Deployment and Access**

The Streamlit app was deployed locally and exposed to the web using **LocalTunnel**, a tool that provides secure URLs to access local servers over the internet. This allows users to access the AQI prediction app from anywhere, making it publicly available for real-time AQI forecasts.

To make app serverless, app is then deployed on **Streamlit Cloud**.

**8. Conclusion**

This project successfully implemented a machine learning-based AQI prediction model, integrated it with a user-friendly web application, and deployed the application for real-time AQI predictions. The app allows users to access historical AQI data, view the current AQI status, and receive predictions for the next three days.

By providing real-time AQI predictions and visualizations, this tool can serve as a valuable resource for individuals and organizations looking to monitor and respond to air quality changes.