Project Summary: Air Quality Prediction Using Linear Regression

1. Project Objective

The objective of this project is to predict air quality using linear regression. By analyzing historical air quality data and related environmental factors, we aim to develop a model to predict the Air Quality Index (AQI).

2. Data Collection

The dataset used in this project is sourced from an Excel file named `AirQualityUCI.xlsx`. The dataset includes various features related to air quality and environmental conditions such as:

- Date and time
- Concentrations of pollutants (e.g., PM2.5, PM10, NO2, CO, SO2, O3)
- Meteorological data (e.g., temperature, humidity, wind speed, atmospheric pressure)

3. Data Preprocessing

- **Loading Data**: The dataset is loaded into a pandas DataFrame.

```
import pandas as pd
df = pd.read_excel('AirQualityUCI.xlsx')
```

- **Initial Inspection**: Displaying the first few rows, basic information, and summary
  ```
  print(df.head())
  print(df.info())
  print(df.describe())
  ```

- **Unique Values Count**: Counting the unique values for each column to understand the categorical features.

```
print(df.nunique())
```

- **Visualizing Missing Values**: A heatmap is used to visualize the missing values in the dataset.

```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(10, 6))
sns.heatmap(df.isnull(), cbar=False, cmap='viridis')
plt.title('Missing Values Heatmap')
plt.show()
```

4. Exploratory Data Analysis (EDA)

- **Histograms**: Histograms are plotted for numerical features to understand their distributions.

```
python
Copy code
df.hist(bins=30, figsize=(20, 15))
plt.suptitle('Histograms of Numeric Columns')
plt.show()
```

- **Box Plots**: Box plots are created to visualize the distribution and detect outliers.

```
plt.figure(figsize=(10, 5))
sns.boxplot(data=df)
plt.xticks(rotation=90)
plt.show()
```

- **Correlation Matrix and Heatmap**: A correlation matrix and heatmap are used to identify relationships between variables.

```
corr_matrix = df.corr()
plt.figure(figsize=(12, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix Heatmap')
plt.show()
```

5. Feature Engineering

- **Date Features**: Extracting new features such as month, day, and hour from the date column.
- **Transformations**: Applying transformations to skewed features to stabilize variance.
- **Categorical Encoding**: Converting categorical variables into numerical format using one-hot encoding.

6. Feature Selection

- **Correlation Analysis**: Selecting features highly correlated with AQI.
- **Variance Inflation Factor (VIF)**: Calculating VIF to detect multicollinearity among predictors.

7. Model Building

- **Train-Test Split**: Splitting the dataset into training and testing sets.

```
from sklearn.model_selection import train_test_split
X = df.drop('AQI', axis=1)
y = df['AQI']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

- **Linear Regression Model**: Building a linear regression model using the training set.

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
```

- **Model Evaluation**: Evaluating the model using R-squared, MAE, MSE, and RMSE on the test set.

```
from sklearn.metrics import mean_squared_error, mean_absolute_error,
r2_score
y_pred = model.predict(X_test)
print(f'R2 Score: {r2_score(y_test, y_pred)}')
print(f'MAE: {mean_absolute_error(y_test, y_pred)}')
print(f'MSE: {mean_squared_error(y_test, y_pred)}')
print(f'RMSE: {np.sqrt(mean_squared_error(y_test, y_pred))}')
```

8. Model Improvement

- **Feature Scaling**: Applying standardization to scale the features.
- **Regularization**: Using Ridge and Lasso regression to handle overfitting.
- **Cross-Validation**: Employing cross-validation techniques to ensure model robustness.

9. Results and Findings

- **Performance Metrics**: Reporting the final model's performance metrics (R-squared, MAE, MSE, RMSE).
- **Feature Importance**: Identifying the most significant predictors of AQI.
- **Residual Analysis**: Using residual plots to check for patterns indicating model shortcomings.

10. Conclusion

The linear regression model successfully predicts air quality with a reasonable degree of accuracy. The model's performance can be further improved with advanced techniques and additional data.

11. Future Work

- **Incorporate Additional Data**: Including more environmental and geographical factors to enhance model accuracy.
- **Advanced Modeling Techniques**: Exploring more complex machine learning algorithms such as Random Forest, Gradient Boosting, or Neural Networks.
- **Real-Time Prediction**: Developing a real-time air quality monitoring system that continuously updates the model with new data.

12. Reporting and Visualization

- **Documentation**: Creating a comprehensive report detailing the analysis, model development, and results.
- **Visualization**: Using various plots and charts to visualize data, model predictions, and feature importance, aiding effective communication of findings.