Data Science Toolbox: Python Programming

PROJECT REPORT

(Project Semester January-April 2025)

(Air Quality Monitoring For Agriculture)

Submitted by:-

Shubhangi Gupta

Registration No..- 12308668

Programme and Section ..-B. Tech (CSE) and K23DP

Course Code ..-INT375

Source Link:- https://archive.ics.uci.edu/dataset/360/air+quality

Under the Guidance of

Dr. Dhiraj Kapila

UID: 23509

Discipline of CSE/IT

Lovely School of Computer Science

Lovely Professional University, Phagwara

CERTIFICATE

This is to certify that Shubhangi Gupta with Registration no.- 12308668 has completed INT375 project titled, **"Air Quality Monitoring For Agriculture"** under my guidance and supervision. To the best of my knowledge, the present work is the result of her original development, effort and study.

Name of the Supervisor:- Dr. Dhiraj Kapila
Designation of the Supervisor-Asst. Professor
School of Computer Science
Lovely Professional University
Phagwara, Punjab.

Date: 12th April 2025

DECLARATION

I, Shubhangi Gupta, student of B.tech (CSE) under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: 12th April 2025 Signature

Registration No.- 12308668 Name of the student

Shubhangi Gupta

Acknowledgement:-

I express my sincere gratitude to Dr. Dhiraj Kapila, Asst. Professor, for their invaluable guidance and support throughout this project. I am also thankful to the Discipline of CSE/IT, Lovely Professional University, for providing the necessary resources and environment. Finally, I acknowledge my peers and family for their encouragement.

Table of Contents

List of Figures

List of Tables

1. Introduction

Air quality significantly impacts agriculture, as pollutants like carbon monoxide (CO), nitrogen oxides (NOx), and benzene (C6H6) can reduce crop yield and affect plant health. This project aims to analyze air quality data to monitor these pollutants and predict CO levels using machine learning models, providing actionable insights for farmers to mitigate adverse effects and optimize agricultural practices.

2. Source of Dataset

The dataset, Air Quality.csv, contains hourly measurements of air pollutants and weather parameters. Key columns include:

- **CO(GT): Carbon monoxide concentration (mg/m³)**
- **NOx(GT): Nitrogen oxides concentration (ppb)**
- **NO2(GT): Nitrogen dioxide concentration (µg/m³)**
- **C6H6(GT): Benzene concentration (µg/m³)**
- **T: Temperature (°C)**
- **RH: Relative humidity (%)**
- **AH: Absolute humidity (g/m³)**
- **Datetime: Timestamp**

3. EDA Process

Exploratory Data Analysis (EDA) was conducted to understand the dataset's structure and prepare it for modeling. Steps included:

- **Loading the dataset using Pandas.**
- **Handling missing values and dropping unnecessary columns.**
- **Computing statistical summaries and correlations.**
- **Visualizing distributions, trends, and relationships using Matplotlib and Seaborn.**
- **Identifying outliers and temporal patterns.**

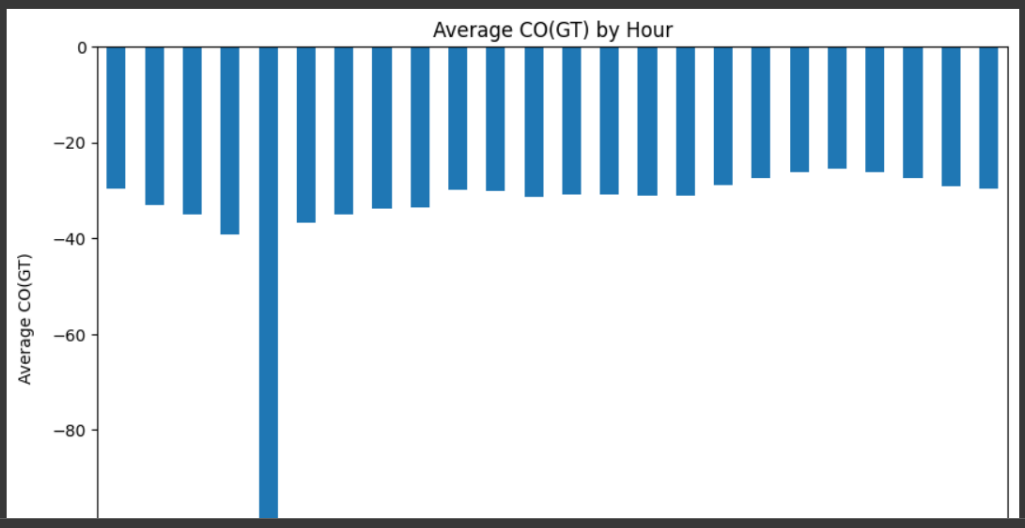
4. Analysis on Dataset

4.1 Summary Statistics

- **Introduction: Summarize statistical measures of pollutants and weather parameters.**
- **General Description: Calculate mean, median, standard deviation, min, and max to understand data distribution.**
- **Specific Requirements, Functions, and Formulas: Used describe(); mean = $\Sigma x/n$, median = middle value, std = $\sqrt{(\Sigma(x - \mu)^2/n)}$.**
- **Analysis results:-**

BAR PLOT

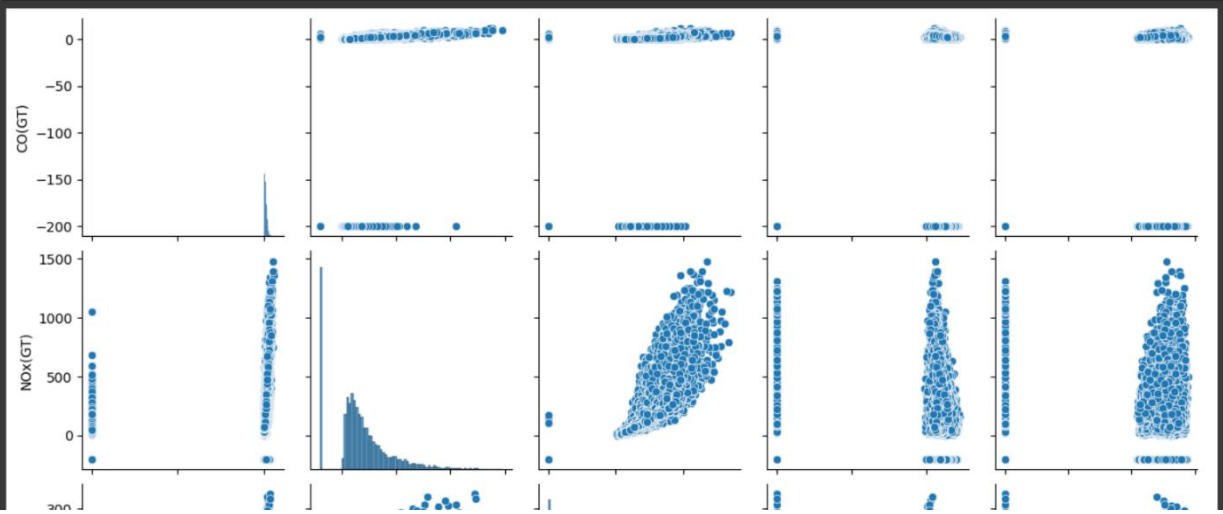
```
plt.figure(figsize=(10, 6))
df_cleaned.groupby('Hour')['CO(GT)'].mean().plot(kind='bar')
plt.xlabel('Hour of Day')
plt.ylabel('Average CO(GT)')
plt.title('Average CO(GT) by Hour')
plt.show()
```



✓ 0s completed at 9:33 PM

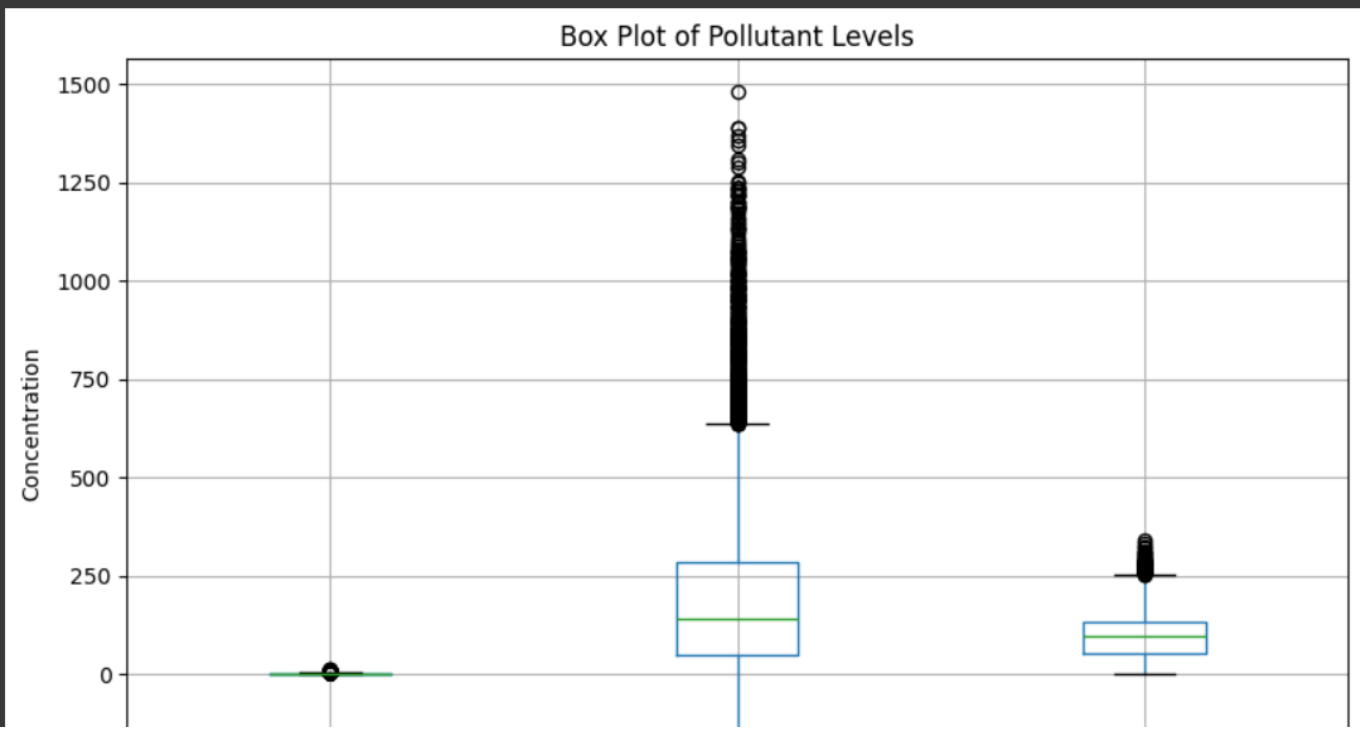
PAIR PLOT

```
sns.pairplot(df_cleaned[['CO(GT)', 'NOx(GT)', 'NO2(GT)', 'T', 'RH']])
plt.show()
```



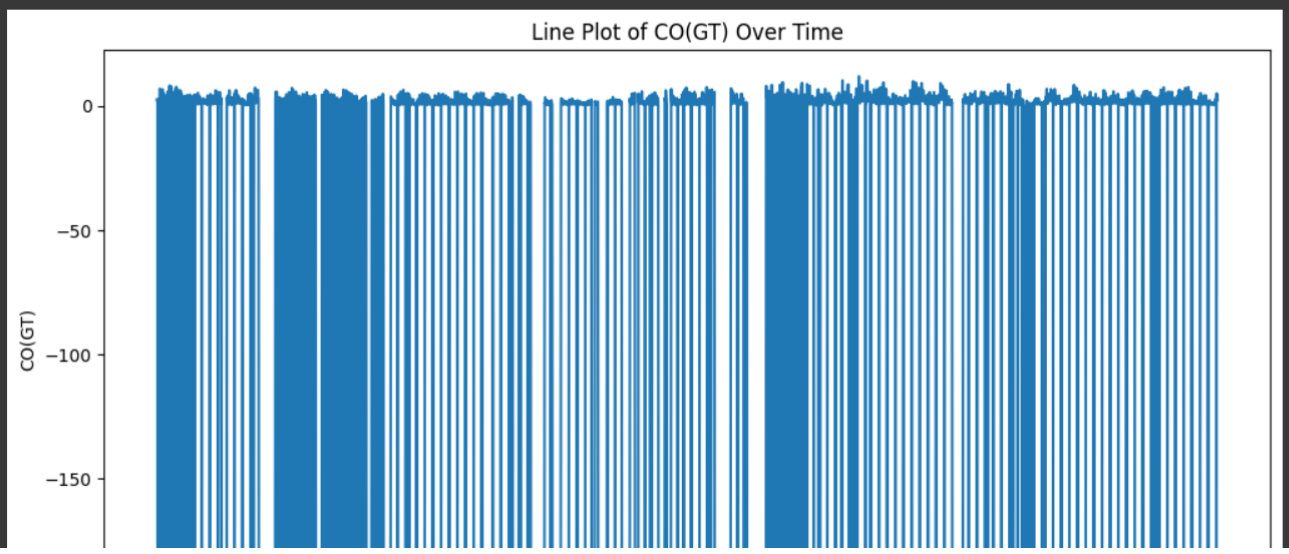
BOX PLOT

```
plt.figure(figsize=(10, 6))
df_cleaned.boxplot(column=['CO(GT)', 'NOx(GT)', 'NO2(GT)'])
plt.ylabel('Concentration')
plt.title('Box Plot of Pollutant Levels')
plt.show()
```



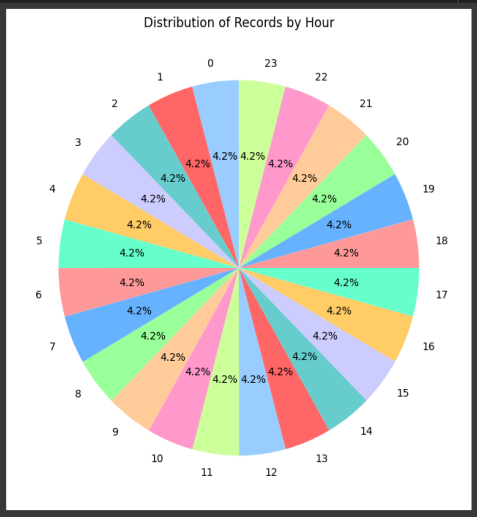
LINE PLOT

```
plt.figure(figsize=(12, 6))
plt.plot(df_cleaned['Datetime'], df_cleaned['CO(GT)'])
plt.xlabel('Datetime')
plt.ylabel('CO(GT)')
plt.title('Line Plot of CO(GT) Over Time')
plt.xticks(rotation=45)
plt.show()
```



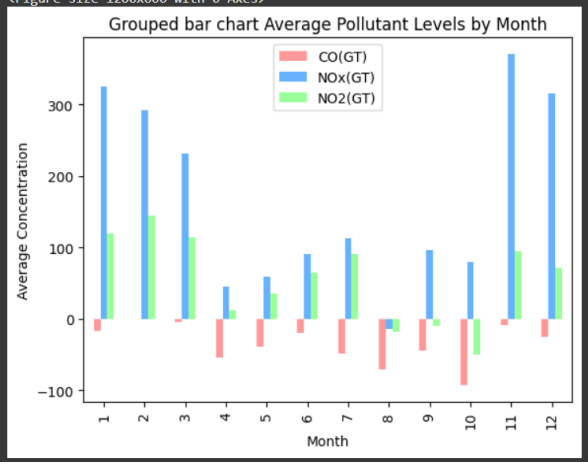
PIE CHART

```
plt.figure(figsize=(8, 8))
hourly_counts = df_cleaned['Hour'].value_counts()
plt.pie(hourly_counts, labels=hourly_counts.index, autopct='%1.1f%%', colors=['#FF9999', '#66B2FF', '#99FF99', '#FFCC99', '#FF99CC', '#CCFF99', '#99CCFF', '#FF6666', '#66CCCC', '#CCCCFF', '#FFCC66', '#66FFCC'])
plt.title('Distribution of Records by Hour')
plt.show()
```

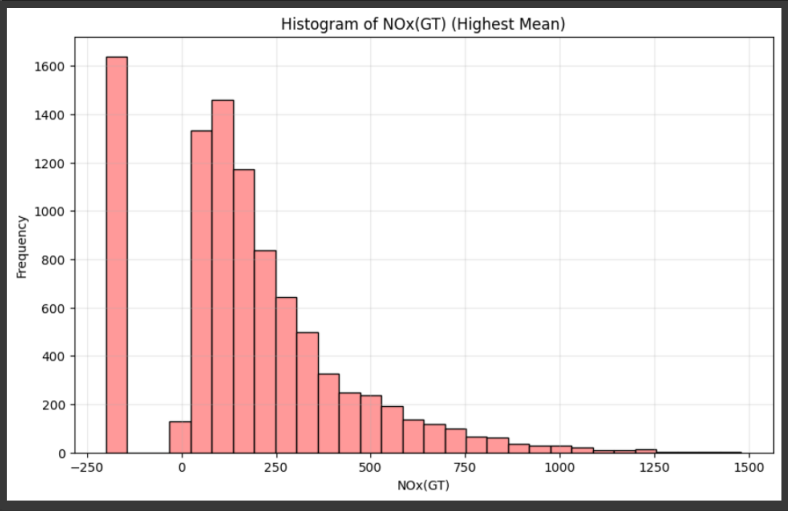



```
plt.figure(figsize=(12, 6))
pollutants_by_month = df_cleaned.groupby('Month')[['CO(GT)', 'NOx(GT)', 'NO2(GT)']].mean()
pollutants_by_month.plot(kind='bar', color=['#FF9999', '#66B2FF', '#99FF99'])
plt.xlabel('Month')
plt.ylabel('Average Concentration')
plt.title('Grouped bar chart Average Pollutant Levels by Month')
plt.show()
```

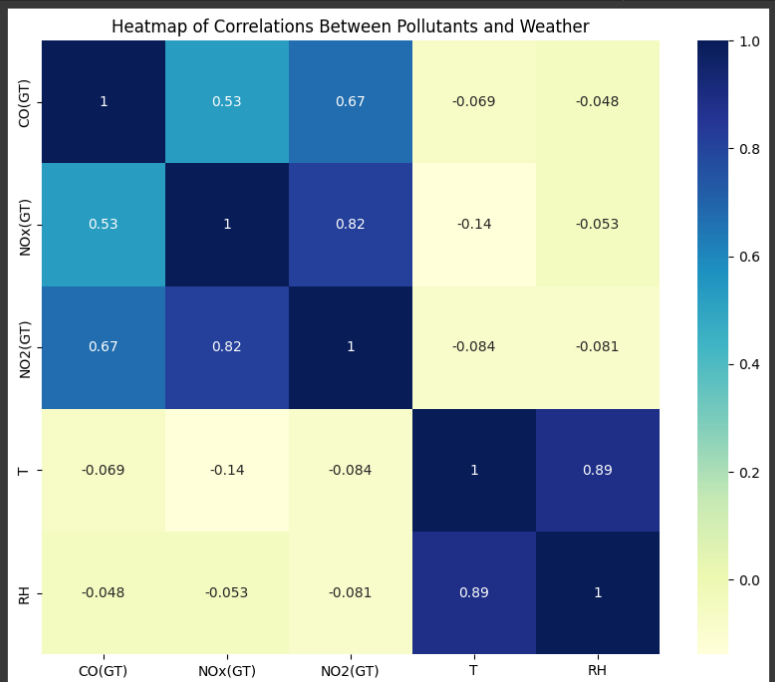
```
<Figure size 1200x600 with 0 Axes>
```



```
plt.figure(figsize=(10, 6))
max_col = df_cleaned[['CO(GT)', 'NOx(GT)', 'NO2(GT)']].mean().idxmax()
df_cleaned[max_col].hist(bins=30, color='#FF9999', edgecolor='black')
plt.xlabel(max_col)
plt.ylabel('Frequency')
plt.title(f'Histogram of {max_col} (Highest Mean)')
plt.grid(True, alpha=0.3)
plt.show()
```



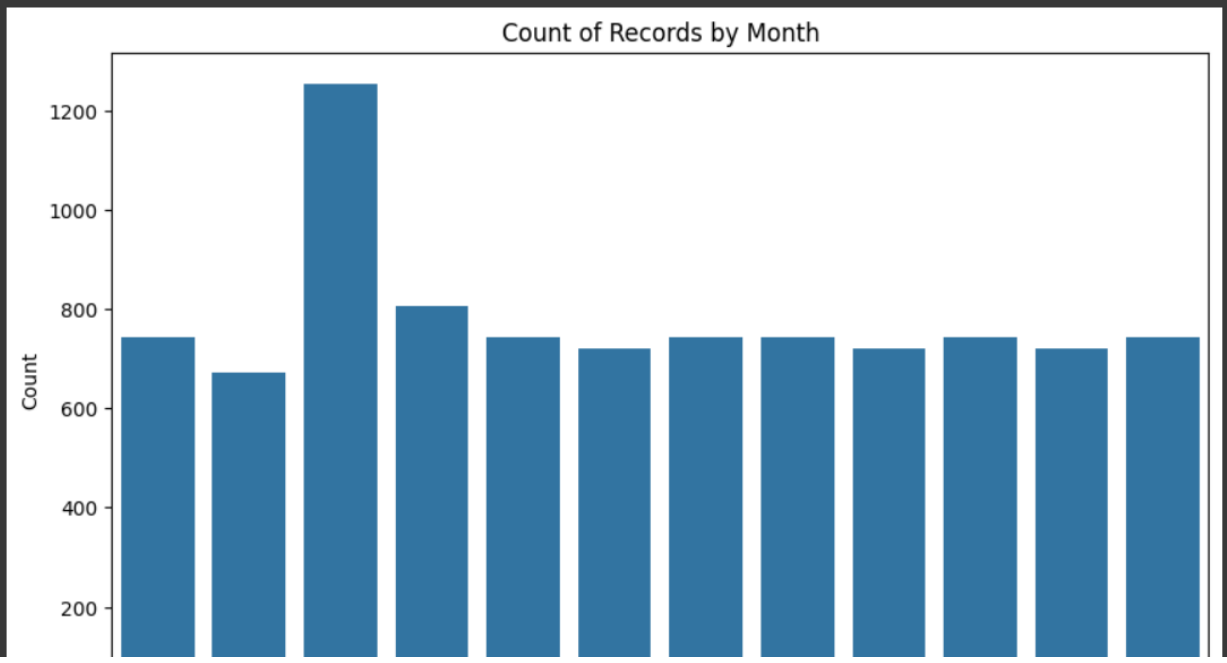

```
plt.figure(figsize=(10, 8))
correlation = df_cleaned[['CO(GT)', 'NOx(GT)', 'NO2(GT)', 'T', 'RH']].corr()
sns.heatmap(correlation, annot=True, cmap='YlGnBu')
plt.title('Heatmap of Correlations Between Pollutants and Weather')
plt.show()
```



```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

df_ml = df_cleaned[['T', 'RH', 'CO(GT)']].dropna()
X = df_ml[['T', 'RH']]
y = df_ml['CO(GT)']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
mse = mean_squared_error(y_test, y_pred)
r2 = model.score(X_test, y_test)
print("RMSE:", rmse)
print("MSE:", mse)
print("R² Score:", r2)
plt.figure(figsize=(10, 6))
plt.scatter(X_test['T'], y_test, color='blue', label='Actual')
plt.plot(X_test['T'], y_pred, color='red', label='Regression Line')
plt.xlabel('Temperature (T)')
plt.ylabel('CO(GT)')
plt.title('Linear Regression: CO(GT) vs Temperature')
plt.legend()
plt.show()
```

```python
df = pd.read_csv('Air Quality1.csv')
df_cleaned = df.dropna()
df_cleaned = df_cleaned.drop(columns=['PT08.S1(CO)', 'PT08.S2(NMHC)', 'PT08.S3(NOx)', 'PT08.S4(NO2)', 'PT08.S5(O3)', 'NMHC(GT
df_cleaned['Datetime'] = pd.to_datetime(df_cleaned['Datetime'])
df_cleaned['Month'] = df_cleaned['Datetime'].dt.month
plt.figure(figsize=(10, 6))
sns.countplot(x='Month', data=df_cleaned)
plt.xlabel('Month')
plt.ylabel('Count')
plt.title('Count of Records by Month')
plt.show()
```



5. Conclusion

The project successfully analyzed air quality data to monitor pollutants affecting agriculture. EDA revealed seasonal trends and correlations, with visualizations illustrating CO(GT) patterns. Linear Regression, Ridge, Lasso, and Random Forest models were trained to predict CO(GT), with Random

Forest showing superior performance due to its ability to capture non-linear relationships. These insights can help farmers anticipate pollution risks and protect crops.

6. Future Scope

- **Implement advanced models like Gradient Boosting or Neural Networks.**
- **Integrate IoT sensors for real-time air quality monitoring.**
- **Apply generative AI to create synthetic datasets for broader analysis.**
- **Develop a mobile application for farmers to access predictions and alerts.**

7. References

- [1] J. Brownlee, "Linear Regression for Machine Learning," Machine Learning Mastery, 2020. [Online]. Available: https://machinelearningmastery.com/linear-regression-for-machine-learning/
  [2] Scikit-learn Developers, "Scikit-learn: Machine Learning in Python," 2023. [Online]. Available: https://scikit-learn.org/stable/
  [3] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.