CROP YIELD PREDICTION

USING ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

TABLE OF CONTENT

TEAM MEMBERS

NAMES	ROLL NO .
• Ananya Sharma	2210990111
• Ajay Rana	2210990080
• Abhishek Garg	2210990047
• Abhishek Tayal	2210990052

INTRODUCTION

Predicting crop yields is a game-changer for modern agriculture. It empowers farmers and policymakers to make smarter decisions. These predictions are based on powerful computer models that analyze vast amounts of data, including weather patterns, pesticide use, and soil conditions. By understanding how these factors influence future harvests, we can optimize agricultural practices and ensure a stable food supply for everyone.

PROBLEM STATEMENT

- Accurately predicting crop yields is difficult despite advancements in technology.

- **Reasons for Difficulty:**

- Several factors significantly impact crop production:
- Pesticide usage
- Rainfall variability (unpredictable rainfall patterns)
- Temperature fluctuations

OBJECTIVE

Here are the 3 main objectives of a crop yield prediction project:

Boost Accuracy:
Predict crop yields with greater precision, considering complex interactions like weather and pests.

Optimize Decisions:
Empower farmers to make informed choices about planting, resources, and risk mitigation based on yield forecasts.

Enhance Food Security:
Contribute to a stable food supply by identifying potential shortages and surpluses across regions.

DATA PREPARATION

DATA COLLECTION

1. PESTICIDE USE
2. RAINFALL
3. TEMPERATURE
4. CROP YIELD

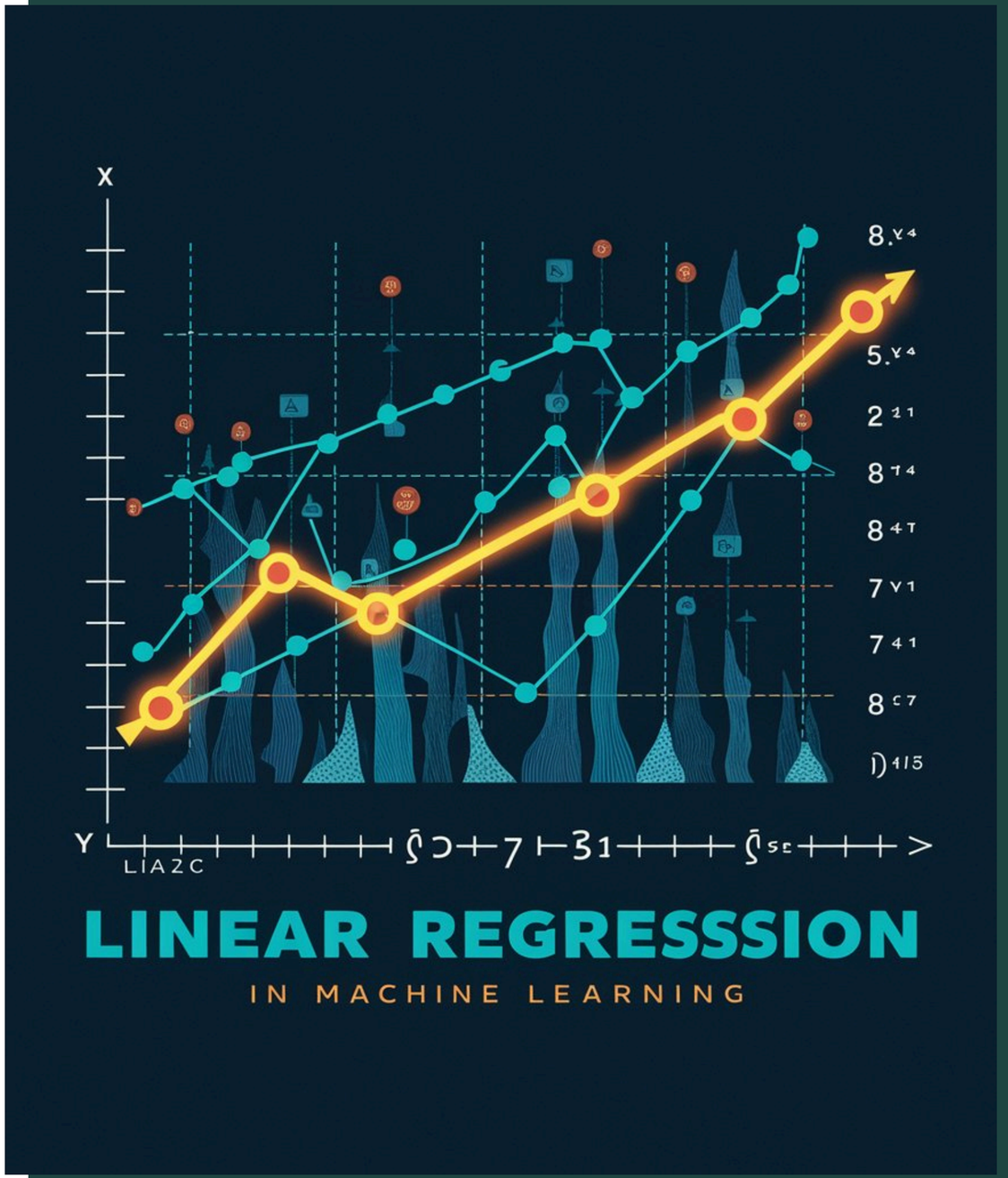
DATA PREPROCESSING

- HANDLING MISSING VALUES
- STANDARDIZED UNITS
- LABEL ENCODING

MODEL

LINEAR REGRESSION

Linear regression is a statistical method used to model the relationship between a dependent variable (crop yield) and one or more independent variables (year, area, pesticide usage, rainfall, temperature)



PROJECT OVERVIEW

IMPORTING LIBRARY

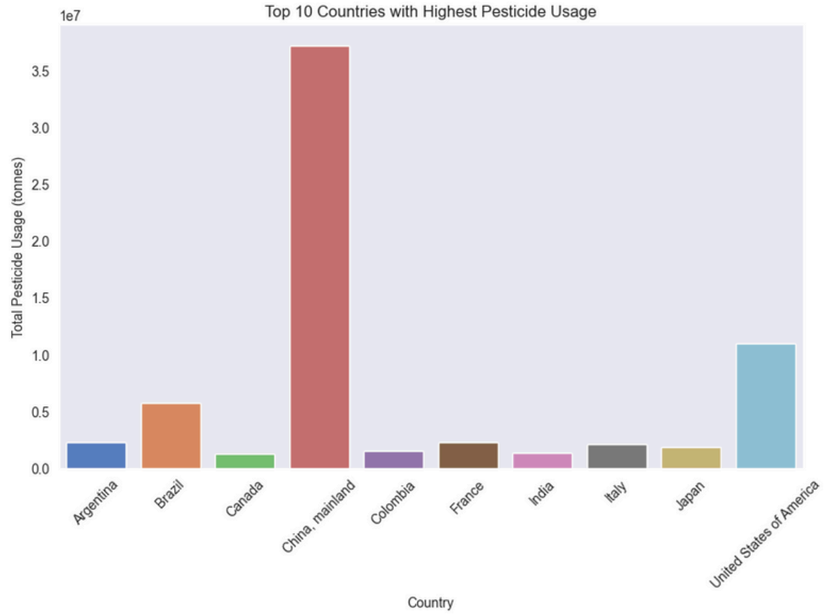
```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error
from sklearn.preprocessing import LabelEncoder
```

```python
# Loading Datasets
pesticides_df = pd.read_csv('pesticides.csv')
rainfall_df = pd.read_csv('rainfall.csv')
temp_df = pd.read_csv('temp.csv')
yield_df = pd.read_csv('yield.csv')

# Merge datasets based on 'Year' and 'Area'
merged_df = pd.merge(pesticides_df, rainfall_df, on=['Year', 'Area'], how='inner')
merged_df = pd.merge(merged_df, temp_df, left_on=['Year', 'Area'], right_on=['year', 'country'], how='inner')
merged_df = pd.merge(merged_df, yield_df, left_on=['Year', 'Area'], right_on=['Year', 'Area'], how='inner')

# Clean Data- Remove unwanted Values.
merged_df.replace('..', np.nan, inplace=True)
merged_df.dropna(inplace=True)

# Encode categorical data
label_encoder = LabelEncoder()
merged_df['Area'] = label_encoder.fit_transform(merged_df['Area'])

# Prepare features and target
X = merged_df[['Year', 'Area', 'Value_x', 'average_rain_fall_mm_per_year', 'avg_temp']]
X.columns = ['Year', 'Area', 'Pesticides', 'Rainfall', 'Temperature']
y = merged_df['Value_y']
```

LOADING DATASETS , MERGE DATASETS– BASED ON YEAR AND AREA, CLEAN DATA, ENCODE CATEGORICAL DATA, PREPARE FEATURE AND TARGET

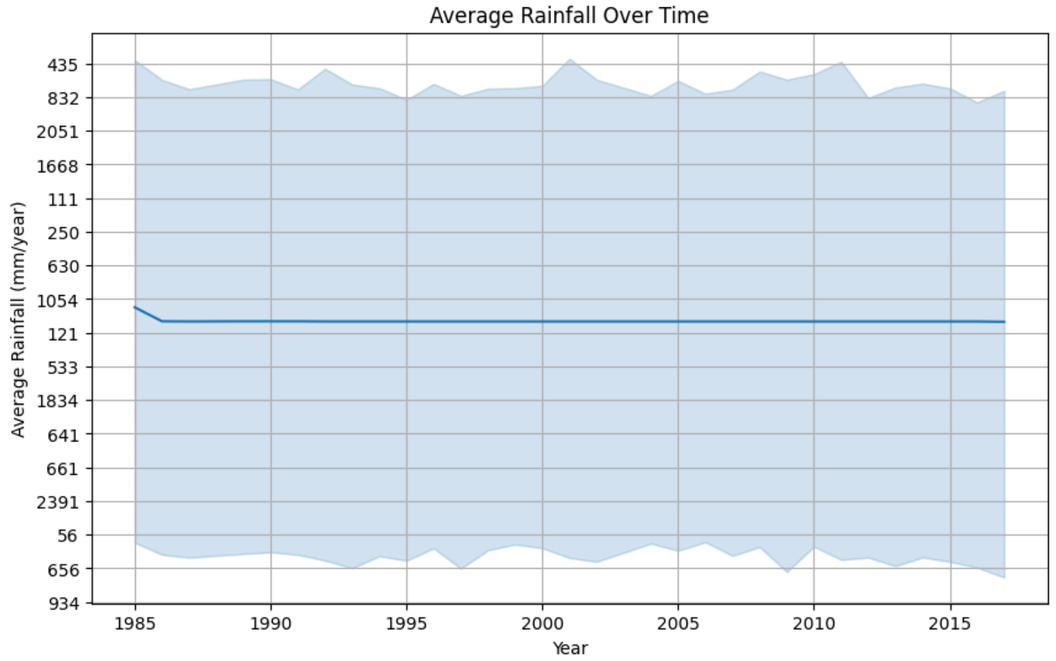
COUNTRIES WITH HIGHEST PESTICIDES USAGE

```
import matplotlib.pyplot as plt
import seaborn as sns

# Sort the pesticides data by total usage and select the top 10 countries
top_10_countries = pesticides_df.groupby('Area')['Value'].sum().nlargest(10).index

# Filter the pesticides data for the top 10 countries
pesticides_top_10 = pesticides_df[pesticides_df['Area'].isin(top_10_countries)]

plt.figure(figsize=(10, 6))

sns.barplot(data=pesticides_top_10, x='Area', y='Value', estimator=sum, ci=None, palette='muted')
plt.title('Top 10 Countries with Highest Pesticide Usage')
plt.xlabel('Country')
plt.ylabel('Total Pesticide Usage (tonnes)')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.show()
```



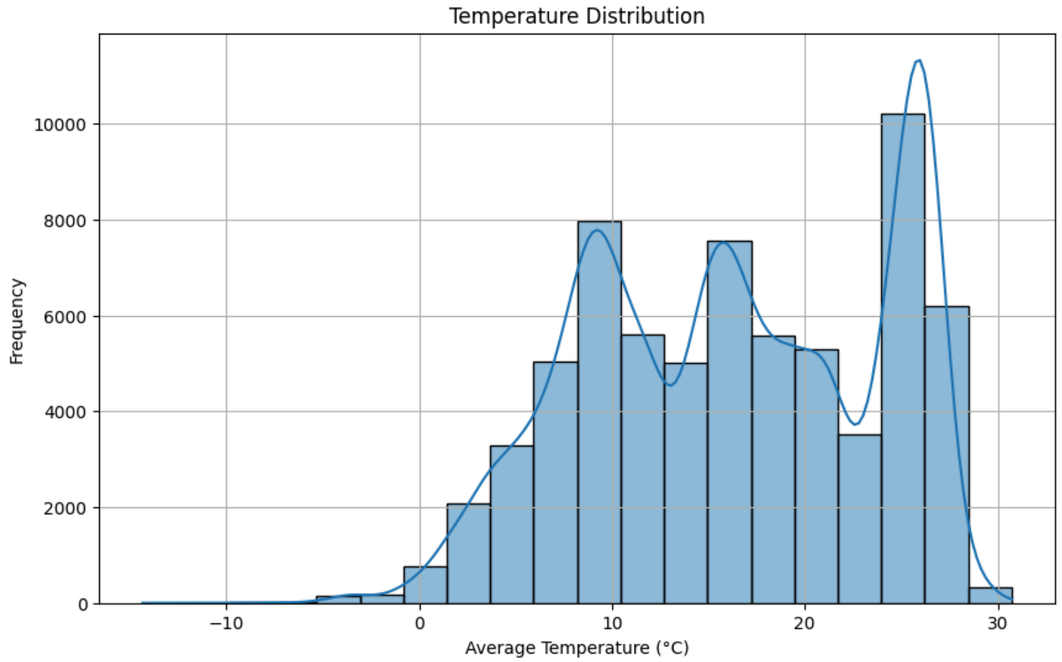
AVERAGE RAINFALL OVER TIME

```
import seaborn as sns

plt.figure(figsize=(10, 6))
sns.lineplot(data=rainfall_df, x='Year', y='average_rain_fall_mm_per_year')
plt.title('Average Rainfall Over Time')
plt.xlabel('Year')
plt.ylabel('Average Rainfall (mm/year)')
plt.grid(True)
plt.show()
```



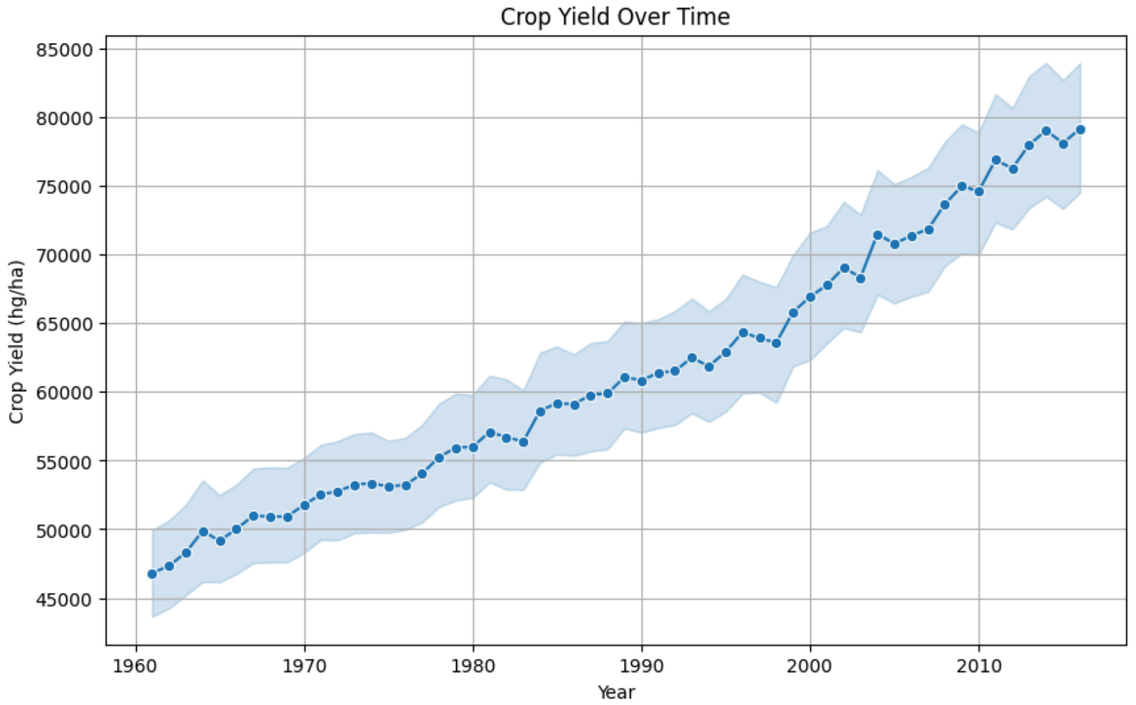
TEMPERATURE DISTRIBUTION

```
plt.figure(figsize=(10, 6))
sns.histplot(temp_df['avg_temp'].dropna(), bins=20, kde=True)
plt.title('Temperature Distribution')
plt.xlabel('Average Temperature (°C)')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```



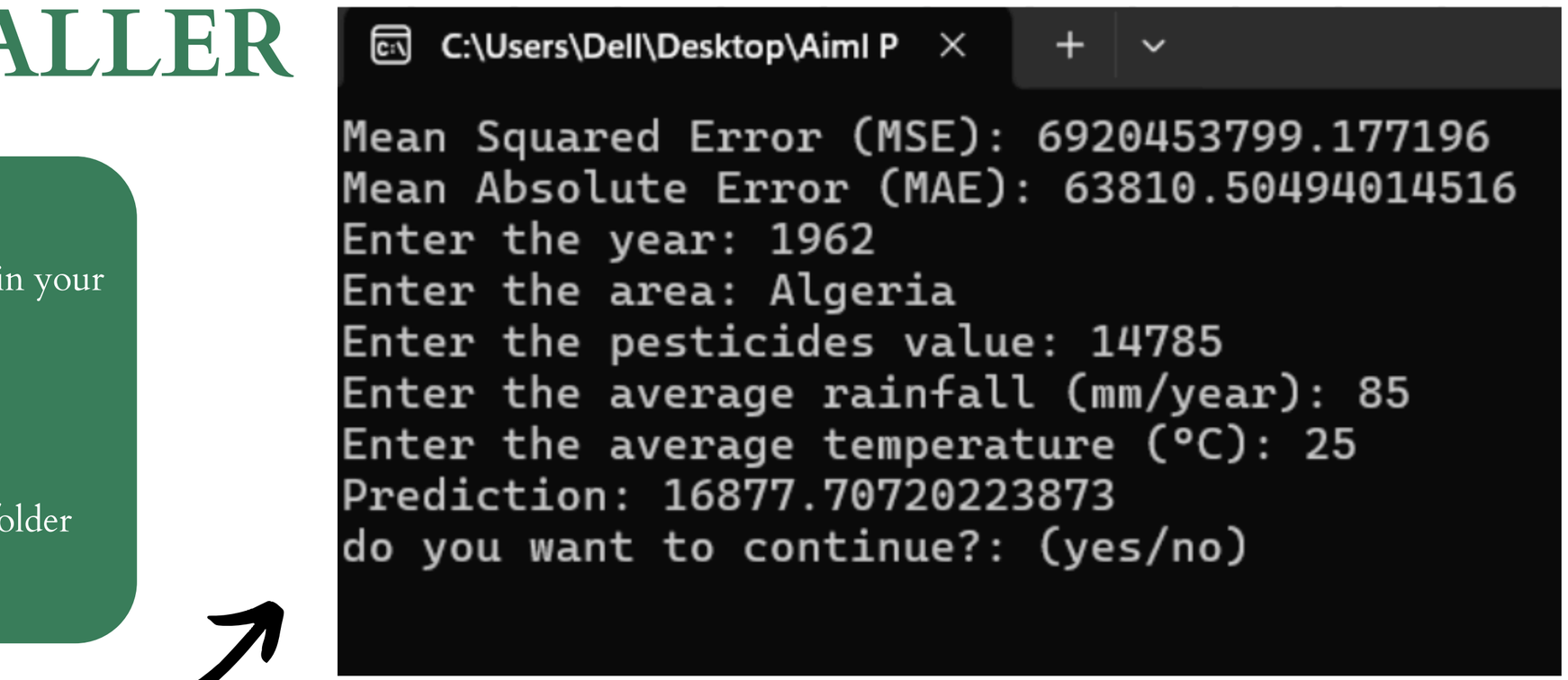
CROP YIELD OVER TIME

```
plt.figure(figsize=(10, 6))
sns.lineplot(data=yield_df, x='Year', y='Value', marker='o')
plt.title('Crop Yield Over Time')
plt.xlabel('Year')
plt.ylabel('Crop Yield (hg/ha)')
plt.grid(True)
plt.show()
```



STEPS FOR .exe File USING PYINSTALLER

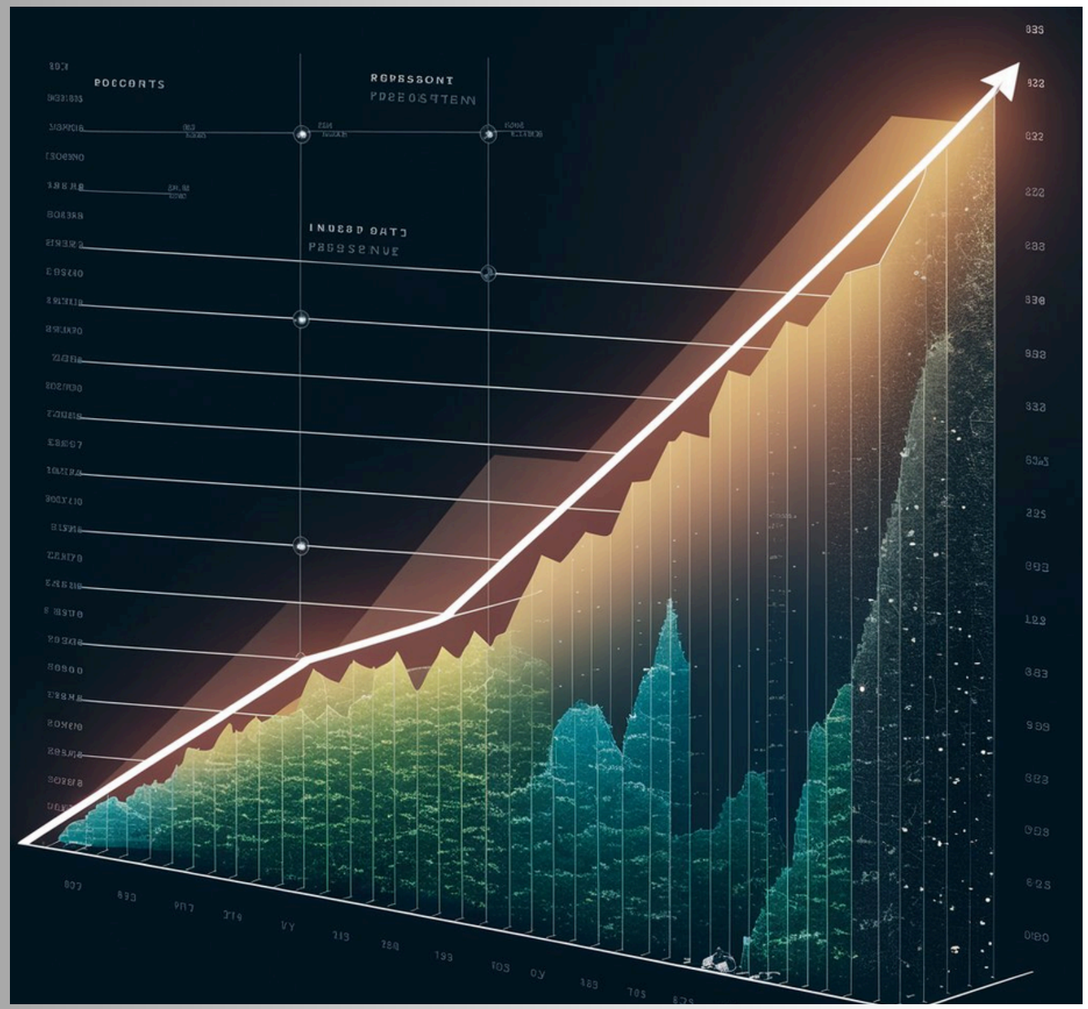
1.Convert the .ipynb file to .py file
2.install pyinstaller -{pip install pyinstaller} in your cmd prompt
3.move terminal to file location
4.dist folder is created at the same location
5.pyinstaller project.py
6.locate project.py file is present in the dist folder
7.on doubleclick run--->



convert a Python script (.py file) related to crop yield prediction into an executable (.exe file), you're essentially creating a standalone application that can run on a computer without requiring Python to be installed.

FUTURE SCOPE

Further research could involve incorporating additional variables like soil nutrients and crop varieties. Advanced modeling techniques such as Random Forests or Gradient Boosting could improve prediction accuracy, while real-time monitoring and stakeholder engagement could enhance the model's applicability in agricultural decision-making.



RESULT

The analysis utilizing linear regression revealed significant correlations between pesticide usage, rainfall, temperature, and crop yield. The model accurately predicted crop yields based on these factors, providing valuable insights for optimizing agricultural practices, resource allocation, and ensuring food security in the face of changing environmental conditions.


Thank You