Eco shield
Save Planet

ECOSHIELD

SAVE PLANET

PREPARED BY
MSG-DATA DOER

PREPARED FOR
TECHWIZ 5

Acknowledgment

We, the team **MSG - Data Doer**, would like to express our sincere gratitude to all those who contributed to the successful completion of our project, **ECOSHIELD**, in the category of Data Science for **Techwiz 5**.

First and foremost, we are deeply grateful to **Aptech** for organizing such an incredible platform that fosters creativity and innovation among students. The opportunity to participate in this global competition has been invaluable in helping us hone our skills in data science.

We would like to extend special thanks to our mentor, **Sir Daniyal Ghani**, for his unwavering guidance and support throughout the project. His expertise and insights were crucial in shaping our approach and overcoming the challenges we faced.

Our heartfelt thanks also go out to our **teammates**, whose dedication and hard work made this project possible. Working together as **MSG - Data Doer** has been a rewarding experience, and each member's unique contributions added immense value to our efforts.

Lastly, we are thankful to our families and friends for their constant encouragement and belief in us. Your support gave us the strength to persevere and complete this project successfully.

Team

Name	Enrollment
Muhammad Ahsan	Student1397641
Muhammad Umer	Student1398931
Hashir Ahmed Khan	Student1395302
Nimra Atta	Student1388812
Aleena Syed	Student1392363

Contents

Chapter 1: Introduction

1.1 Motivation

The **EcoShield** project is inspired by the pressing need to address the growing impact of climate change on global food production. As climate patterns continue to shift, the agricultural sector faces severe challenges, including unpredictable weather, rising temperatures, and extreme events like droughts and floods, which threaten food security worldwide. These challenges make it crucial to develop innovative solutions to protect the agricultural supply chain.

EcoShield leverages the power of **Big Data** and **Data Science** to tackle these issues. By collecting and analyzing vast amounts of climate and agricultural data, the project utilizes advanced machine learning algorithms such as **Random Forest**, **Support Vector Machine**, and **Naive Bayes** to build predictive models that estimate future crop yields under various climate scenarios. These models provide valuable insights into how farmers and policymakers can adapt agricultural practices to changing conditions.

The motivation behind **EcoShield** is to deliver actionable intelligence that enables stakeholders to anticipate climate-driven disruptions, optimize farming techniques, and contribute to global food security. By applying cutting-edge technologies, **EcoShield** aims to reduce the negative impacts of climate change on agriculture and foster a more resilient and sustainable food production system.

1.2 Problem Statement

Understanding the impact of climatic patterns on the global food supply chain is critical for addressing sustainability and food security challenges. Climate change has introduced a range of significant disruptions to food systems worldwide, threatening both the production and availability of food. These challenges are exacerbated by factors such as growing global populations, land degradation, and the loss of cropland to urbanization.

As climate change intensifies, particularly in regions vulnerable to drought and famine, the agricultural sector is expected to face increasing pressures. To mitigate these effects, there is a growing interest in harnessing **Big Data** and **Data Science** to analyze vast datasets, including temperature variations and food security concerns. However, developing accurate models that detect climatic changes and predict their impact on agriculture remains a complex challenge.

This project aims to address the critical problem of climate change's impact on global food production by leveraging **Big Data** and **Data Science** techniques. The proposed solution involves developing an application that can process and analyze climate-related data, providing actionable insights to mitigate the effects of climate change on food systems and contribute to long-term agricultural sustainability.

Chapter 2: Data Collection

2.1 Introduction

There are two types of data one is primary and the other, secondary. Primary data refers to the information that is directly collected from the source. The primary data is considered as the best data in research world. Primary data can be collected by the following methods:

- Interviews
- Observations
- Surveys and questionnaires
- Focus groups
- Oral histories

Secondary data refers to the data which already exists and can be used to analyze the information. Primary data refers to the authentic source of information but on the other hand, various situations require different material where secondary data can be of great value for particular purposes. Secondary data can be collected through different methods like:

- Government archives
- Internet
- Libraries

The data used in this research is the secondary data. The method for data collection involves internet and government archives.

2.2 Data Information

The datasets used in this thesis were gathered from multiple credible sources, reflecting a wide range of climate-related and agricultural factors. These datasets encompass various domains, such as pesticide use, rainfall patterns, crop yields, and temperature trends across different countries and years.

1. **Pesticide Use**: This dataset records pesticide usage in different area from different years, with quantities measured in tonnes of active ingredients. It provides consistent annual records showing the fluctuations in pesticide use over time.
2. **Rainfall Patterns**: The dataset includes the average annual rainfall (in mm) for Afghanistan from 1985 to 1990. These consistent yearly records highlight the stable rainfall patterns during this period, a crucial factor for agricultural productivity.

3. **Crop Yields**: This dataset focuses on maize yield (measured in hectograms per hectare), capturing the fluctuations in crop productivity due to varying external factors, such as climatic conditions.
4. **Temperature Trends**: The average annual temperature dataset for Côte D'Ivoire covers the years from 1849 to 1853. Although there are some missing values, it provides an important historical view of temperature trends, which is essential for analyzing long-term climate impacts.

2.3.1 Data Preprocessing and Integration

Once collected, the datasets underwent rigorous preprocessing to ensure consistency and reliability. The steps included:

- **Handling Missing Data**: Missing values, particularly in the temperature dataset, were addressed either by interpolation or careful exclusion to maintain the integrity of the analysis.
- **Normalization and Standardization**: To align data from different sources, various units of measurement (e.g., tonnes, mm, and hectograms per hectare) were standardized.
- **Data Cleaning**: Inconsistent entries, such as discrepancies in country names or measurement units, were corrected to ensure uniformity. For example, variations in country names like "Côte D'Ivoire" were standardized.
- **Equal-Width Binning for Classification:** To enhance classification and understanding, continuous data was categorized into low, medium, and high ranges using an equal-width binning process. Specifically, values ranging from 50 to 167,061 were classified as **Low**, those between 167,176 and 334,130 as **Medium**, and values from 335,302 to 501,412 as **High**. This segmentation helped in organizing the data for better interpretation and clearer insights.

These preprocessing steps were essential to ensuring the dataset's accuracy and enhancing the overall analysis.

After the data was cleaned and preprocessed, the datasets were **combined into a unified file** that allowed for comprehensive analysis. By integrating climate variables such as temperature, rainfall, and crop yield data, the combined dataset enabled the development of predictive models used in the EcoShield project. This unified dataset provided a holistic view of the relationship between climate change and agricultural productivity, facilitating deeper insights for the analysis. The processed data is also added in our model to enhance the future model accuracy and results.

2.3 Data Stats

The data used for analysis in this document consists of 28,242 rows, sourced from a comprehensive dataset providing insights into various aspects under study. After cleaning and preprocessing, six incomplete or irrelevant records were removed, resulting in a final dataset of 28,236 rows, ready for thorough analysis. This data refinement ensured greater accuracy and consistency for the analysis process.

Figure 2.3.1 shows the data of total count of areas. The maximum record was found from India.

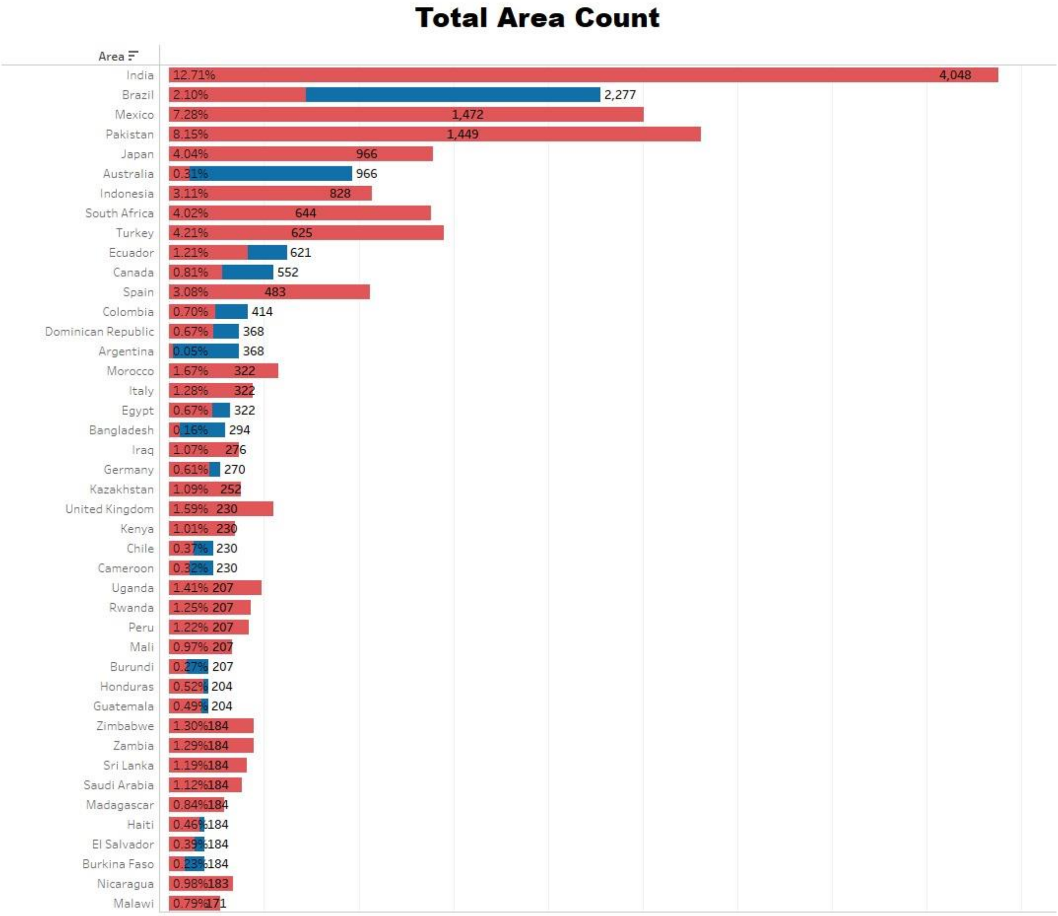


Figure 2.3.1 Bar graph representing total counts of area

Figure 2.3.2 representing the data of crops in total that includes Maize, Wheat, Rice, and multiple crops.

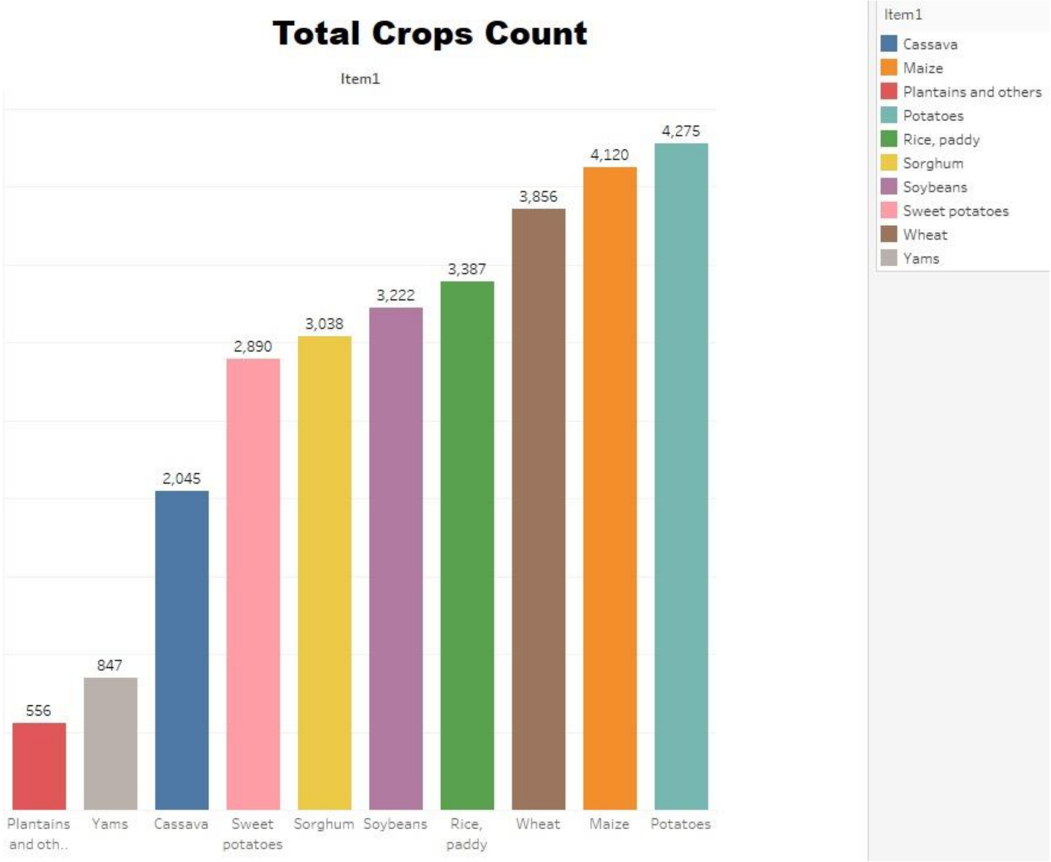


Figure 2.3.2 Bar graph representing crop wise total data

Figure 2.3.3 shows the count and percentage category wise. By analyzing data, it is estimated that major values are related to **Low** yield category.

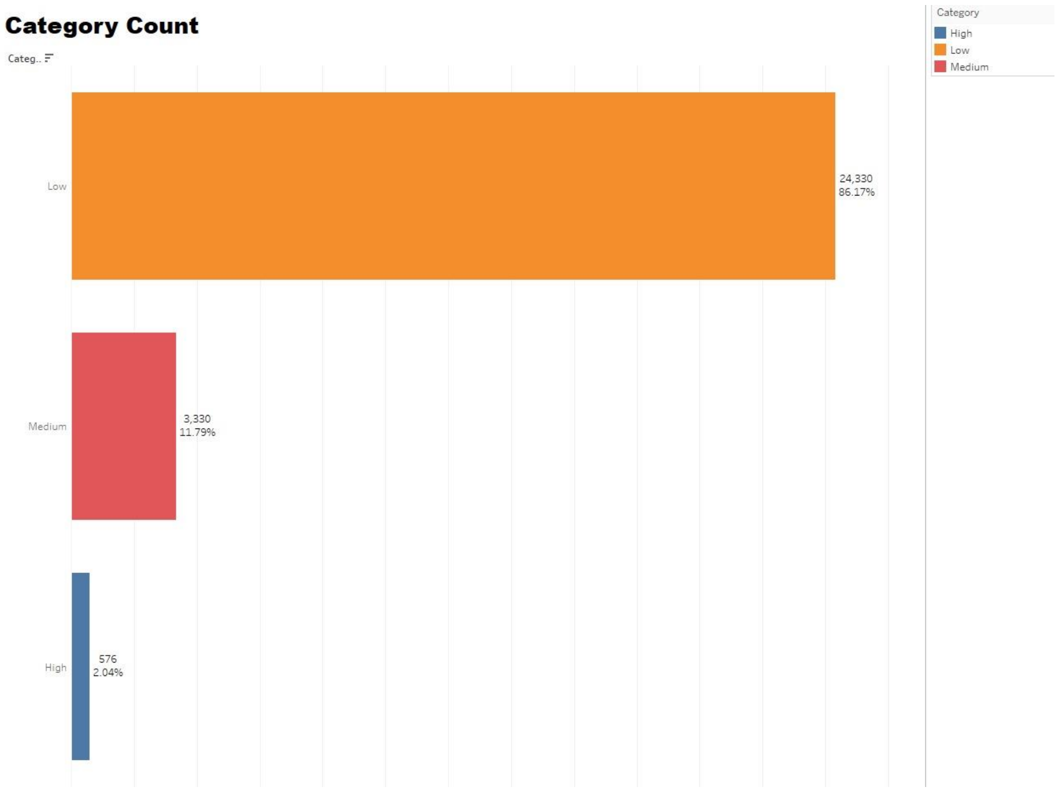
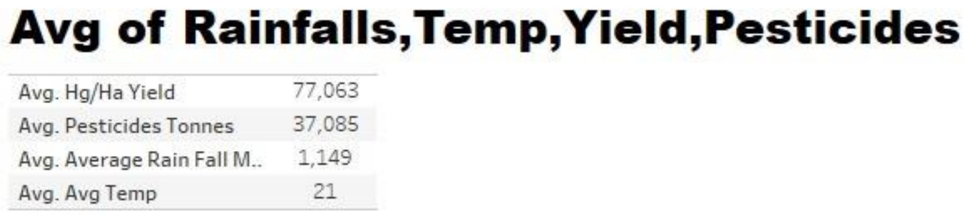


Figure 2.3.3 Bar graph shows total and percentage of categories

Figure 2.3.4 shows the average data of rainfall, yield, temperature and pesticides.



Avg of Rainfalls,Temp,Yield,Pesticides

Avg. Hg/Ha Yield	77,063
Avg. Pesticides Tonnes	37,085
Avg. Average Rain Fall M..	1,149
Avg. Avg Temp	21

Figure 2.3.4 Figure shows average of rainfall, pesticides, temperature, and yield

Figure 2.3.5 Illustrates the data b/w area and average pesticides.

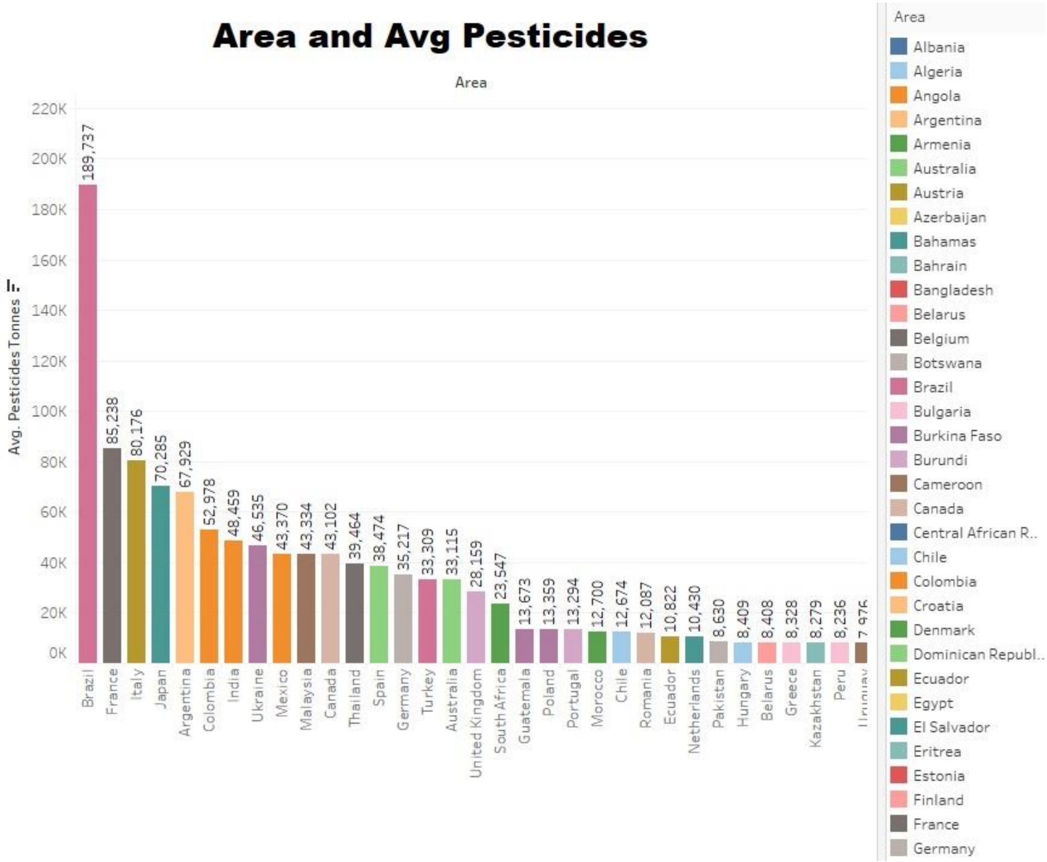


Figure 2.3.5 Bar graph representing area and average pesticides

Figure 2.3.6 Illustrates the data of average rainfall.

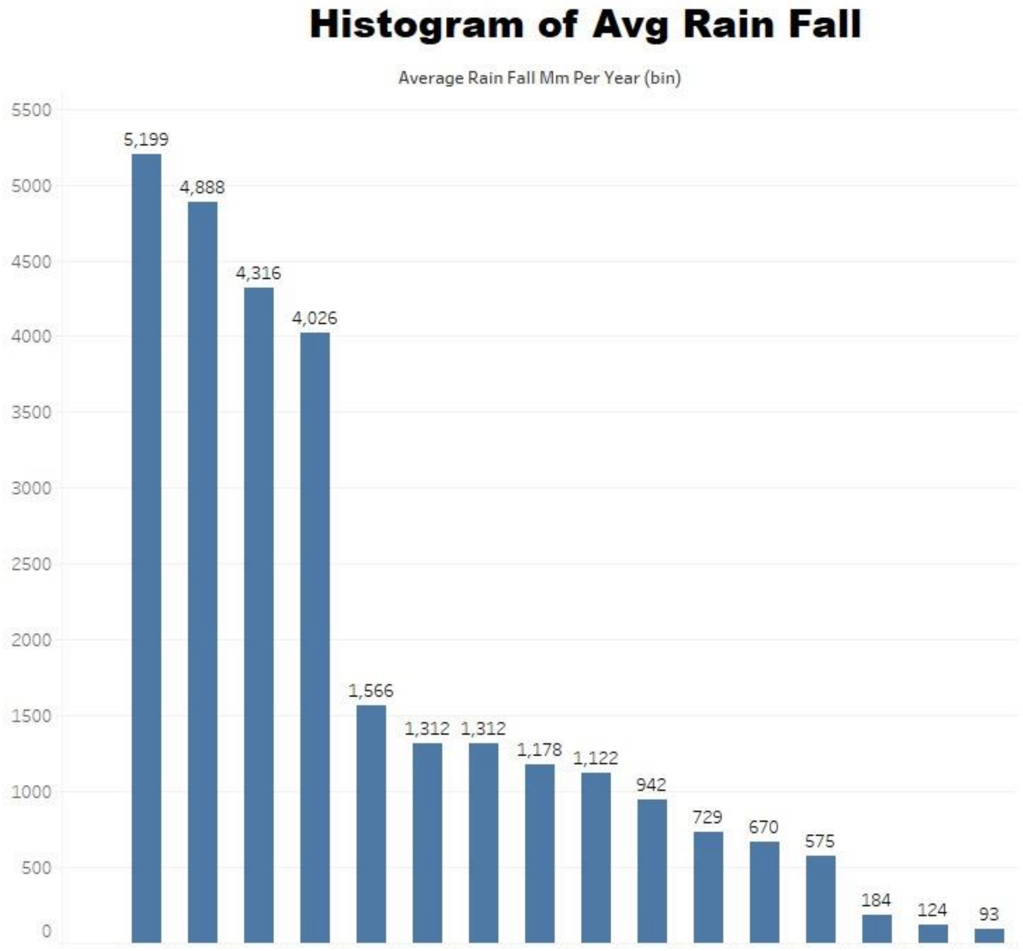


Figure 2.3.6 Bra graph of average rainfall

Chapter 3: Data Science in Agriculture

3.1 Introduction to Data Science

Data Science is a multidisciplinary field that focuses on extracting meaningful insights from vast amounts of data using statistical techniques, machine learning algorithms, and advanced data processing methods. In the context of agriculture, it plays a crucial role in addressing the growing challenges posed by climate change, as it allows stakeholders to make data-driven decisions. By leveraging large datasets related to climate, soil conditions, crop yields, and agricultural inputs, Data Science helps identify patterns, predict future trends, and optimize farming practices.

3.2 Role of Data Science in the EcoShield Project

The EcoShield project employs a combination of machine learning algorithms, data visualization, and predictive analytics to model climate-agriculture interactions. Data Science techniques help in cleaning, organizing, and processing complex data from diverse sources, allowing for the creation of robust predictive models that can inform food security strategies.

Chapter 4: Data Mining Models for EcoShield

4.1 Introduction to Data Mining

Data mining refers to the process of extracting useful information from large datasets by identifying patterns, trends, and relationships. In agriculture, data mining can uncover insights into how different factors—such as weather patterns, soil quality, and crop types—affect food production. For the EcoShield project, data mining is essential in processing large volumes of agricultural and climate data, allowing the discovery of key factors influencing crop yields under varying climate conditions.

4.2 Types of Data Mining Models

Data mining models are categorized into two primary types:

4.2.1 Predictive Model

Predictive models are used to predict values from the results of various data. The predictive modeling is based on the results and outcomes of the historical data. Predictive model data mining tasks are performed using different techniques that are listed below:

- Regression
- Time series analysis
- Classification
- Prediction

4.2.2 Descriptive Models

The descriptive model is used to distinguish the relationships and patterns in data. It does not work on historical data like a predictive model rather it focuses on key events and factors affecting them.

- Regression
- Naïve Bayes
- Decision Tree
- Artificial Neural Network

4.3 Model Used in the EcoShield Project

For the purposes of the EcoShield project, the **classification technique** is primarily used as part of the predictive data mining model. The goal is to categorize crop yields into various risk levels (e.g., low, medium, high) based on climatic factors. This enables the prediction of which regions or crops are more likely to suffer due to adverse climate conditions.

4.3.1 Classification

Classification is a supervised learning technique that assigns data into predefined classes or categories based on training and testing data. For EcoShield, this involves training models with historical crop yield and climate data to predict future yield categories. This method is crucial for forecasting the impact of variables like temperature and rainfall on agricultural productivity.

Key classification techniques used in the project include:

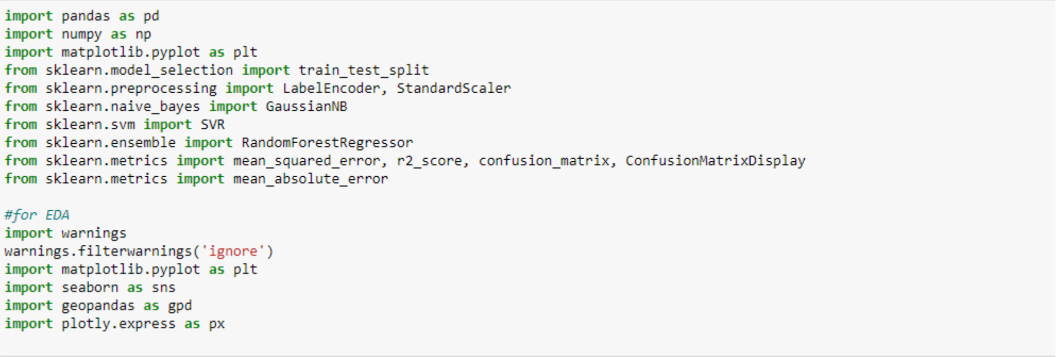
- **Naive Bayes (GaussianNB):** A probabilistic classifier designed for continuous data, assuming that the features follow a Gaussian (normal) distribution.
- **Random Forest (RandomForestRegressor):** An ensemble method used for regression tasks on continuous data, combining multiple decision trees to enhance prediction accuracy and generalization.
- **Support Vector Machine (SVR):** A regression algorithm tailored for continuous data, which finds the optimal hyperplane to predict values while minimizing error.

Chapter 5: Code Implementation and Results

5.1 Code Execution and Algorithm Details

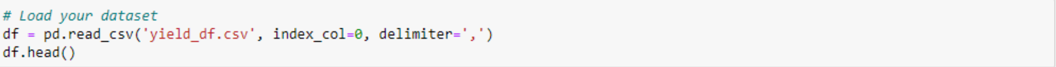
This section contains the code snippets used for implementing the algorithms and methods applied in the analysis.

5.1.1 Importing Libraries



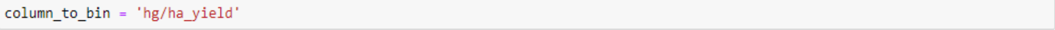
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVR
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score, confusion_matrix, ConfusionMatrixDisplay
from sklearn.metrics import mean_absolute_error

#for EDA
import warnings
warnings.filterwarnings('ignore')
import matplotlib.pyplot as plt
import seaborn as sns
import geopandas as gpd
import plotly.express as px
```

5.1.2 Loading Data Set

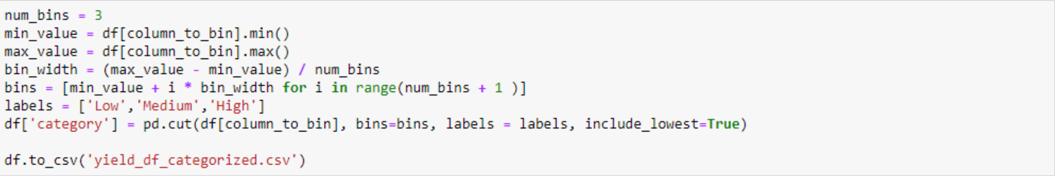


```
# Load your dataset
df = pd.read_csv('yield_df.csv', index_col=0, delimiter=',')
df.head()
```

5.1.3 Creating Bins for Classification

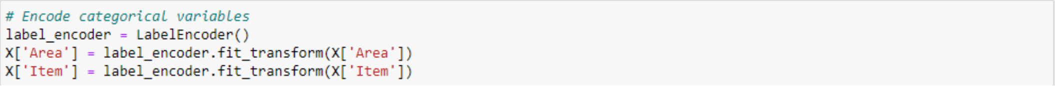


```
column_to_bin = 'hg/ha_yield'
```



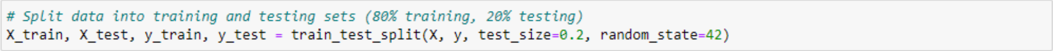
```
num_bins = 3
min_value = df[column_to_bin].min()
max_value = df[column_to_bin].max()
bin_width = (max_value - min_value) / num_bins
bins = [min_value + i * bin_width for i in range(num_bins + 1 )]
labels = ['Low','Medium','High']
df['category'] = pd.cut(df[column_to_bin], bins=bins, labels = labels, include_lowest=True)

df.to_csv('yield_df_categorized.csv')
```

5.1.4 Encoding Categorical Variables



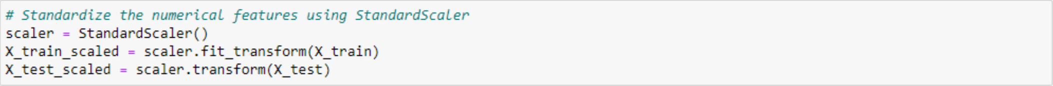
```
# Encode categorical variables
label_encoder = LabelEncoder()
X['Area'] = label_encoder.fit_transform(X['Area'])
X['Item'] = label_encoder.fit_transform(X['Item'])
```

5.1.5 Splitting Data into Training and Testing



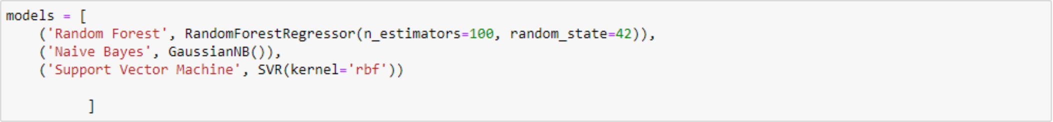
```
# Split data into training and testing sets (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

5.1.6 Standardizing using Standard Scaler Method



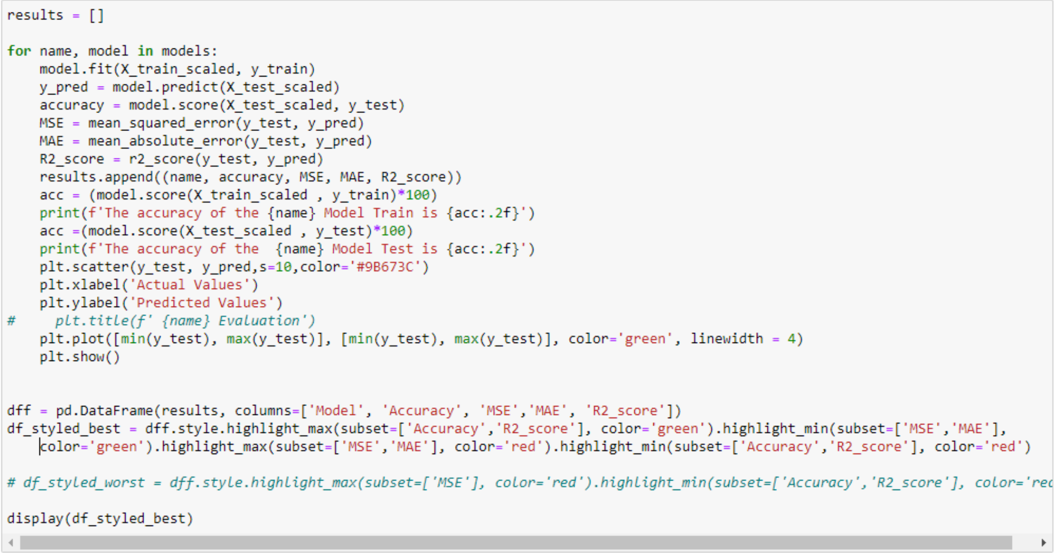
```
# Standardize the numerical features using StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

5.1.7 Defining Models Array



```python
models = [
    ('Random Forest', RandomForestRegressor(n_estimators=100, random_state=42)),
    ('Naive Bayes', GaussianNB()),
    ('Support Vector Machine', SVR(kernel='rbf'))

        ]
```

5.1.8 Model Fitting, Accuracy Measurement, Prediction



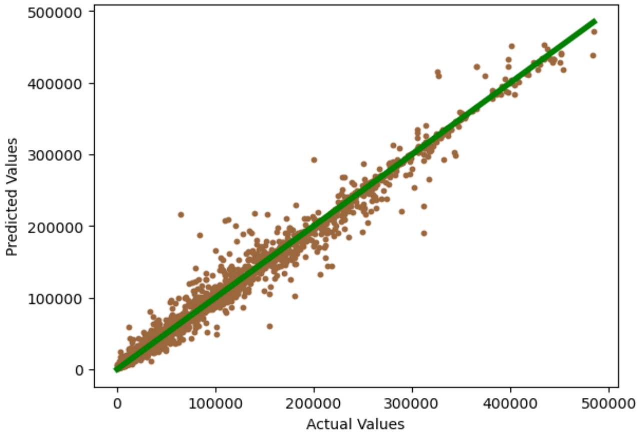
```python
results = []

for name, model in models:
    model.fit(X_train_scaled, y_train)
    y_pred = model.predict(X_test_scaled)
    accuracy = model.score(X_test_scaled, y_test)
    MSE = mean_squared_error(y_test, y_pred)
    MAE = mean_absolute_error(y_test, y_pred)
    R2_score = r2_score(y_test, y_pred)
    results.append((name, accuracy, MSE, MAE, R2_score))
    acc = (model.score(X_train_scaled , y_train)*100)
    print(f'The accuracy of the {name} Model Train is {acc:.2f}')
    acc =(model.score(X_test_scaled , y_test)*100)
    print(f'The accuracy of the  {name} Model Test is {acc:.2f}')
    plt.scatter(y_test, y_pred,s=10,color='#9B673C')
    plt.xlabel('Actual Values')
    plt.ylabel('Predicted Values')
#     plt.title(f' {name} Evaluation')
    plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='green', linewidth = 4)
    plt.show()


dff = pd.DataFrame(results, columns=['Model', 'Accuracy', 'MSE','MAE', 'R2_score'])
df_styled_best = dff.style.highlight_max(subset=['Accuracy','R2_score'], color='green').highlight_min(subset=['MSE','MAE'],
    color='green').highlight_max(subset=['MSE','MAE'], color='red').highlight_min(subset=['Accuracy','R2_score'], color='red')

# df_styled_worst = dff.style.highlight_max(subset=['MSE'], color='red').highlight_min(subset=['Accuracy','R2_score'], color='re

display(df_styled_best)
```

5.2 Results and Output Analysis

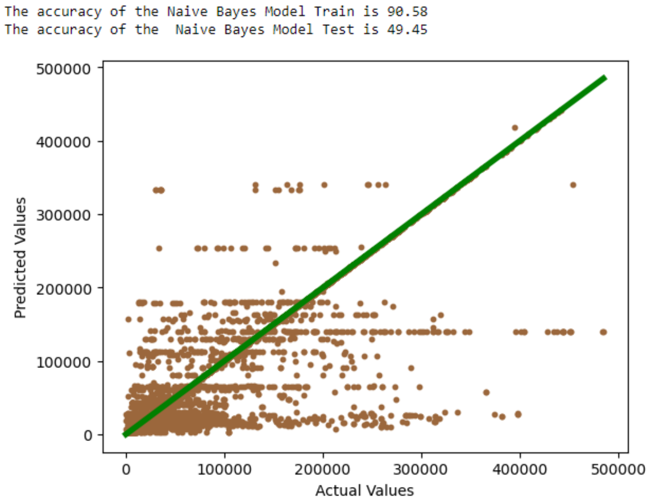
Here, the results generated from the code execution are presented, along with their corresponding analysis and interpretation.

5.2.1 Accuracy of Random Forest

```
The accuracy of the Random Forest Model Train is 99.81
The accuracy of the  Random Forest Model Test is 98.81
```

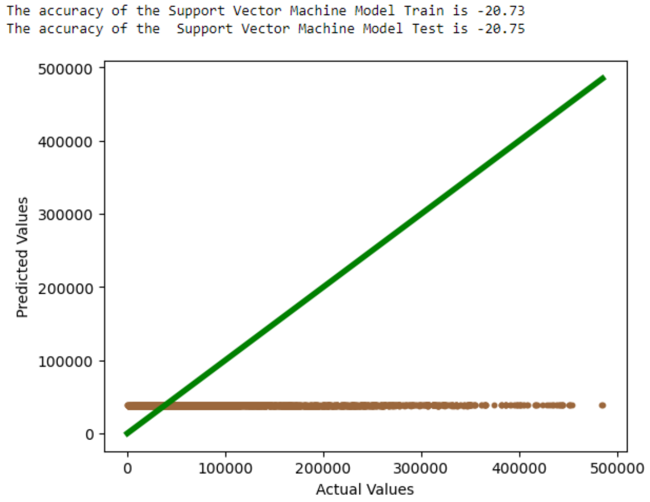


5.2.2 Accuracy of Naïve Bayes



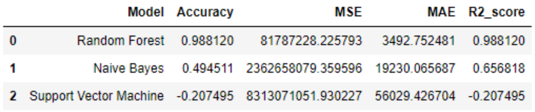
```
The accuracy of the Naive Bayes Model Train is 90.58
The accuracy of the  Naive Bayes Model Test is 49.45
```

5.2.3 Accuracy of Support Vector Machine (SVM)



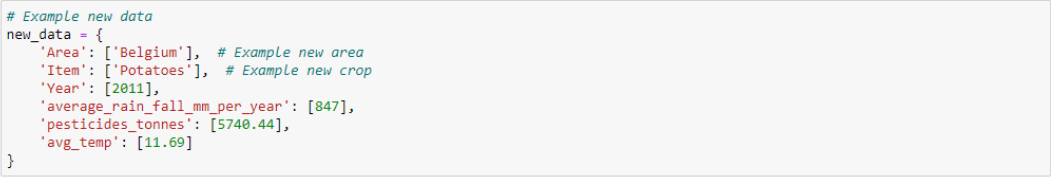
```
The accuracy of the Support Vector Machine Model Train is -20.73
The accuracy of the  Support Vector Machine Model Test is -20.75
```

5.2.4 Accuracy Table



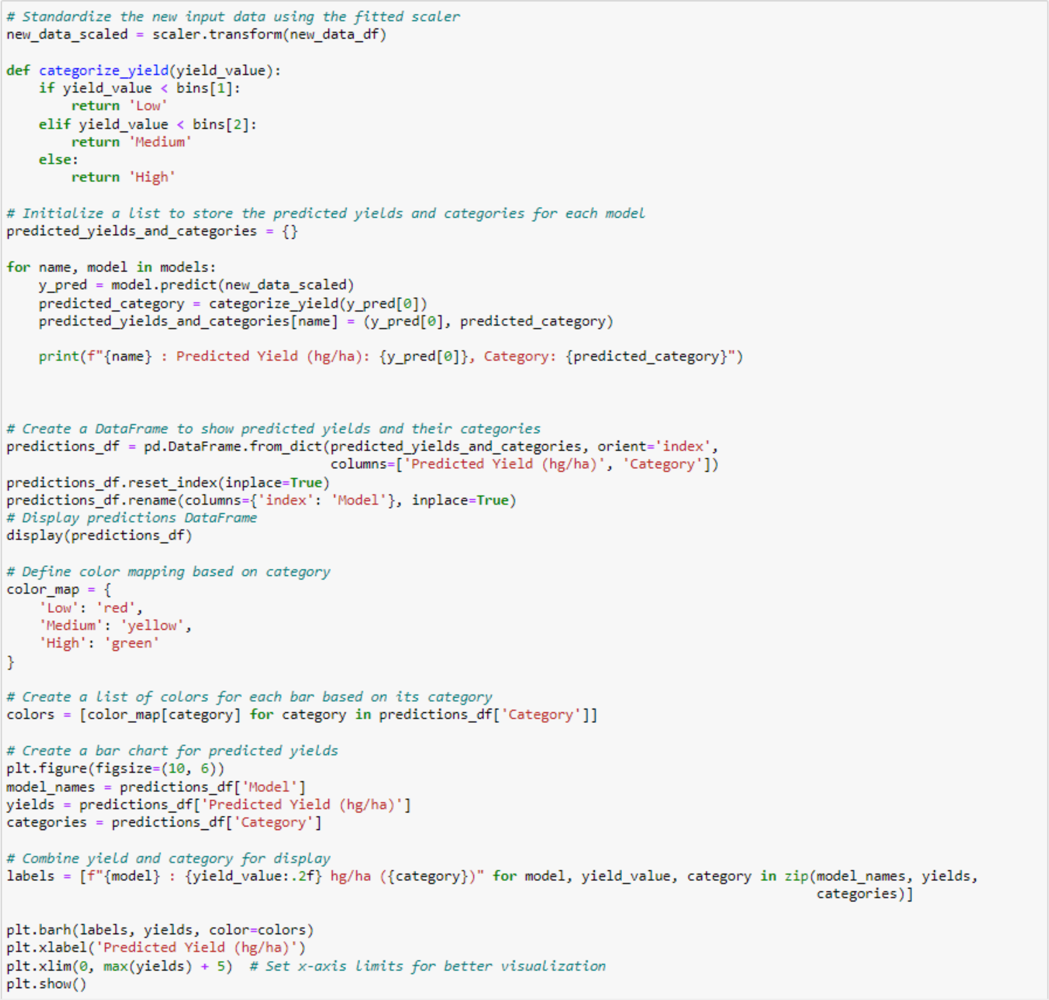
	Model	Accuracy	MSE	MAE	R2_score
0	Random Forest	0.988120	81787228.225793	3492.752481	0.988120
1	Naive Bayes	0.494511	2362658079.359596	19230.065687	0.656818
2	Support Vector Machine	-0.207495	8313071051.930227	56029.426704	-0.207495

5.2.5 Sample Data



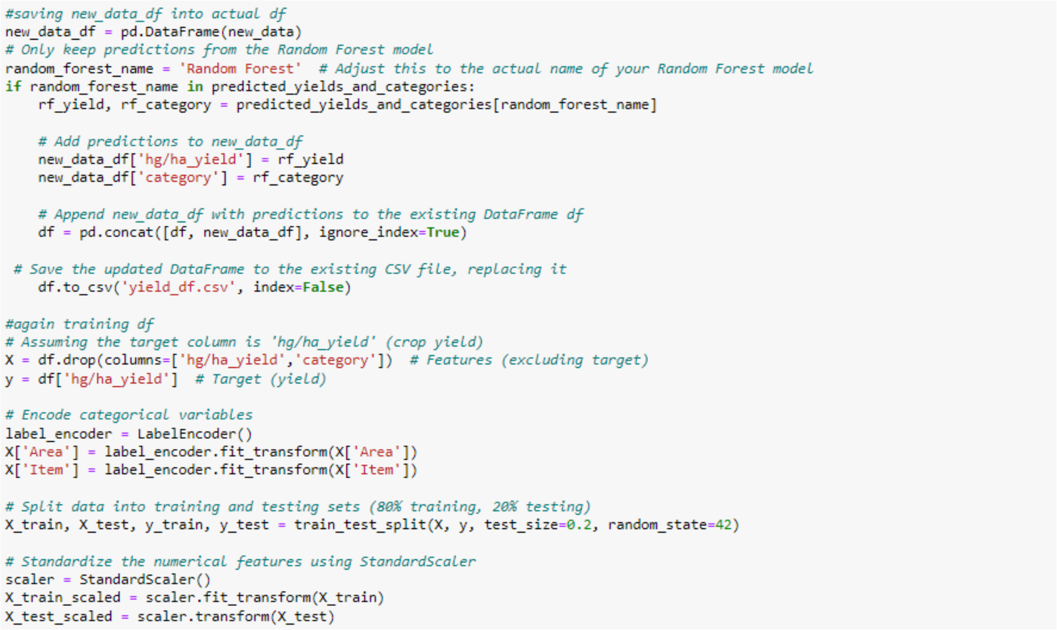
```
# Example new data
new_data = {
    'Area': ['Belgium'],  # Example new area
    'Item': ['Potatoes'],  # Example new crop
    'Year': [2011],
    'average_rain_fall_mm_per_year': [847],
    'pesticides_tonnes': [5740.44],
    'avg_temp': [11.69]
}
```

5.2.6 Standardizing Inputs for Prediction



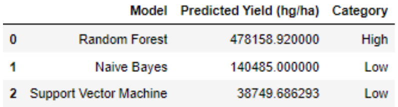
```
# Standardize the new input data using the fitted scaler
new_data_scaled = scaler.transform(new_data_df)

def categorize_yield(yield_value):
    if yield_value < bins[1]:
        return 'Low'
    elif yield_value < bins[2]:
        return 'Medium'
    else:
        return 'High'

# Initialize a list to store the predicted yields and categories for each model
predicted_yields_and_categories = {}

for name, model in models:
    y_pred = model.predict(new_data_scaled)
    predicted_category = categorize_yield(y_pred[0])
    predicted_yields_and_categories[name] = (y_pred[0], predicted_category)

    print(f"{name} : Predicted Yield (hg/ha): {y_pred[0]}, Category: {predicted_category}")



# Create a DataFrame to show predicted yields and their categories
predictions_df = pd.DataFrame.from_dict(predicted_yields_and_categories, orient='index',
                                        columns=['Predicted Yield (hg/ha)', 'Category'])
predictions_df.reset_index(inplace=True)
predictions_df.rename(columns={'index': 'Model'}, inplace=True)
# Display predictions DataFrame
display(predictions_df)

# Define color mapping based on category
color_map = {
    'Low': 'red',
    'Medium': 'yellow',
    'High': 'green'
}

# Create a list of colors for each bar based on its category
colors = [color_map[category] for category in predictions_df['Category']]

# Create a bar chart for predicted yields
plt.figure(figsize=(10, 6))
model_names = predictions_df['Model']
yields = predictions_df['Predicted Yield (hg/ha)']
categories = predictions_df['Category']

# Combine yield and category for display
labels = [f"{model} : {yield_value:.2f} hg/ha ({category})" for model, yield_value, category in zip(model_names, yields,
                                                                                                    categories)]

plt.barh(labels, yields, color=colors)
plt.xlabel('Predicted Yield (hg/ha)')
plt.xlim(0, max(yields) + 5)  # Set x-axis limits for better visualization
plt.show()
```

5.2.7 Showcasing Final Result

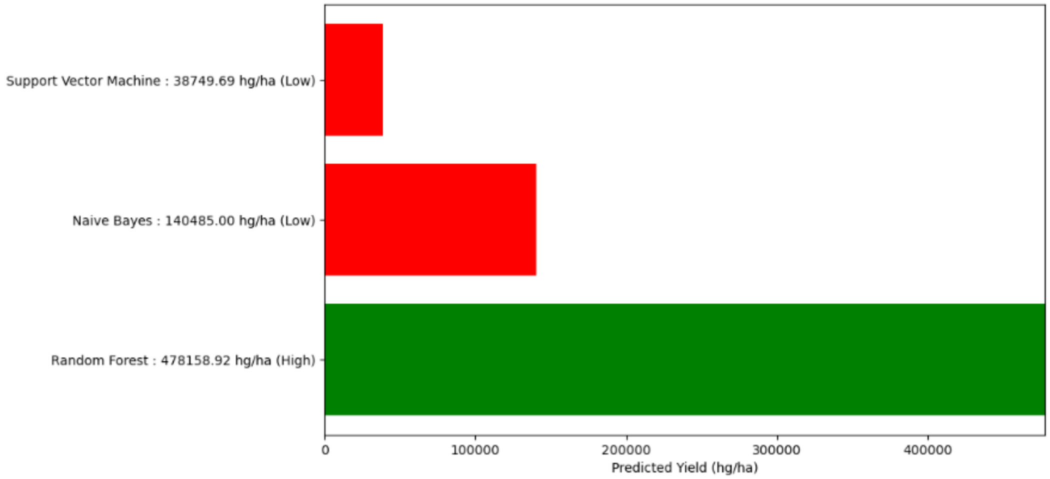


```
#saving new_data_df into actual df
new_data_df = pd.DataFrame(new_data)
# Only keep predictions from the Random Forest model
random_forest_name = 'Random Forest'  # Adjust this to the actual name of your Random Forest model
if random_forest_name in predicted_yields_and_categories:
    rf_yield, rf_category = predicted_yields_and_categories[random_forest_name]

    # Add predictions to new_data_df
    new_data_df['hg/ha_yield'] = rf_yield
    new_data_df['category'] = rf_category

    # Append new_data_df with predictions to the existing DataFrame df
    df = pd.concat([df, new_data_df], ignore_index=True)

 # Save the updated DataFrame to the existing CSV file, replacing it
    df.to_csv('yield_df.csv', index=False)

#again training df
# Assuming the target column is 'hg/ha_yield' (crop yield)
X = df.drop(columns=['hg/ha_yield','category'])  # Features (excluding target)
y = df['hg/ha_yield']  # Target (yield)

# Encode categorical variables
label_encoder = LabelEncoder()
X['Area'] = label_encoder.fit_transform(X['Area'])
X['Item'] = label_encoder.fit_transform(X['Item'])

# Split data into training and testing sets (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize the numerical features using StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

5.2.8 Prediction Result in Table and Graph



	Model	Predicted Yield (hg/ha)	Category
0	Random Forest	478158.920000	High
1	Naive Bayes	140485.000000	Low
2	Support Vector Machine	38749.686293	Low



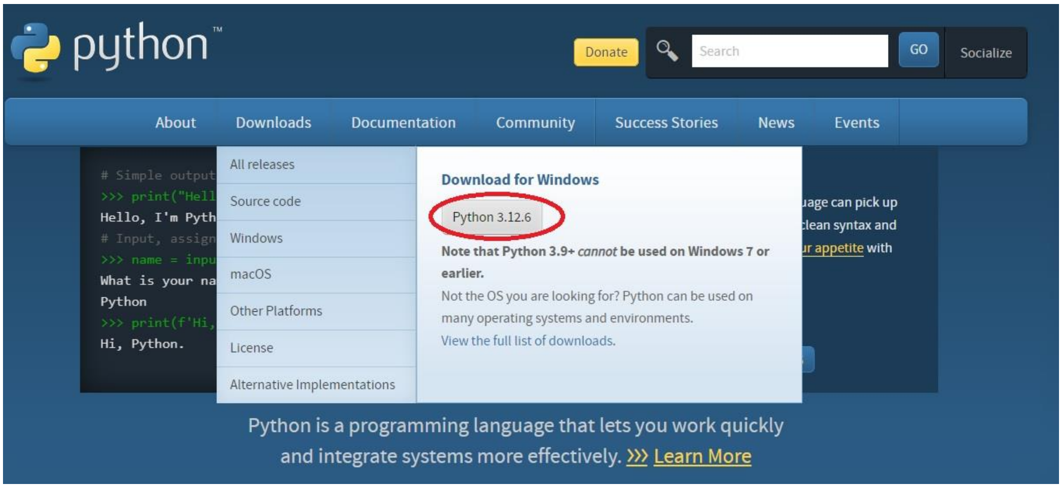
5.3 Final Findings

It is clearly mentioned from the graph and accuracy that Random Forest is more accurate on data set than Naïve Bayes and Support Vector Machine.

Chapter 6: Installation Guide

6.1 Install Python

- Go to https://www.python.org/ and download python latest version



```
# Simple output
>>> print("Hell
Hello, I'm Pyth
# Input, assign
>>> name = inpu
What is your na
Python
>>> print(f'Hi,
Hi, Python.
```

Download for Windows

Note that Python 3.9+ *cannot* be used on Windows 7 or earlier.

Not the OS you are looking for? Python can be used on many operating systems and environments.

View the full list of downloads.

uage can pick up
clean syntax and
r appetite with

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

- Install Python
- Open CMD and write ***python*** to check if it is installed

6.2 Install Jupyter Notebook

- First download Anaconda from https://www.anaconda.com/download, it is a python distribution
- Install Anaconda, then start Anaconda
- Start Jupyter Notebook from the given software

6.3 Install Node.JS

- Go to https://nodejs.org/en/download/package-manager and download LTS version
- Install Node.JS
- Open CMD and write ***node*** to check if it is installed

6.4 Install Tableau

- Go to https://www.tableau.com/products/public and install tableau public

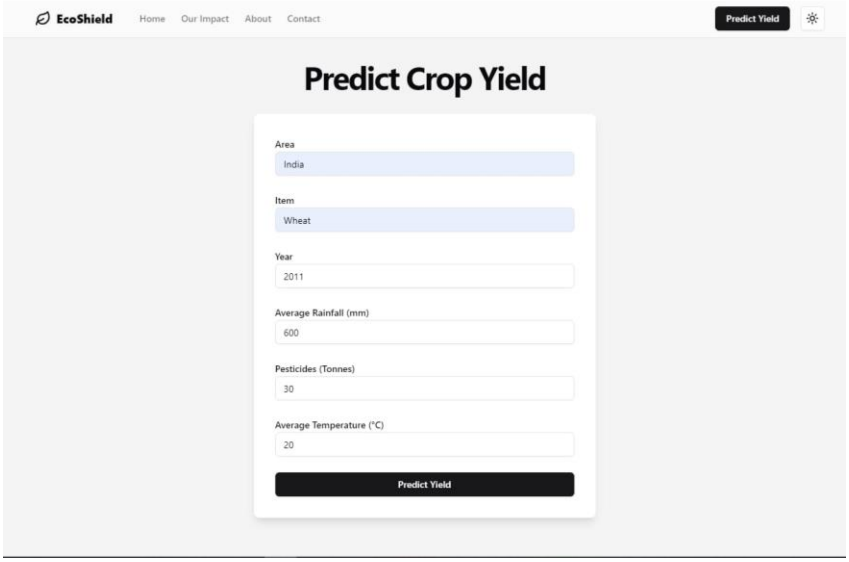
6.5 Install Libraries

- All packages are listed in ***requirement.txt*** file in ***Source Code*** folder->***Flask-app*** folder
- Run command ***pip install -r requirements.txt***

Chapter 7: Steps to Execute Project

7.1 User Interface

- Open Next.JS project and run command ***npm i*** to download node modules
- Run command ***npm run dev*** to execute project. The project will start on localhost
- On UI, there is a home page, navigation bar, findings of project. Along with that there is a form for user to input data for prediction.



Predict Crop Yield

Area: India

Item: Wheat

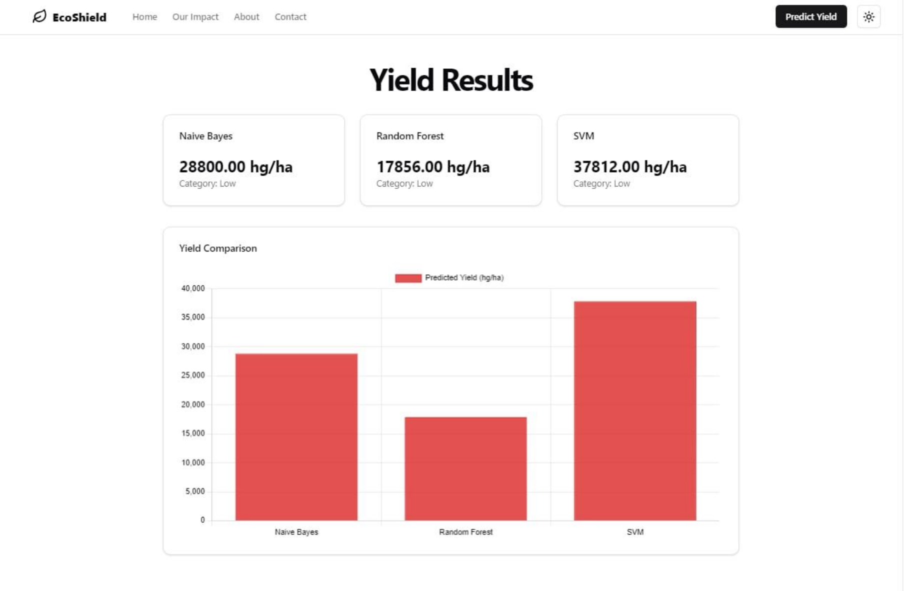
Year: 2011

Average Rainfall (mm): 600

Pesticides (Tonnes): 30

Average Temperature (°C): 20

- Input data like area, item, year, rainfall in mm, pesticide in tonnes, and average temperature in centigrade.
- Click on ***Predict Yield*** button, it will show the predicted result



Yield Results

Naive Bayes
28800.00 hg/ha
Category: Low

Random Forest
17856.00 hg/ha
Category: Low

SVM
37812.00 hg/ha
Category: Low

7.2 Python Programming

- Open the Jupyter Notebook file and execute scripts. Already mentioned in Chapter 5
- Execute ***jupyter nbconvert --to python [Python File Address]*** command to convert .***ipynb*** file into **.py** file
- *Example: jupyter nbconvert --to python C:\Users\Lab-212\Documents\EcoShield\misc\merged_notebook.ipynb*

Chapter 8: Technologies and Platforms

8.1 Tools

Jupyter Notebook and Tableau are utilized for data analysis and visualization, aiding in interactive computing and graphical representation.

8.2 Software

VS Code and PyCharm serve as integrated development environments (IDEs), providing efficient coding, debugging, and testing platforms.

8.3 Languages/Frameworks

Python, NextJS, Tailwind, and flask are the core programming languages and frameworks used for data processing, analysis, and backend development in the project.

8.4 Libraries

Library	Description
NumPy	A fundamental library for numerical computing in Python, providing support for arrays and a wide range of mathematical functions.
Pandas	A powerful data manipulation and analysis library that offers data structures like DataFrames for handling structured data.
Scikit-learn	A comprehensive library for machine learning that provides simple and efficient tools for data mining and data analysis.
Seaborn	A statistical data visualization library built on Matplotlib, offering a high-level interface for drawing attractive and informative graphics.
Matplotlib	A versatile plotting library for creating static, animated, and interactive visualizations in Python.
GeoPandas	An extension of Pandas that enables the handling and analysis of geospatial data, allowing for easy manipulation of geographic information.
Plotly	A graphing library that enables the creation of interactive visualizations and dashboards, making it easy to share results online.

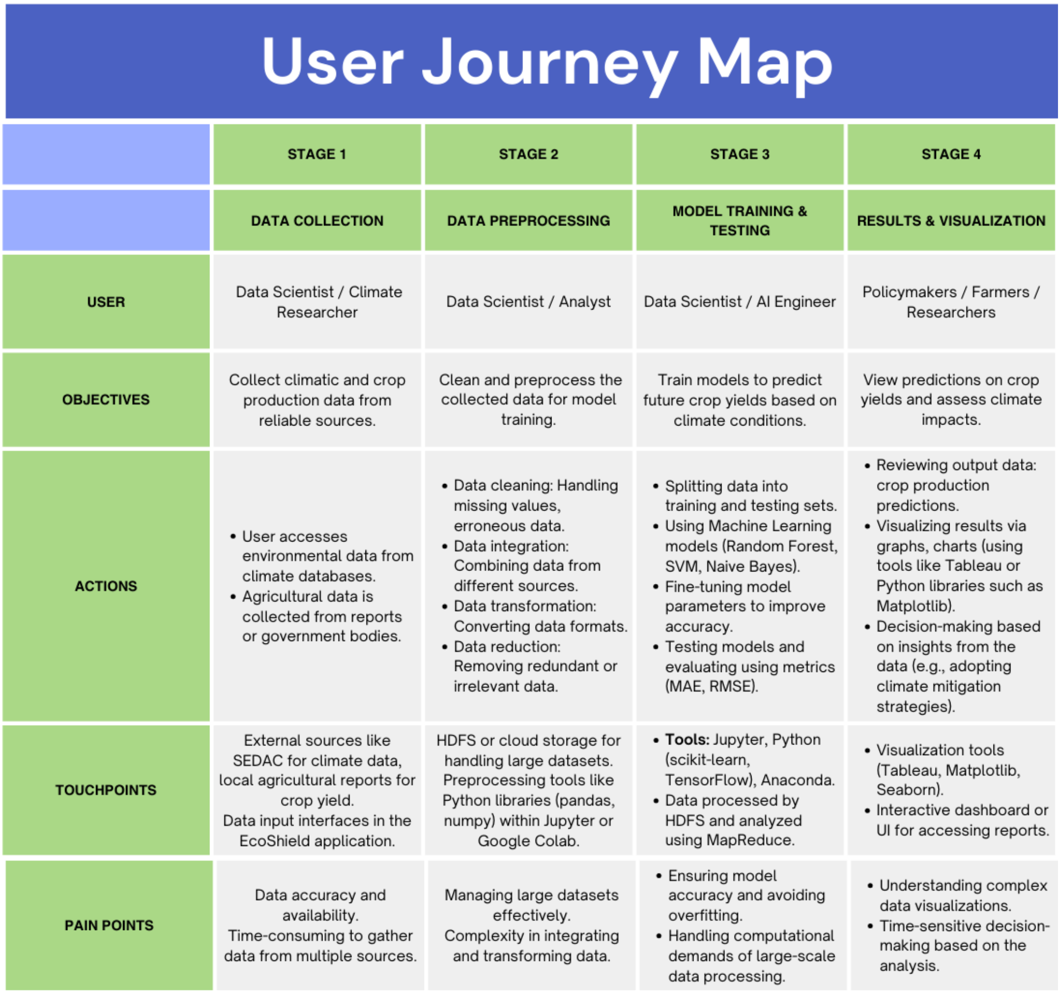
8.5 Functions

Function	Description
train_test_split	A function to split datasets into training and testing sets for model evaluation.
LabelEncoder	A utility for converting categorical labels into numerical format.
StandardScaler	A tool for standardizing features by removing the mean and scaling to unit variance.

OneHotEncoder	A method for converting categorical variables into a format suitable for machine learning by creating binary columns for each category.
MinMaxScaler	A technique that scales features to a fixed range, typically [0, 1], to ensure uniformity.
GaussianNB	A Naive Bayes classifier based on the Gaussian distribution, used for probabilistic classification.
SVR	Support Vector Regression, a method for predicting continuous values based on support vector machines.
RandomForestRegressor	An ensemble learning method that builds multiple decision trees for regression tasks to improve accuracy.
mean_squared_error	A function that calculates the average squared difference between predicted and actual values, useful for assessing model performance.
r2_score	A metric that provides an indication of how well the model's predictions approximate the actual data points.
confusion_matrix	A function that creates a matrix to visualize the performance of a classification algorithm by showing true vs. predicted classifications.
ConfusionMatrixDisplay	A utility for visualizing confusion matrices in an easily interpretable format.
mean_absolute_error	A metric that calculates the average absolute difference between predicted and actual values, providing insight into model accuracy.

Chapter 9: Diagrams

9.1 User Journey Map

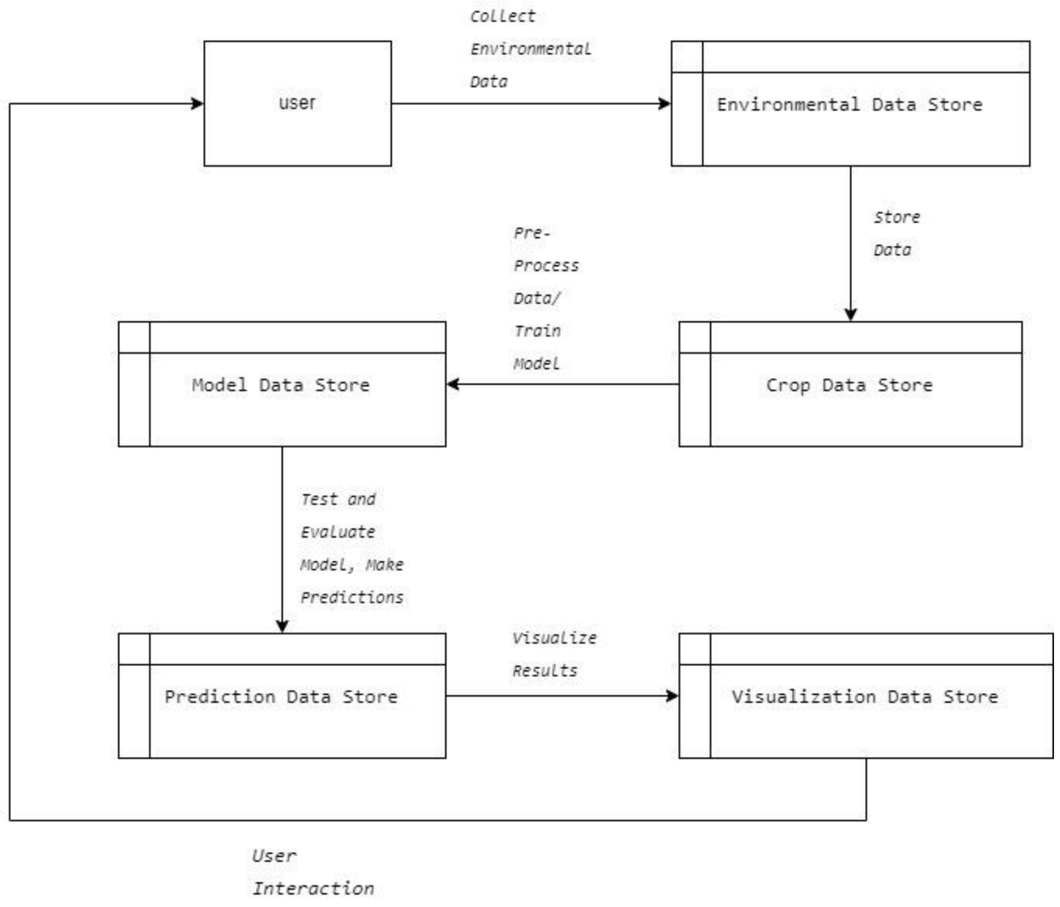


User Journey Map

	STAGE 1	STAGE 2	STAGE 3	STAGE 4
	DATA COLLECTION	DATA PREPROCESSING	MODEL TRAINING & TESTING	RESULTS & VISUALIZATION
USER	Data Scientist / Climate Researcher	Data Scientist / Analyst	Data Scientist / AI Engineer	Policymakers / Farmers / Researchers
OBJECTIVES	Collect climatic and crop production data from reliable sources.	Clean and preprocess the collected data for model training.	Train models to predict future crop yields based on climate conditions.	View predictions on crop yields and assess climate impacts.
ACTIONS	• User accesses environmental data from climate databases. • Agricultural data is collected from reports or government bodies.	• Data cleaning: Handling missing values, erroneous data. • Data integration: Combining data from different sources. • Data transformation: Converting data formats. • Data reduction: Removing redundant or irrelevant data.	• Splitting data into training and testing sets. • Using Machine Learning models (Random Forest, SVM, Naive Bayes). • Fine-tuning model parameters to improve accuracy. • Testing models and evaluating using metrics (MAE, RMSE).	• Reviewing output data: crop production predictions. • Visualizing results via graphs, charts (using tools like Tableau or Python libraries such as Matplotlib). • Decision-making based on insights from the data (e.g., adopting climate mitigation strategies).
TOUCHPOINTS	External sources like SEDAC for climate data, local agricultural reports for crop yield. Data input interfaces in the EcoShield application.	HDFS or cloud storage for handling large datasets. Preprocessing tools like Python libraries (pandas, numpy) within Jupyter or Google Colab.	• **Tools:** Jupyter, Python (scikit-learn, TensorFlow), Anaconda. • Data processed by HDFS and analyzed using MapReduce.	• Visualization tools (Tableau, Matplotlib, Seaborn). • Interactive dashboard or UI for accessing reports.
PAIN POINTS	Data accuracy and availability. Time-consuming to gather data from multiple sources.	Managing large datasets effectively. Complexity in integrating and transforming data.	• Ensuring model accuracy and avoiding overfitting. • Handling computational demands of large-scale data processing.	• Understanding complex data visualizations. • Time-sensitive decision-making based on the analysis.

The **EcoShield** user journey outlines the key steps a user follows to interact with the platform. It begins with **data collection and input**, where users gather and upload relevant climate and agricultural datasets. After input, users proceed to the **data processing stage**, where the system cleans and pre-processes the data, making it suitable for model training. The next step involves **model selection**, where users choose from various machine learning algorithms, such as Random Forest, Support Vector Machine, or Naive Bayes, to predict crop yields. Once the model is trained and tested, users move on to the **prediction and visualization stage**, where they can analyze predictions through interactive dashboards featuring charts and heatmaps. Finally, users have the option to **refine and iterate** by adjusting input parameters or exploring different models for more accurate results. This user-centric flow ensures that stakeholders gain actionable insights to address the impact of climate change on food production.

9.2 Data Flow Diagram



The **EcoShield** Data Flow Diagram (DFD) illustrates the flow of information and processes within the system. The process starts with the **user** collecting environmental and crop-related data, which is stored in the **Environmental Data Store** and **Crop Data Store**, respectively. The data is then pre-processed, and a model is trained using the stored datasets. This model is saved in the **Model Data Store**, where it undergoes evaluation and testing. Once the predictions are generated, they are stored in the **Prediction Data Store**. Finally, the prediction results are visualized using the **Visualization Data Store**, enabling users to analyze and make informed decisions. The entire process emphasizes user interaction, ensuring that the data collection, model evaluation, and visualization are integrated seamlessly for effective analysis of climate change impacts on global food production.

Chapter 10: Task Sheet

Task No.	Task	Team Member	Due Date	Status	Comments/Notes
1	**Requirement Gathering**	Ahsan, Umer, Aleena, Nimra, Hashir	18-Sep-2024	Completed	Review SRS document and confirm functional and non-functional requirements.
2	**Dataset Collection**	Ahsan	18-Sep-2024	Completed	Download datasets from SEDAC Crop-Climate. Ensure dataset completeness and format compatibility.
3	**Data Preprocessing**	Ahsan	19-Sep-2024	Completed	Clean, integrate, and transform data. Apply data reduction techniques for efficient processing.
4	**Model Training**	Ahsan	20-Sep-2024	Completed	Implement Random Forest, SVM, and Naive Bayes models. Split the dataset into training and testing subsets.
5	**Model Evaluation & Testing**	Ahsan, Umer, Hashir	21-Sep-2024	Completed	Evaluate models using MAE, MSE, RMSE, and R^2 metrics. Conduct significance tests for statistical validity.
6	**Prediction and Output**	Ahsan	21-Sep-2024	Completed	Generate predicted crop yield estimates and production outputs. Ensure accurate presentation of data.
7	**Visualization Development**	Hashir	22-Sep-2024	Completed	Create graphs, charts, and heatmaps to represent model accuracy and crop yield predictions using Tableau or Matplotlib.
8	**User Interface (UI) Design**	Umer	22-Sep-2024	Completed	Design a user-friendly interface for data input and result display. Integrate feedback mechanisms.

9	**System Integration**	Umer	23-Sep-24	Completed	Integrate data collection, preprocessing, model training, evaluation, and prediction components.
10	**Final Testing & QA**	Ahsan, Umer, Hashir	23-Sep-2024	Completed	Test the entire system for bugs, data accuracy, and output reliability. Conduct stress testing on large datasets.
11	**Project Documentation**	Nimra	23-Sep-2024	Completed	Write project report, including problem definition, design specifications, diagrams (DFD, User Flow), and installation instructions. Upload project to GitHub.
12	**Blog Writing (2000 words)**	Aleena	23-Sep-2024	Completed	Write and publish a blog on the EcoShield project, focusing on its significance, methodology, and outcomes. Submit the blog link.
13	**Video Demonstration**	Hashir	23-Sep-2024	Completed	Create a detailed video demonstrating the application’s working, explaining functionalities and key features. Upload and submit the video in .mp4 format.
14	**GitHub Repository Setup**	Umer	23-Sep-2024	Completed	Set up GitHub repository, ensure public access, upload the complete project code, README file, and project documentation. Share the repository link with the team and stakeholders.

Chapter 11: Conclusion & Future Work

11.1 Conclusion

The EcoShield project successfully addresses the critical issue of climate change and its effects on global food production by leveraging Big Data and Data Science techniques. Through the integration of machine learning models like Random Forest, Support Vector Machine (SVM), and Naive Bayes, along with advanced data processing tools such as Hadoop Distributed File System (HDFS) and MapReduce, EcoShield provides accurate predictions of crop yield and production under varying climatic conditions. The project has proven to be an innovative solution for policymakers, researchers, and farmers, offering data-driven insights that aid in making informed decisions to mitigate the impact of climate change on agriculture.

By processing and analyzing large datasets that include climatic and agricultural data, the project demonstrates the potential of modern technology in addressing global food security challenges. The predictive capabilities of EcoShield, combined with user-friendly visualizations, ensure that stakeholders can interpret and apply the results to improve agricultural practices, thus contributing to a more sustainable future for the planet.

11.2 Future Work

While EcoShield has made significant strides in predicting the impacts of climate change on food production, several areas can be explored in future iterations to enhance its capabilities:

1. **Incorporation of Real-Time Data**: Future work could involve integrating real-time data sources, such as IoT-based agricultural sensors, satellite imagery, and live weather data, to refine predictions and offer more dynamic, real-time insights.
2. **Expansion of Climate Models**: Including additional climate models and scenarios will allow for more robust predictions, catering to diverse geographical regions and agricultural practices. This could make the tool adaptable to various global contexts and emerging climate patterns.
3. **Decision Support System**: Developing an automated decision support system that provides actionable recommendations for farmers and policymakers based on the predicted data would significantly enhance EcoShield's practical value. This could include tailored advice on crop selection, planting schedules, and irrigation practices.
4. **AI-Driven Optimization**: Incorporating advanced artificial intelligence techniques like reinforcement learning could further optimize crop yield predictions and resource management strategies by simulating various climate and farming conditions.
5. **User Interface Improvements**: Improving the user interface by making it more interactive and accessible to non-technical users could broaden EcoShield's

usability, ensuring that more stakeholders, especially local farmers, can benefit from its insights.

6. **Global Collaboration**: Expanding the platform to encourage collaboration between governments, research institutions, and agricultural organizations worldwide will enable the creation of a global repository of climate-agriculture data, fostering collective efforts toward climate resilience.

In summary, EcoShield provides a solid foundation for climate change prediction and mitigation in agriculture, with future enhancements likely to make it an indispensable tool in the global effort to ensure food security in the face of a changing climate.

Chapter 12: Git Hub Link of Project

https://github.com/umer-rukhsar/MSG-Data-Doer-EcoShield-

Chapter 13: Blog’s Link

https://danthewanderer.blogspot.com/2024/09/eco-shield-bridging-technology-and.html