

**APTECH GARDEN CENTER**

**E-Project**

**EarthScape**

|  |  |
| --- | --- |
|  |  |

APWA Complex, 1st Floor, Agha Khan 3 Rd, Garden East Saddar Town, Karachi Pakistan.

Hardware/ Software Requirements

**Hardware**

* Intel Core i5/i7 Processor or higher
* 8 GB RAM or higher Color SVGA
* 500 GB Hard Disk space
* Mouse
* Keyboard

Technologies Used in the Project

In the development of this project, I have utilized a combination of powerful technologies to ensure efficient functionality, smooth operations, and insightful reporting. The following technologies were employed:

* **Python**: Used as the primary programming language for the development of the project. Python provided a solid foundation for implementing core functionalities and logic.
* **Django**: Employed as the web framework to build the backend of the application. Django was used for developing APIs and handling server-side logic efficiently.
* **SQLite 3**: Chosen as the database for the project due to its lightweight and efficient nature, making it ideal for small-scale applications. It was used to manage and store data effectively.
* **Tableau**: Utilized for data visualization and reporting. Tableau helped in representing the project’s analysis and results in an easy-to-understand graphical format, providing valuable insights for stakeholders.
* **Jupyter Notebook: Used for model training and data analysis. Jupyter Notebook provided an interactive environment to experiment with different machine learning models, preprocess data, and visualize results step-by-step, allowing for easy debugging and documentation during the training process.**

Operating System

The dashboard can be accessed through a web browser and does not require any specific installation on your operating system. However, the application relies on the following technologies:

* Streamlit: for the web framework.
* Plotly: for interactive data visualizations.
* Pandas: for data manipulation and analysis.

The EarthScape Futuristic Data Analysis Dashboard is compatible with the following operating systems:  
1. Windows  
2. macOS  
3. Linux

EarthScape Futuristic Data Analysis Dashboard - User Guide

**Introduction**

Welcome to the EarthScape Futuristic Data Analysis Dashboard! This guide will walk you through how to use the platform to upload, visualize, and analyze environmental data effectively. Whether you are working with CSV or Excel files, the dashboard provides comprehensive features for data input, search, filtering, visualization, and report generation.

**Prerequisites**

Before getting started, ensure you have the following:   
1. A CSV or Excel file with environmental data.   
2. A modern web browser (Google Chrome, Mozilla Firefox, etc.) to run the application.

**Dashboard Layout**

The EarthScape dashboard consists of the following sections:   
1. Navigation Bar: Allows easy access to different sections of the dashboard (Home, Data Input, Visualizations, Search Data, About).  
2. Home: A welcome section that explains the purpose of the dashboard.  
3. Data Input: Upload or manually enter your environmental data.  
4. Search Data: Filter and search through your uploaded data.  
5. Visualizations: Create pie charts, bar charts, and scatter plots based on your data.  
6. Report Generation: Download filtered data as CSV or Excel files for further analysis.  
7. About: Learn more about the EarthScape project.

Here’s an updated and comprehensive document that includes instructions for running **sorting**, **cleaning**, and **model training** along with the website:

Installing and Running the Website and Associated Tasks on Your PC

Follow these steps to set up and run the website along with sorting, cleaning, and model training functionalities on your computer.

**1. Download the Project**

* Download the project files as a **ZIP file**).

**2. Install Required Software**

1. **Visual Studio Code**: For editing and running the project files.
   * [Download VS Code](https://code.visualstudio.com/)
2. **Node.js**: For frontend or JavaScript-based dependencies (if applicable).
   * [Download Node.js](https://nodejs.org/)
3. **Python**: Ensure Python is installed (version 3.8 or higher). Check installation with:
4. python --version

**3. Running Sorting and Cleaning Tasks**

**Files to Run:**

* sort.py: For sorting the data.
* clean.py: For cleaning the dataset.

**Steps to Execute:**

1. Open the project folder in Visual Studio Code.
2. Open a new terminal and navigate to the project directory:
3. cd path/to/project
4. Run the **sorting script**:
5. python sort.py
6. Run the **cleaning script**:
7. python clean.py

**Expected Outputs:**

* The sort.py script will generate a sorted version of the dataset.
* The clean.py script will clean the dataset by removing duplicates, filling missing values, etc.
* The output files will be saved in the project folder, typically with names like sorted\_data.csv or cleaned\_data.csv.

**4. Running the Model Training Task**

**File to Run:**

* model\_training.ipynb: Jupyter Notebook for training the machine learning model.

**Steps to Execute:**

1. Open **Jupyter Notebook** using Anaconda or directly in Visual Studio Code (if Jupyter is configured).
2. Open the model\_training.ipynb file.
3. Run all cells in the notebook by clicking **Kernel > Restart & Run All**.
4. The notebook will:
   * Load the cleaned dataset.
   * Train a machine learning model (e.g., regression, classification).
   * Save the trained model to the project directory.

**Expected Outputs:**

* A trained machine learning model (e.g., model.pkl).
* Logs showing training metrics such as accuracy, loss, etc.

**5. Running the Website**

**File to Run:**

* main.py: The main Streamlit application file.

**Steps to Execute:**

1. Open the project folder in Visual Studio Code.
2. Open a terminal in the project directory.
3. Install Streamlit (if not already installed):
4. pip install streamlit
5. Run the Streamlit application:
6. streamlit run main.py

**Accessing the Website:**

* Streamlit will provide a URL (e.g., http://localhost:8501). Open it in your browser to access the website.

**6. Complete Workflow**

1. **Sorting**: Run sort.py to preprocess the raw data.
2. **Cleaning**: Run clean.py to remove inconsistencies and prepare the dataset.
3. **Model Training**: Use model\_training.ipynb to train the machine learning model.
4. **Website**: Run main.py to launch the Streamlit-based web application.

**7. Notes**

* Ensure all required files (sort.py, clean.py, model\_training.ipynb, and main.py) are in the project directory.
* If you encounter missing dependencies, install them using:
* pip install -r requirements.txt
* Place datasets (e.g., Cleaned\_EarthScape\_Dataset\_All.xlsx) in the same directory as the scripts for easy access.

This document provides a complete guide for running all tasks in your project. Let me know if any additional details are needed!

Getting Started

**1. Uploading Data**

In the 'Data Input' section, you have two options for entering your data:

* Option 1: Upload a CSV or Excel file by selecting the file from your computer.
* Option 2: Manually enter your data in CSV format.

**2. Viewing and Searching Data**

Once your data is uploaded, it will be displayed in a table. You can search for specific values by selecting a column and a value from the dropdown menu in the 'Search Data' section.

**3. Data Visualization**

In the 'Visualizations' section, you can create the following types of charts based on your filtered data:

* Pie Chart: Visualize categorical data.
* Bar Chart: Compare numeric data between two columns.
* Scatter Plot: Plot two numeric columns against each other to identify trends.

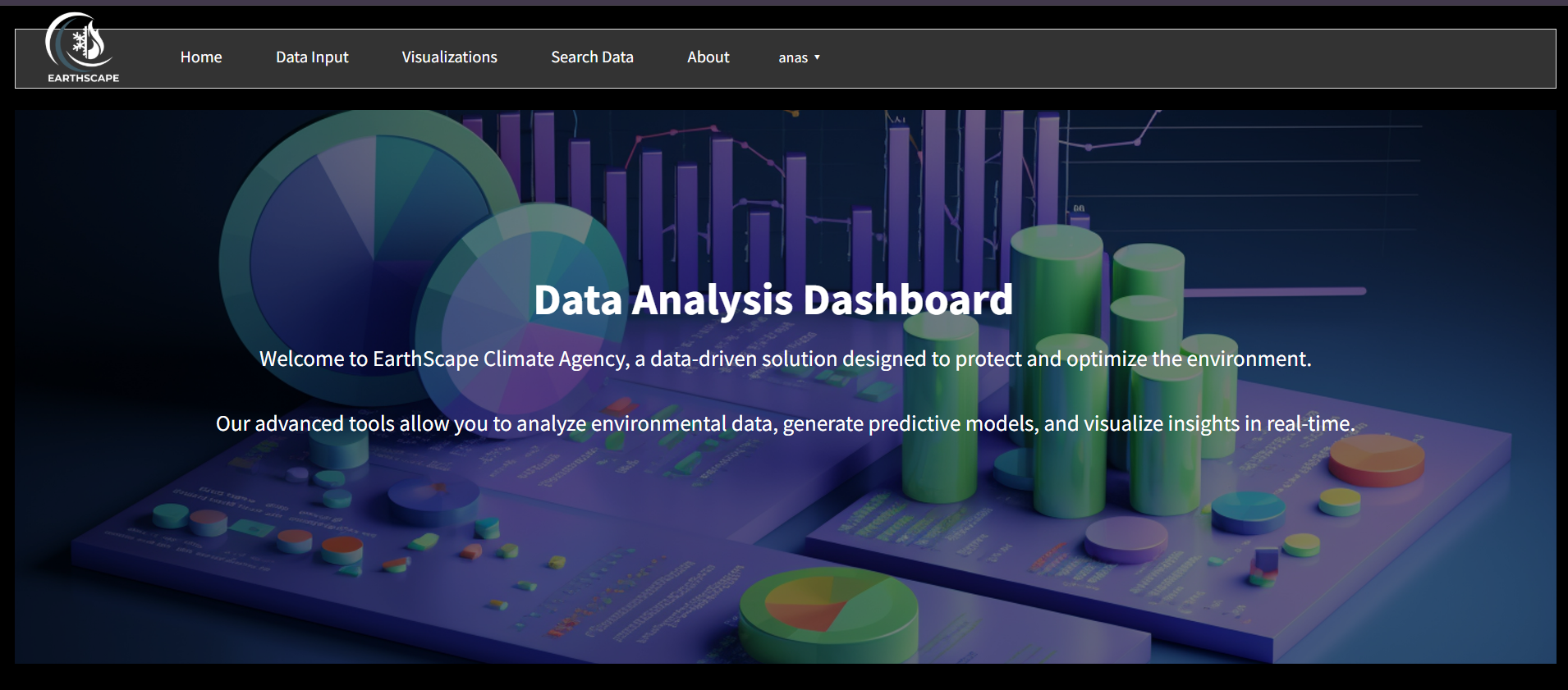
**4. Generating Reports**

After filtering your data, you can generate a report and download it as a CSV or Excel file in the 'Report Generation' section.

**About the Project**

EarthScape is a futuristic data analysis platform designed to help users analyze environmental data efficiently. Our goal is to provide easy-to-use tools that facilitate better decision-making through data.

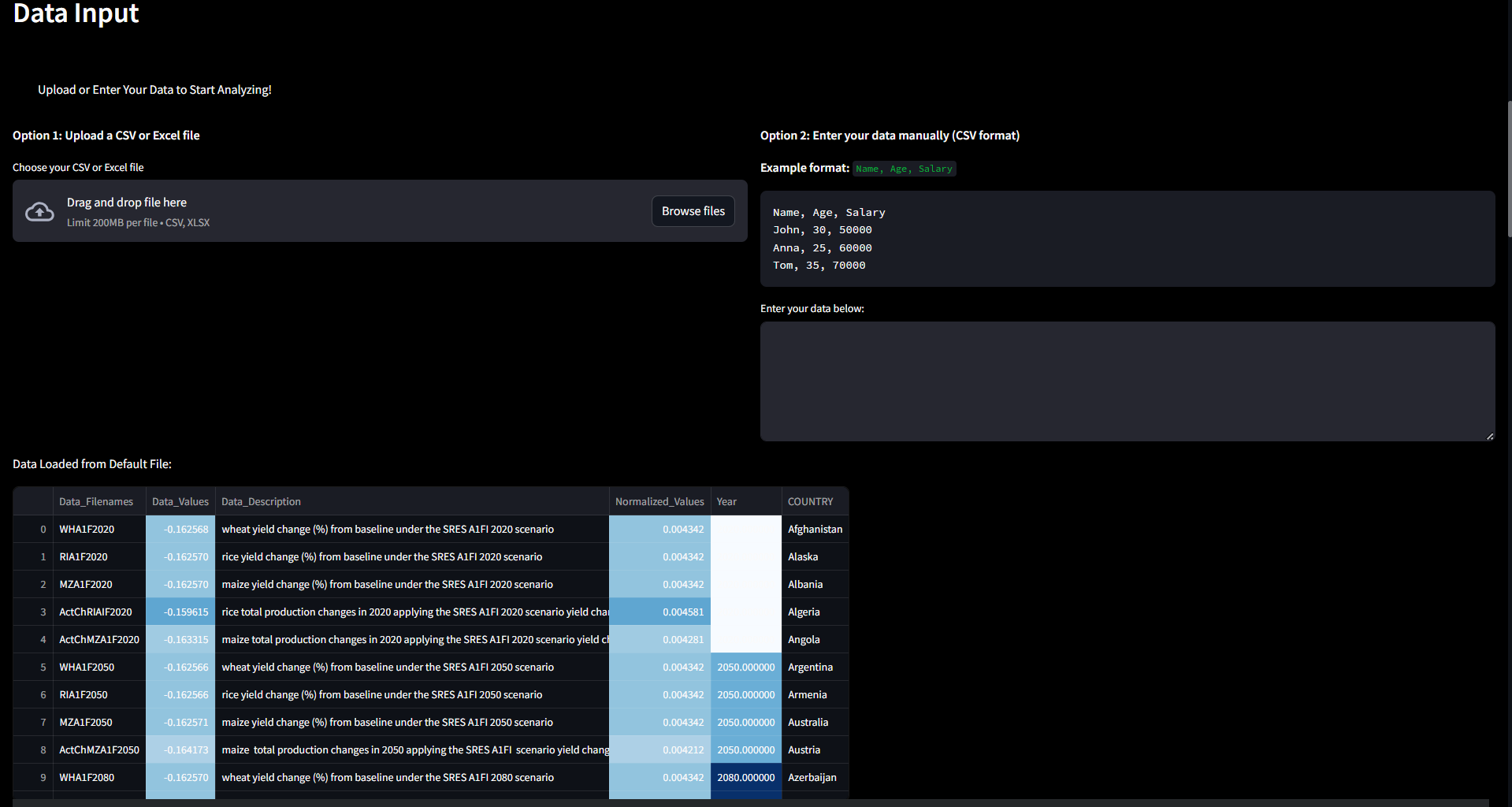
Designing Phase



**EarthScape Futuristic Data Analysis Dashboard**

**1. Home Section**

The Home Section serves as the introductory gateway to the EarthScape platform. It welcomes users with an engaging overview, highlighting the mission of the dashboard: to provide a data-driven solution for environmental protection and optimization. The text emphasizes the dashboard's capability to analyse environmental data, generate predictive models, and visualize insights in real-time, making it a powerful tool for users looking to make informed decisions based on data.



**2. Data Input Section**

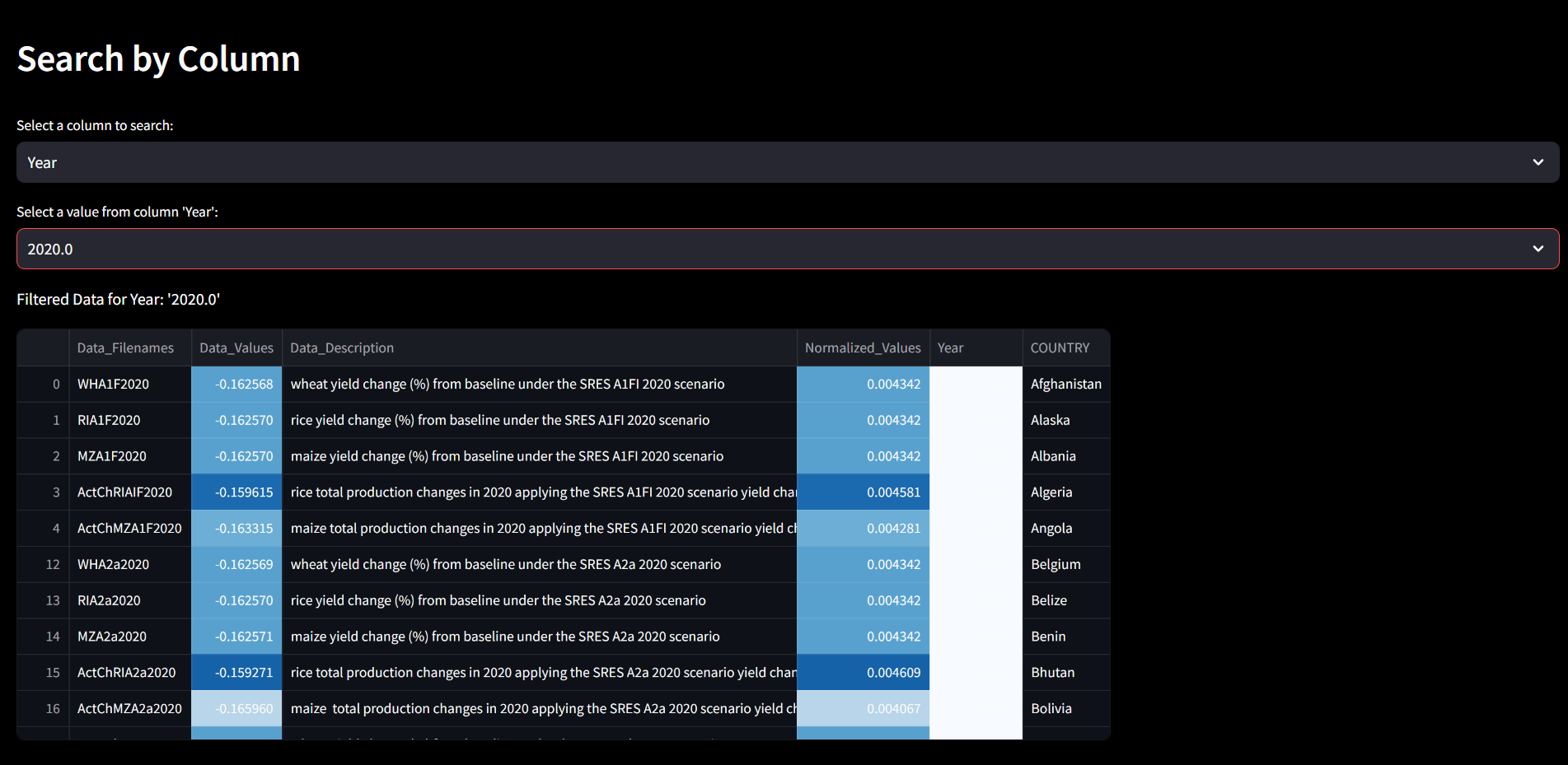
In the Data Input Section, users are given two primary options for inputting their data:

* **File Upload:**

Users can easily upload their CSV or Excel files. Once uploaded, the application displays the data in a user-friendly table format, allowing for quick verification and initial analysis. This feature ensures that users can immediately begin analysing their data without cumbersome setup processes.

* **Manual Data Entry:**

For those who prefer to input data directly, this section includes a text area where users can enter data in a specified format (e.g., Name, Age, Salary). An example format is provided to guide users, making it accessible even for those who may not be familiar with data formats.



**3. Search Data Section**

The Search Data Section enables users to filter their data dynamically. This feature is particularly useful for analysing large datasets. Users can:

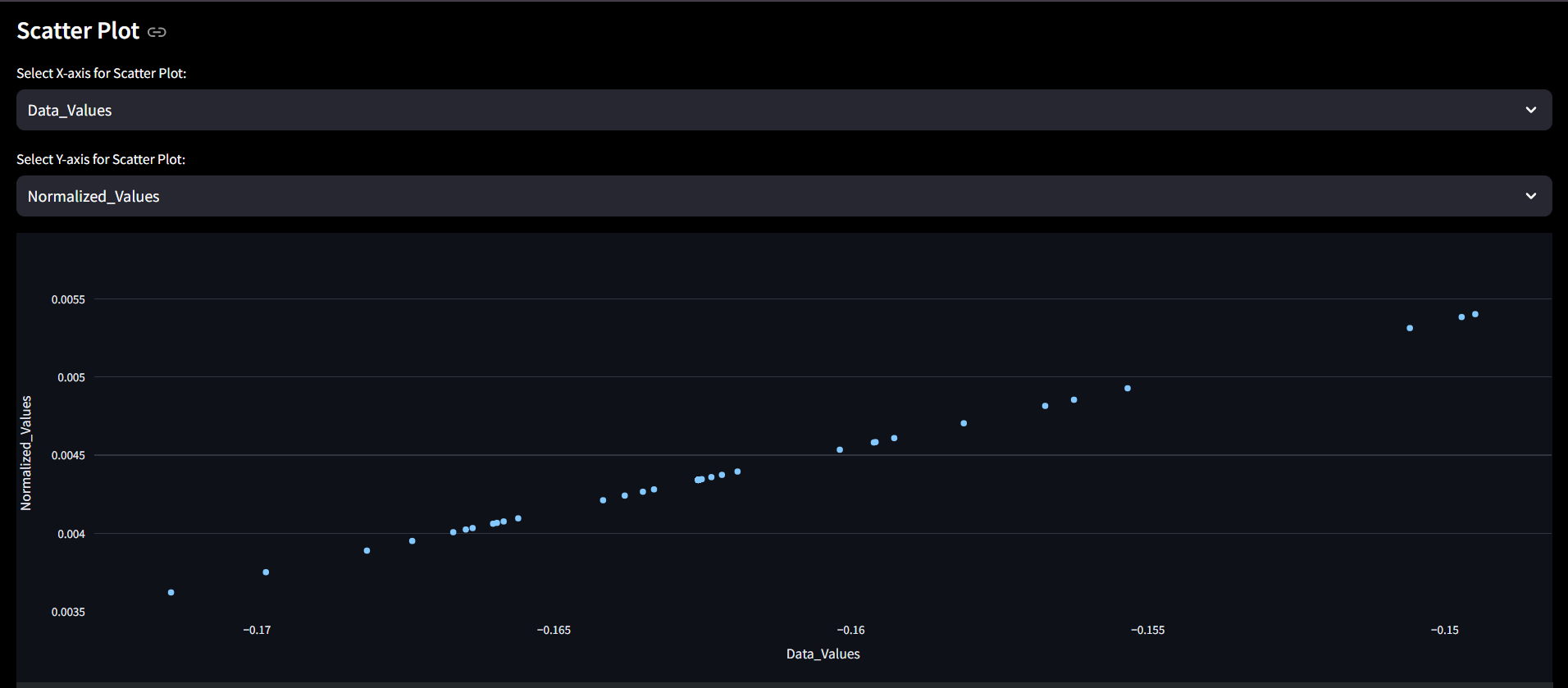
* **Select a Column:**

They choose from the available columns in their dataset, facilitating focused analysis based on specific attributes.

* **Select a Value:**

Once a column is selected, users can pick a specific value from that column to filter the dataset. The filtered results are displayed in a table, allowing users to hone in on specific data points of interest.





**4. Visualizations Section**

The Visualizations Section provides users with tools to create meaningful graphical representations of their filtered data. This section includes:

* **Pie Chart:**

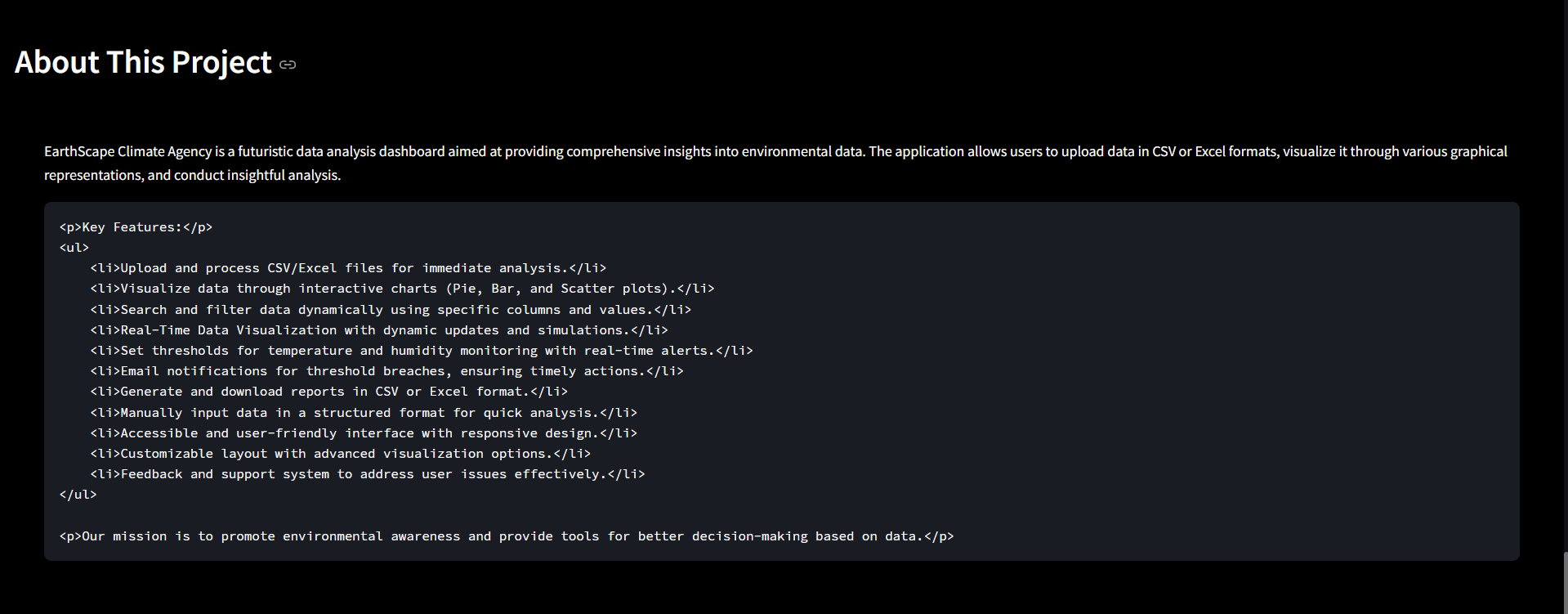
Users can create pie charts from categorical data, helping visualize the distribution of different categories within their dataset.

* **Bar Chart:**

This feature allows users to select numeric columns for the x and y axes, providing a way to visualize relationships and comparisons between different data points.

* **Scatter Plot:**

Users can create scatter plots to explore correlations between two numeric variables, offering insights into trends and patterns in the data.



**5. About Section**

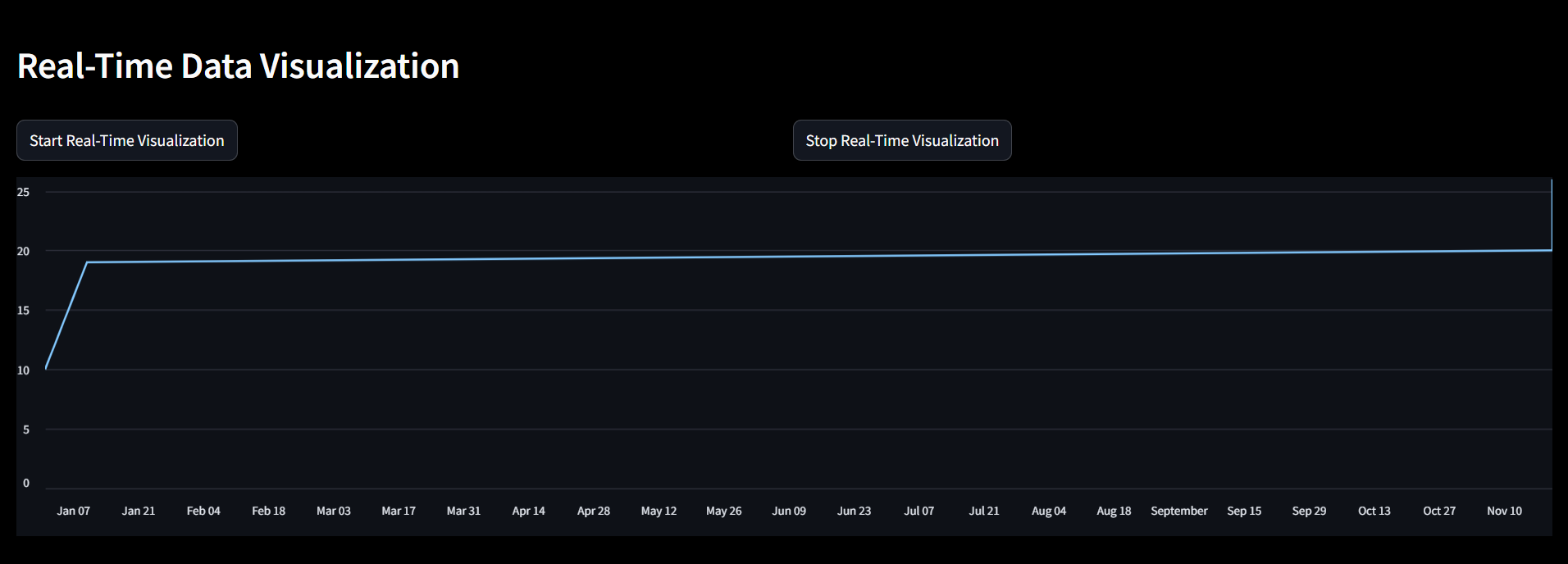
The About Section gives users an overview of the EarthScape project, outlining its objectives and significance. It includes:

* **Project Overview:**

A description of the dashboard’s purpose, focusing on its role in promoting environmental awareness and providing essential tools for data analysis.

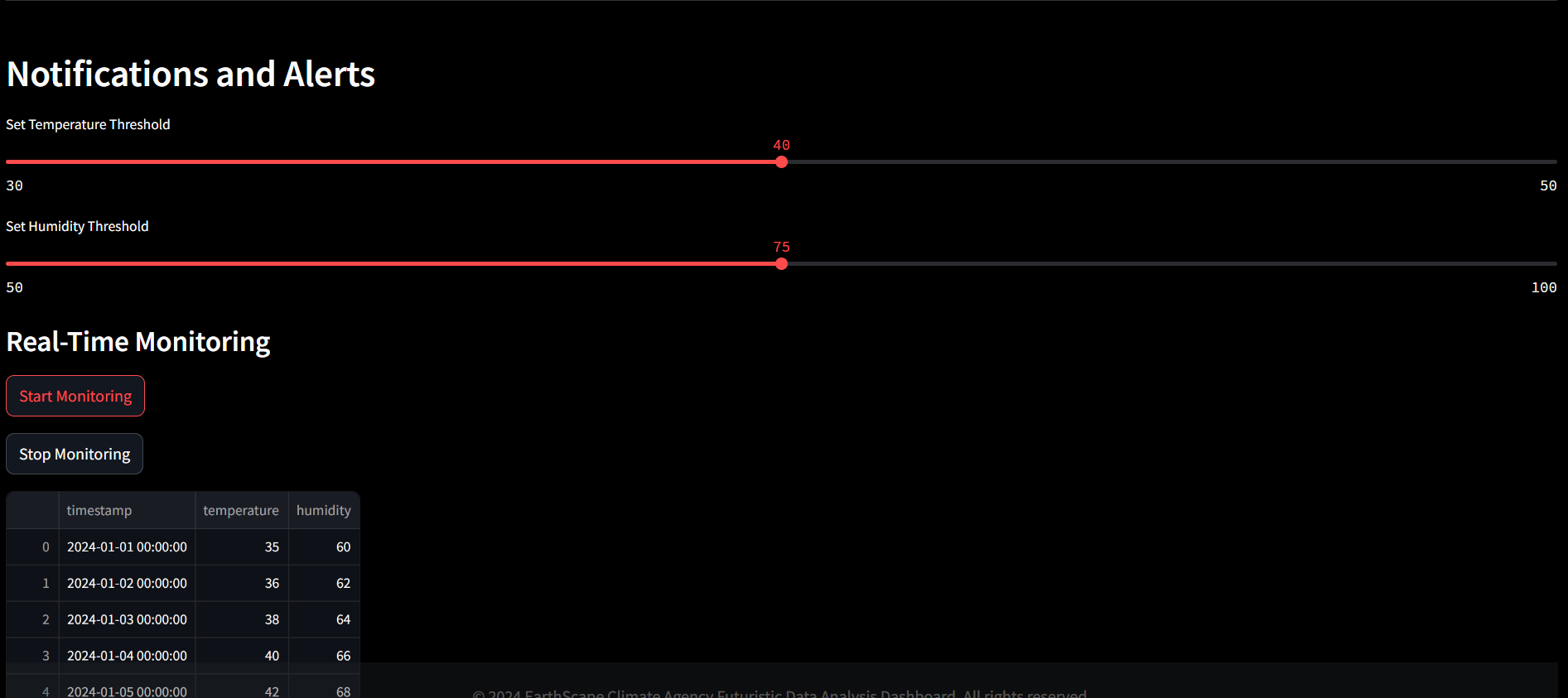
* **List of Features:**

A summary of key functionalities, such as the ability to upload and process files, visualize data through interactive charts, and perform dynamic searches. This section highlights the user-friendly design and accessibility of the dashboard.



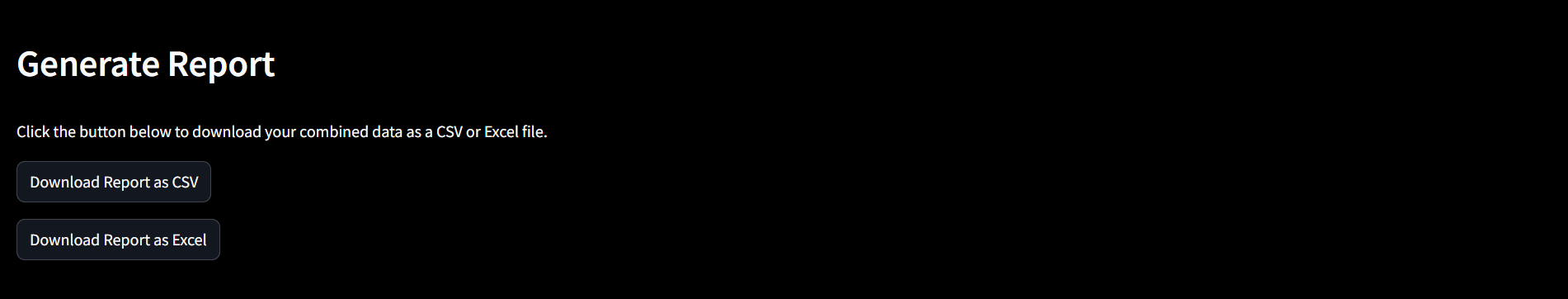
**Real-Time Data Visualization Dashboard**

The image provided represents a dashboard designed for real-time data visualization. This dashboard allows users to monitor and analyze data trends dynamically as they are updated in real-time. It provides an interactive interface for controlling the visualization process and a clean design to make data analysis intuitive and user-friendly. Below is a comprehensive explanation of its components, functionality, and significance.



**Dashboard for Notifications and Alerts with Real-Time Monitoring**

The provided dashboard focuses on **real-time monitoring** and dynamic threshold-based notifications. It allows users to set specific thresholds for environmental factors like temperature and humidity while enabling real-time data updates and monitoring. Here's a detailed explanation of its structure, functionality, and significance.



**Report Generation**

In the Generate Report section of the EarthScape Data Analysis Dashboard, users can easily download the filtered data they have processed. This feature allows for seamless exporting of data in both CSV and Excel formats, enhancing the usability of the application.

**Key Features:**

* **Download as CSV**:

Users can click the "Download Report as CSV" button to export the current filtered dataset in CSV format. This is particularly useful for those who prefer working with data in spreadsheet applications or for further analysis.

* **Download as Excel**:

Users have the option to download the data as an Excel file by clicking the "Download Report as Excel" button. This allows for better formatting and the ability to utilize Excel's advanced features.

**Usage:**

1. **Filtered Data Requirement**:

Before generating a report, users must first upload or enter their data and apply any desired filters. The report will be generated based on the currently displayed filtered dataset.

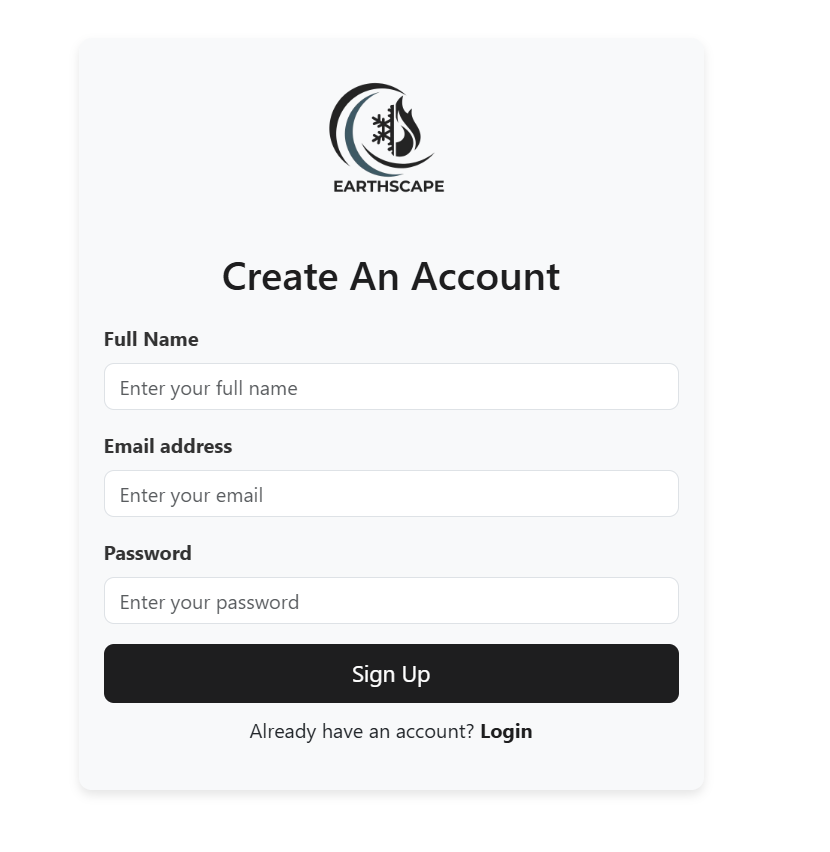
1. **Download Links**:

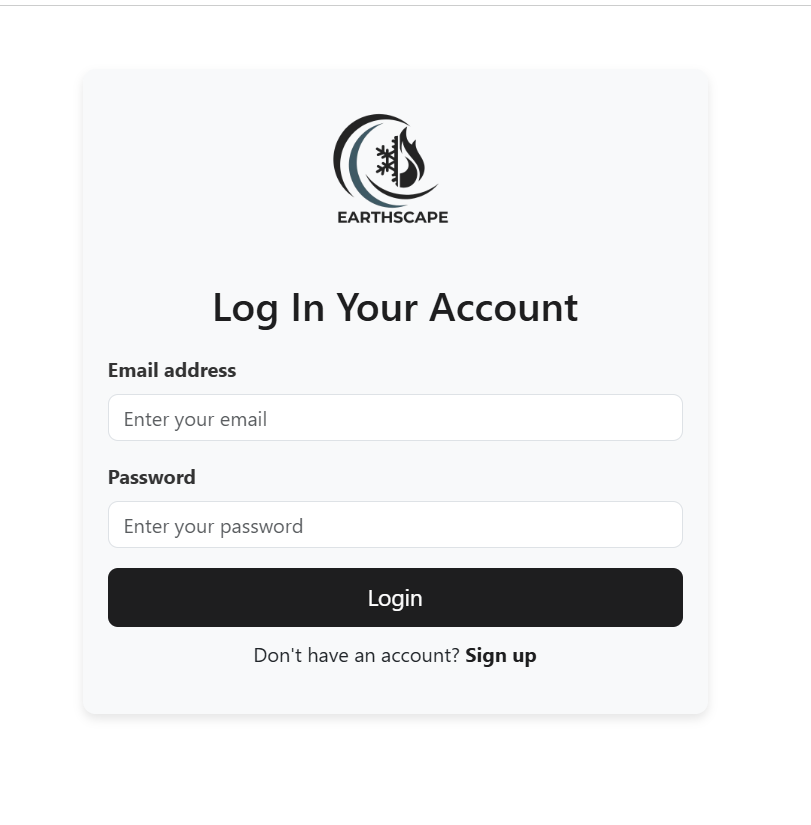
Once the user clicks on the respective download button, a link will be generated. Clicking this link will initiate the download of the report in the chosen format.



**Feedback and Support Dashboard**

The provided image showcases a **Feedback and Support** interface designed to collect user input for improving the system, addressing user concerns, and resolving issues. Below is a detailed description of the form and its functionalities:





Earthscape Login & Signup Details:

1. **Login**: Allows existing users to access their Earthscape accounts using their registered email and password.
2. **Signup**: Enables new users to create an account by providing their name, email, and password, ensuring secure access to Earthscape services.

Model Training Using Jupiter

**1. Data Splitting:**

* The data is split into training and testing sets using an 80/20 split (`train\_test\_split`). `X\_train` and `X\_test` represent the feature sets for training and testing, while `Y\_train` and `Y\_test` represent the target values.

**2. Model Selection:**

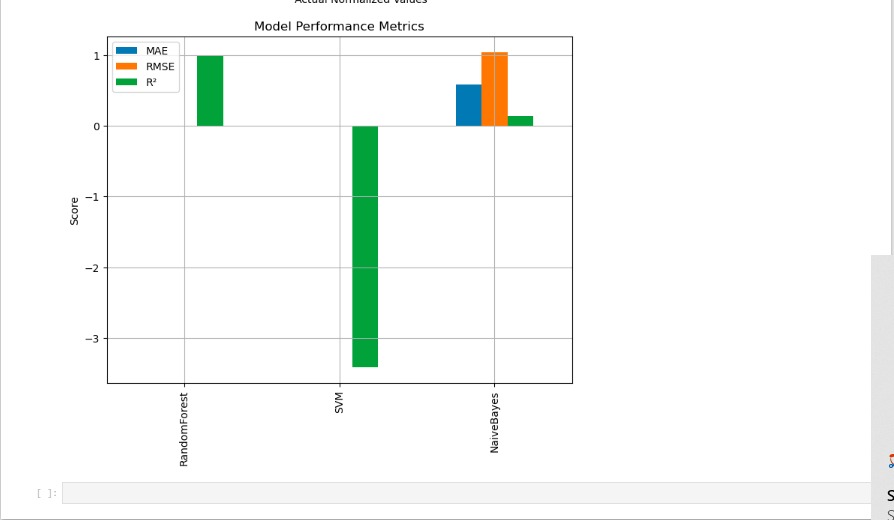
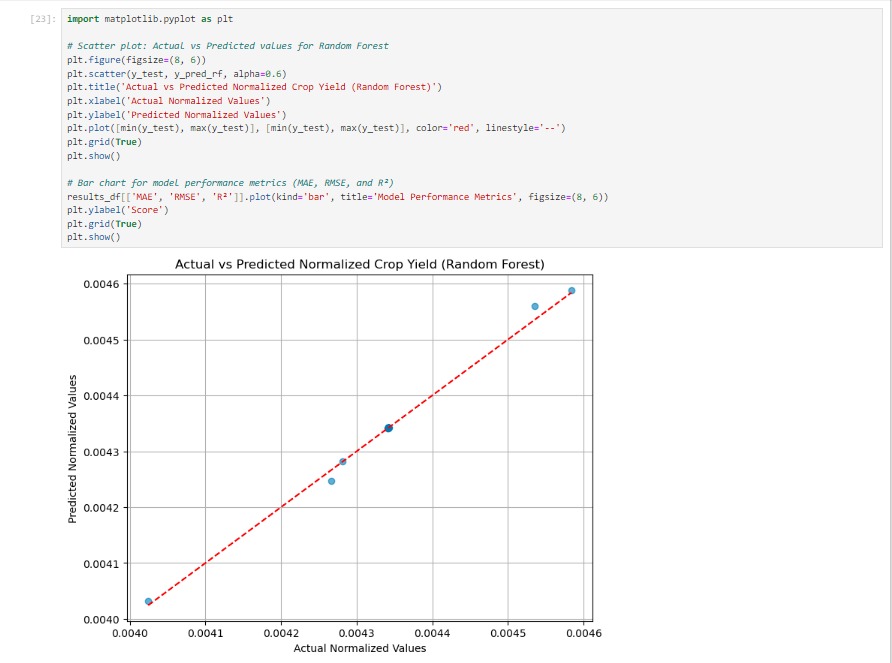
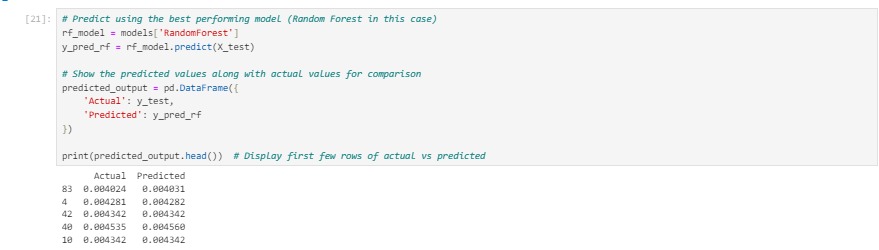
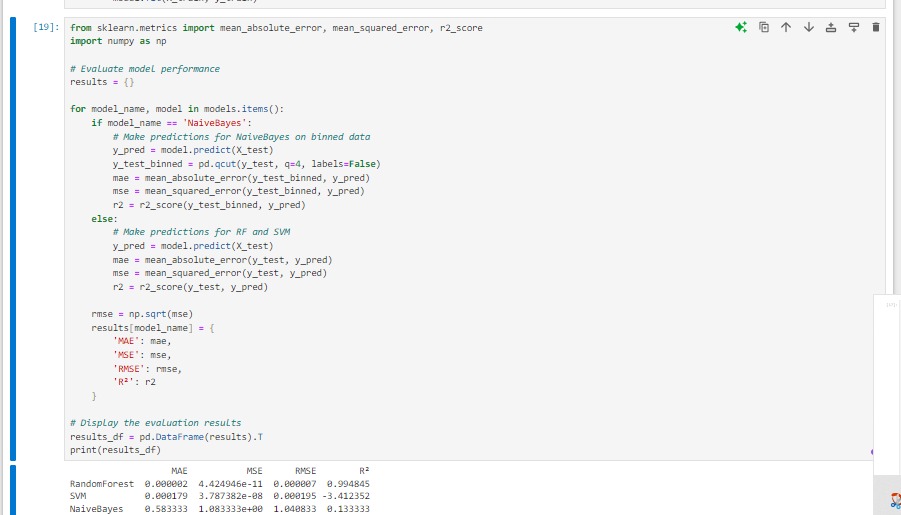
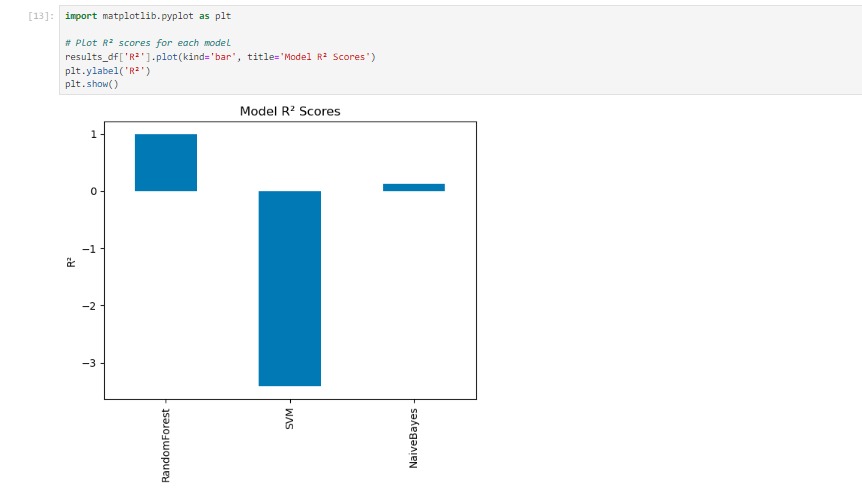
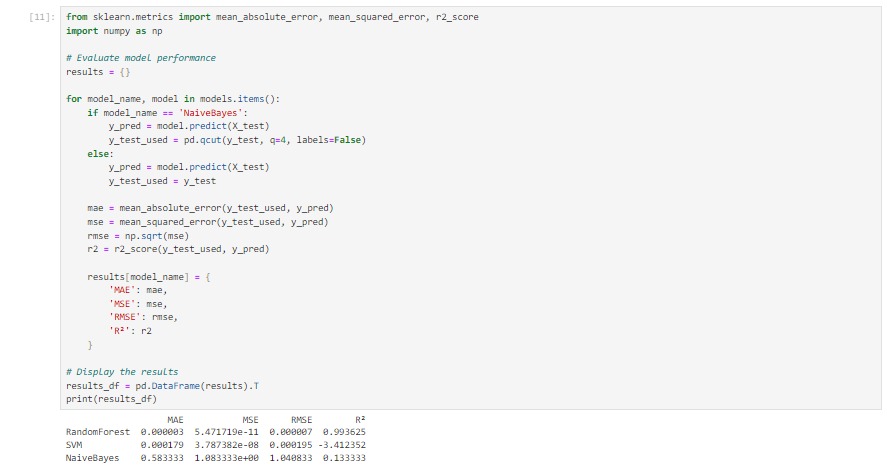
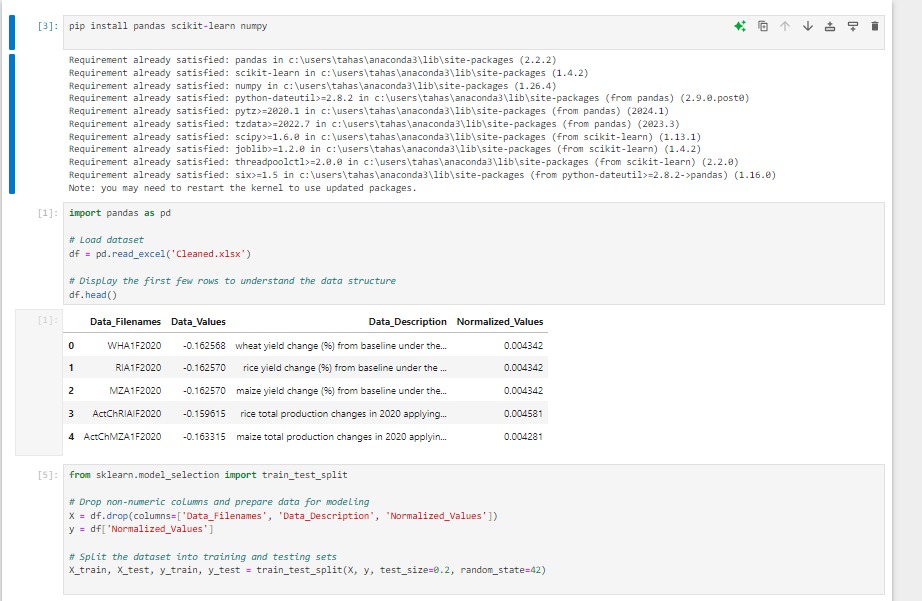
* Three machine learning models are defined:
* RandomForestRegressor: Used for predicting continuous values.
* SVR (Support Vector Regressor): Another regression model used to predict continuous values.

Naive Bayes: Typically used for classification tasks. In this case, the continuous target variable `y\_train` is binned into categories using `pd.qcut()` so that Naive Bayes can work with it.

**3. Model Training:**

The models are trained on the training data:

* For Naive Bayes: The target variable (`y\_train`) is binned into 4 discrete categories before training because Naive Bayes requires categorical targets.
* For Random Forest and SVR: These models are trained directly on the unmodified continuous data (`y\_train`).

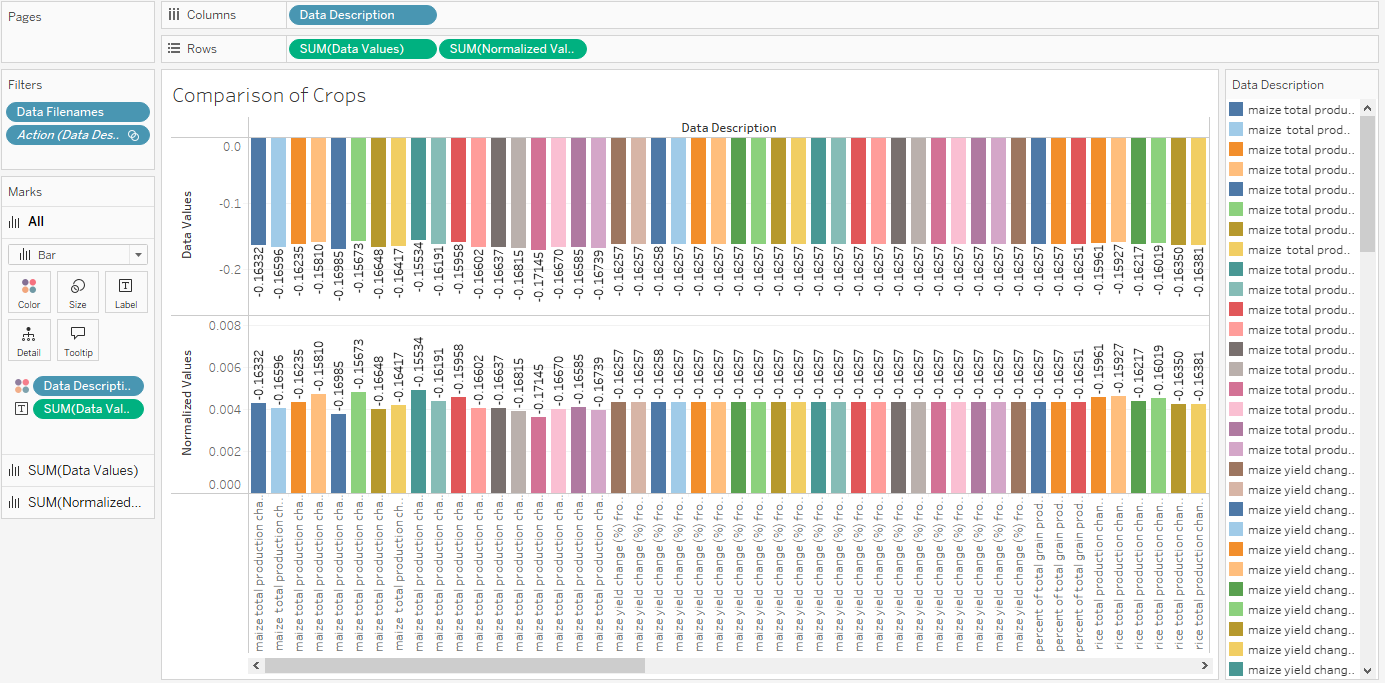


RandomForestRegressor and SVR predict the continuous target values

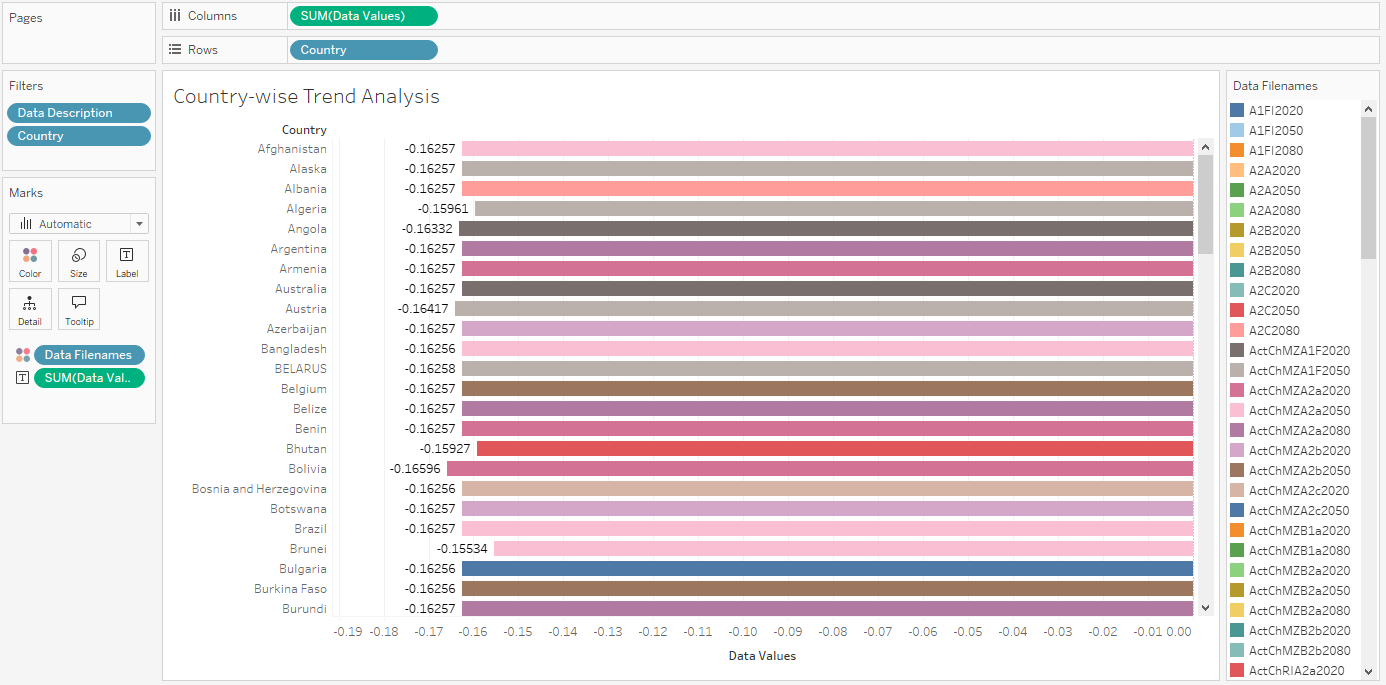
Naive Bayes requires a categorical target, so the continuous target is converted into categories by binning before training.

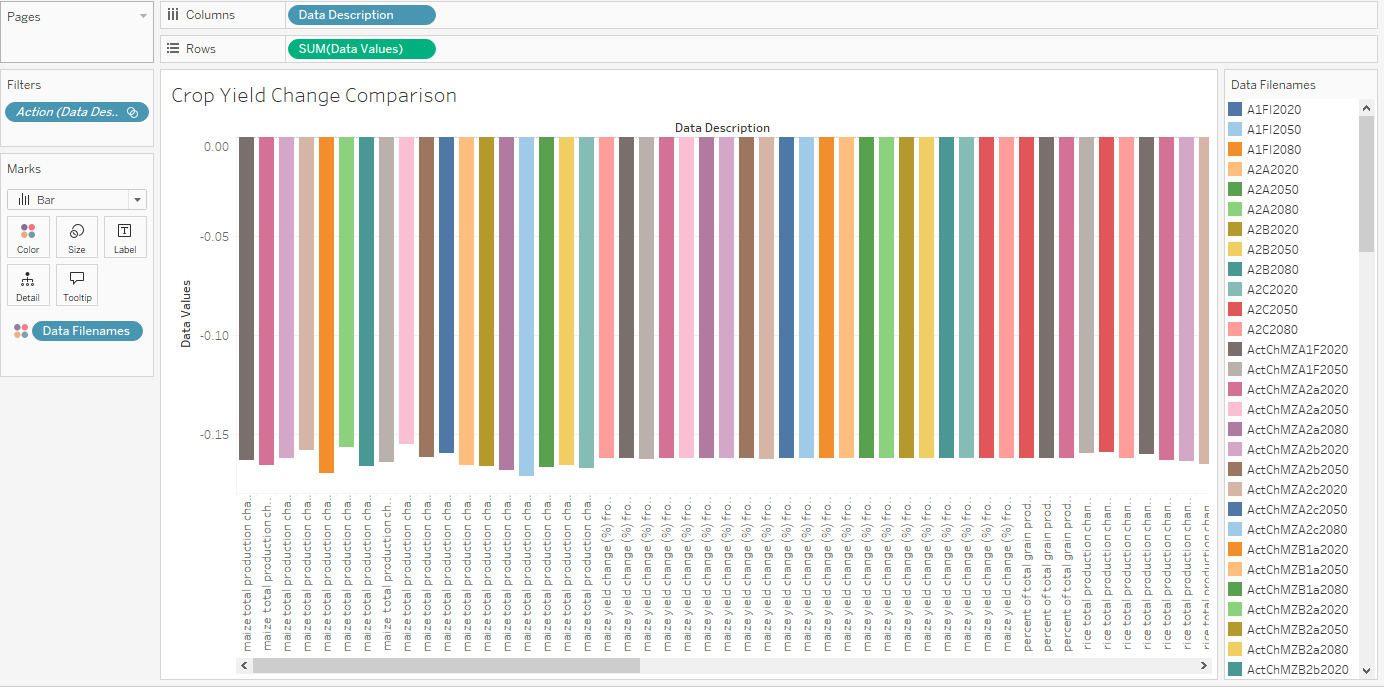
After training, the models can be evaluated or used to make predictions on the test data (`X\_test`).

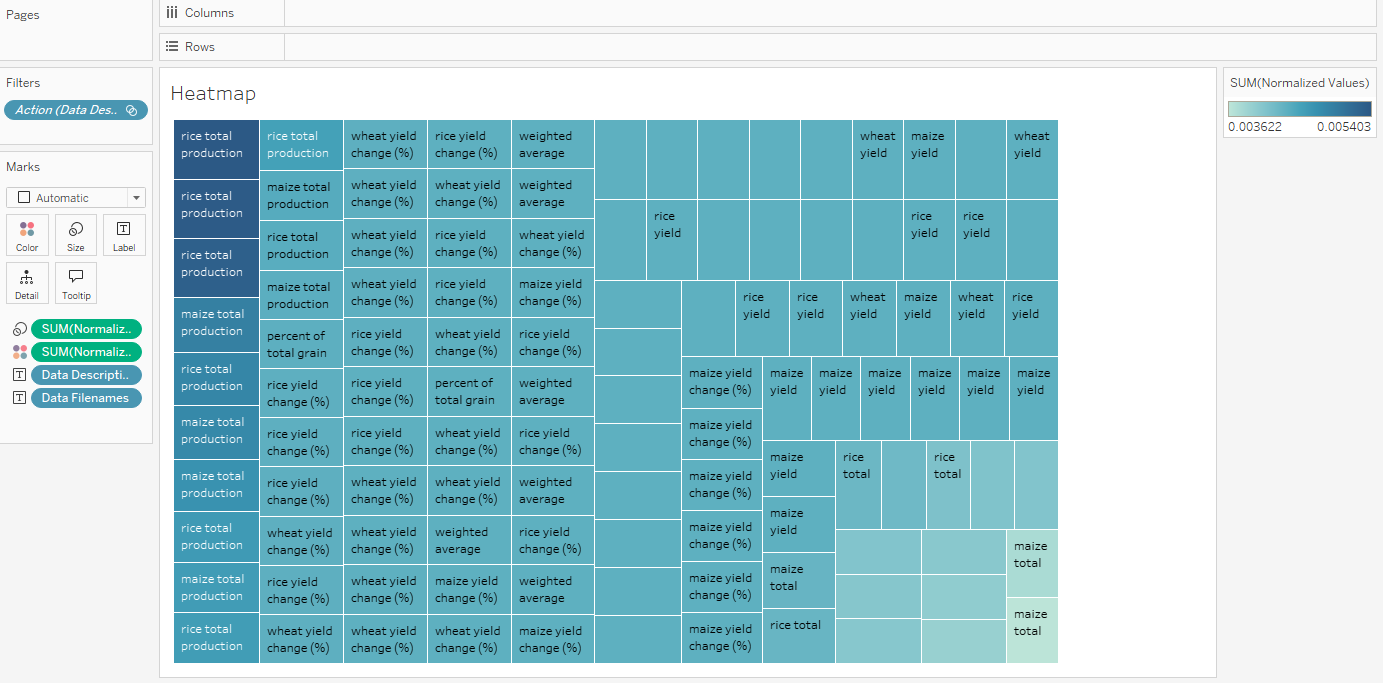
Tableau Designing Phase

****

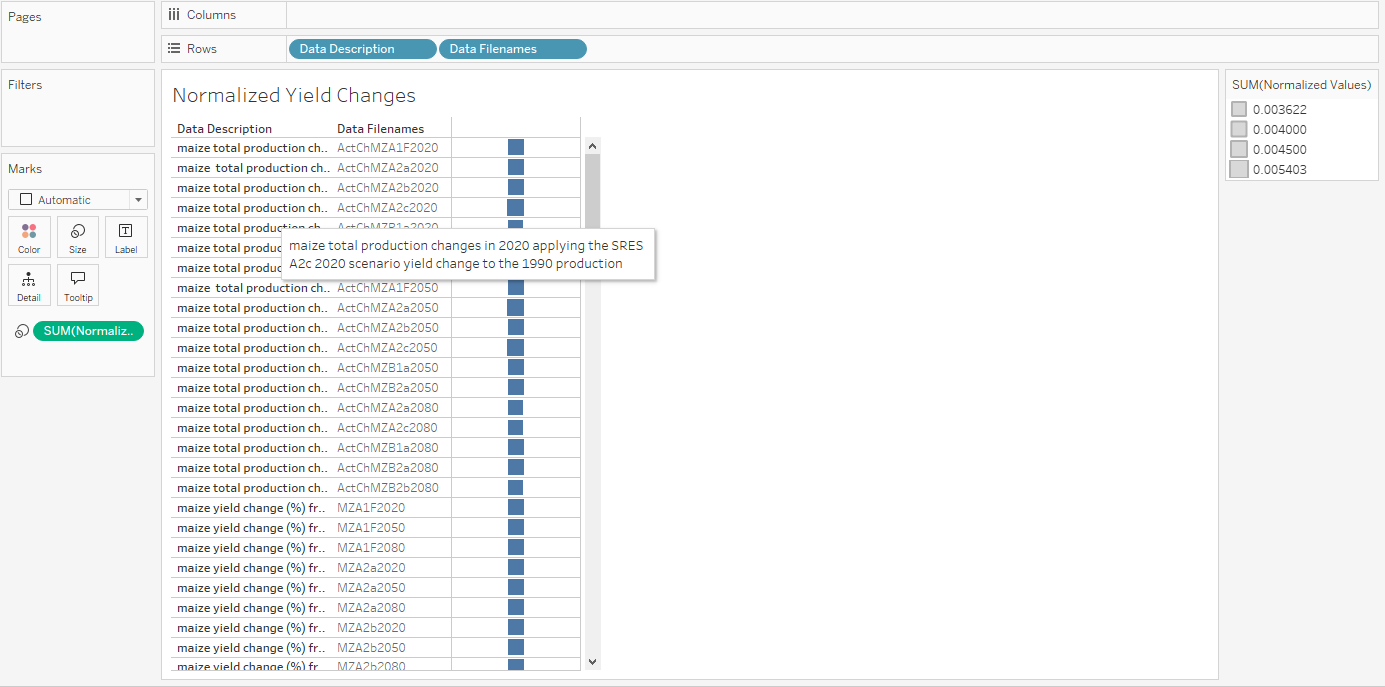
**Comparison of Crops**: This visualization likely compares different crop yields or performance metrics across various conditions or treatments.

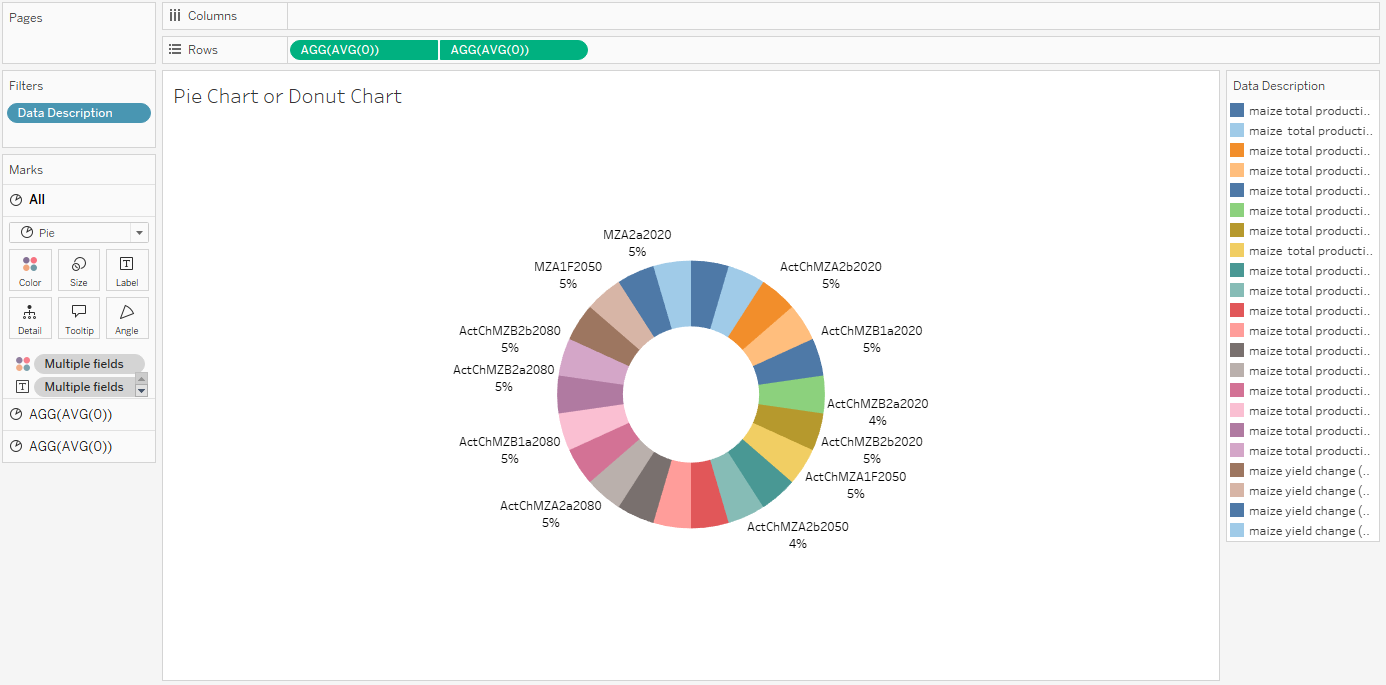
****

**Country Wise Trend Analysis**: This image is probably analyzing trends in crop yields, weather patterns, or environmental impacts on agriculture across different countries.****

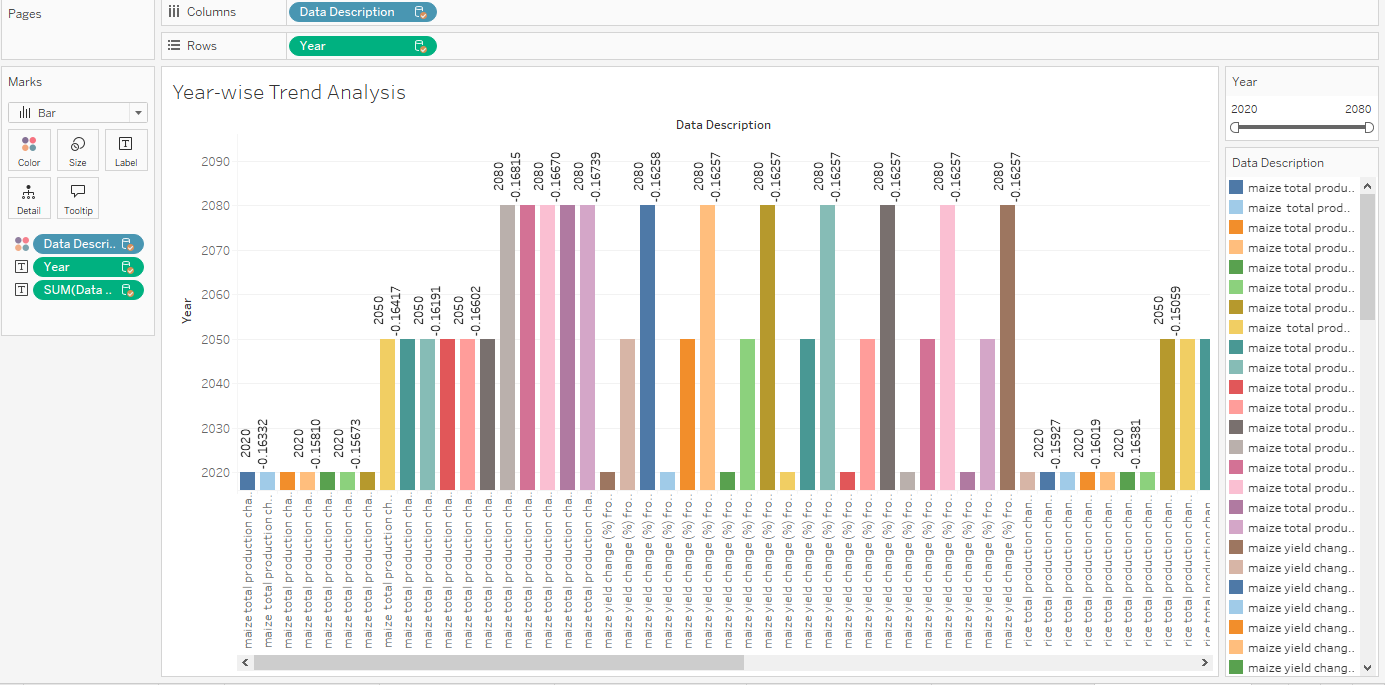
**Crop Yield Change Comparison**: This graph might show the changes in crop yields over time, possibly under different environmental stresses or management strategies.****

**Heatmap**: Typically used to represent data density or intensity in different regions, this could detail temperature variations, precipitation patterns, or other climatic factors affecting agriculture.

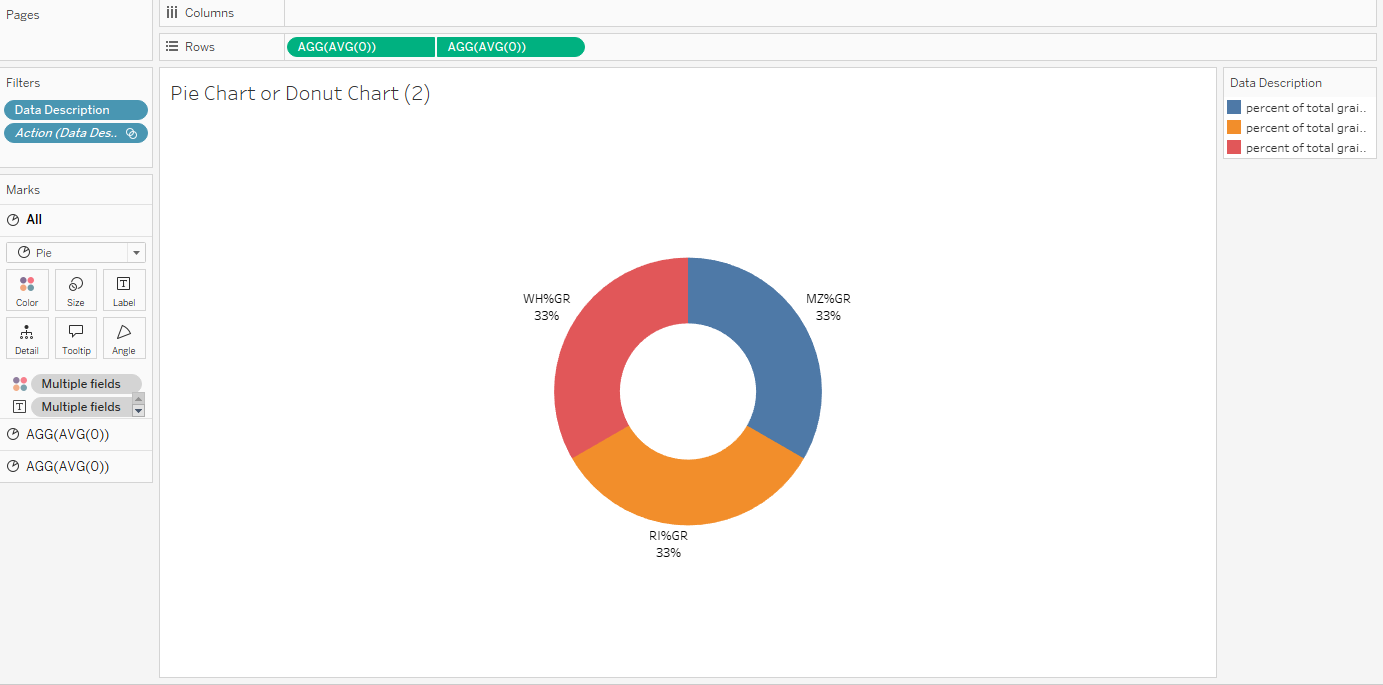
****

**Normalized Yield Changes**: This graph could depict yield changes normalized against a baseline, highlighting deviations or trends important in studying impacts of climate change or agronomic interventions.****

**Pie Chart or Donut Chart**: This visualization would break down percentages of different categories, such as types of crops, environmental impacts, or resource usage.

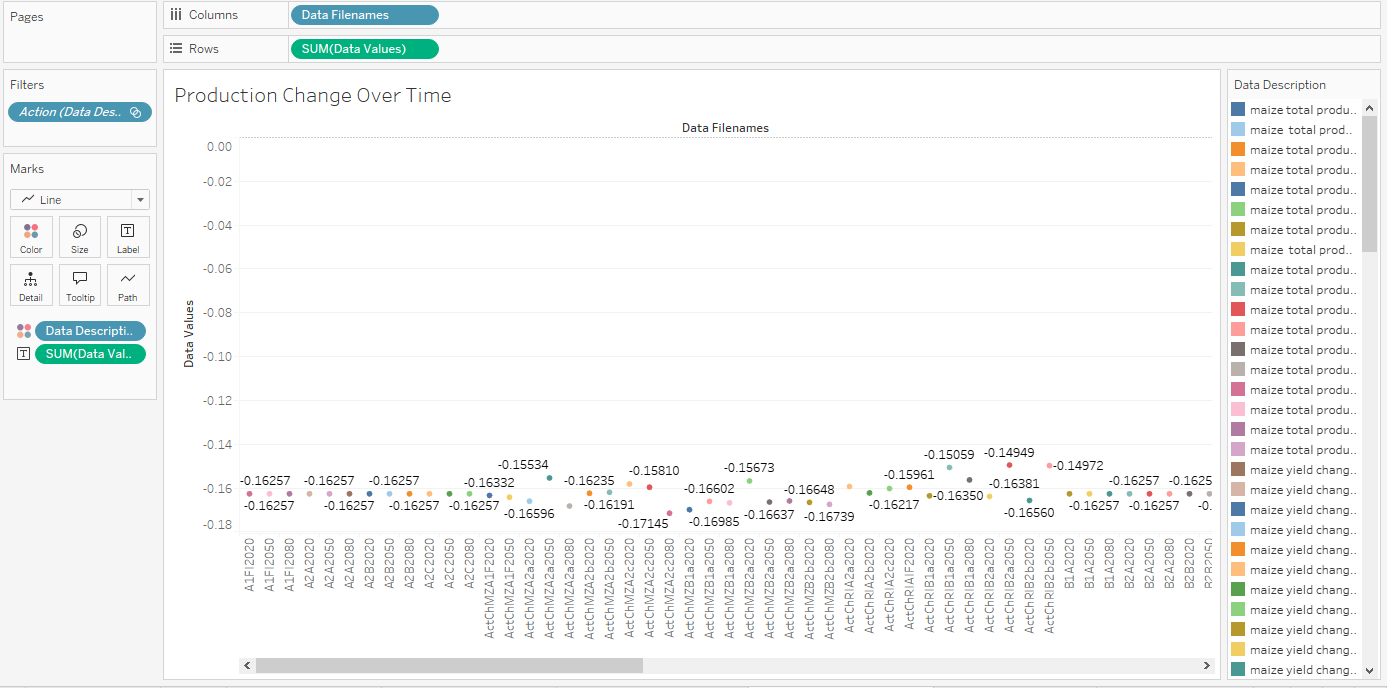
****

**Year Wise Trend Analysis**: This chart is likely tracking the progression of a particular agricultural or environmental metric over the years, showing trends that could be critical for long-term planning and strategy.

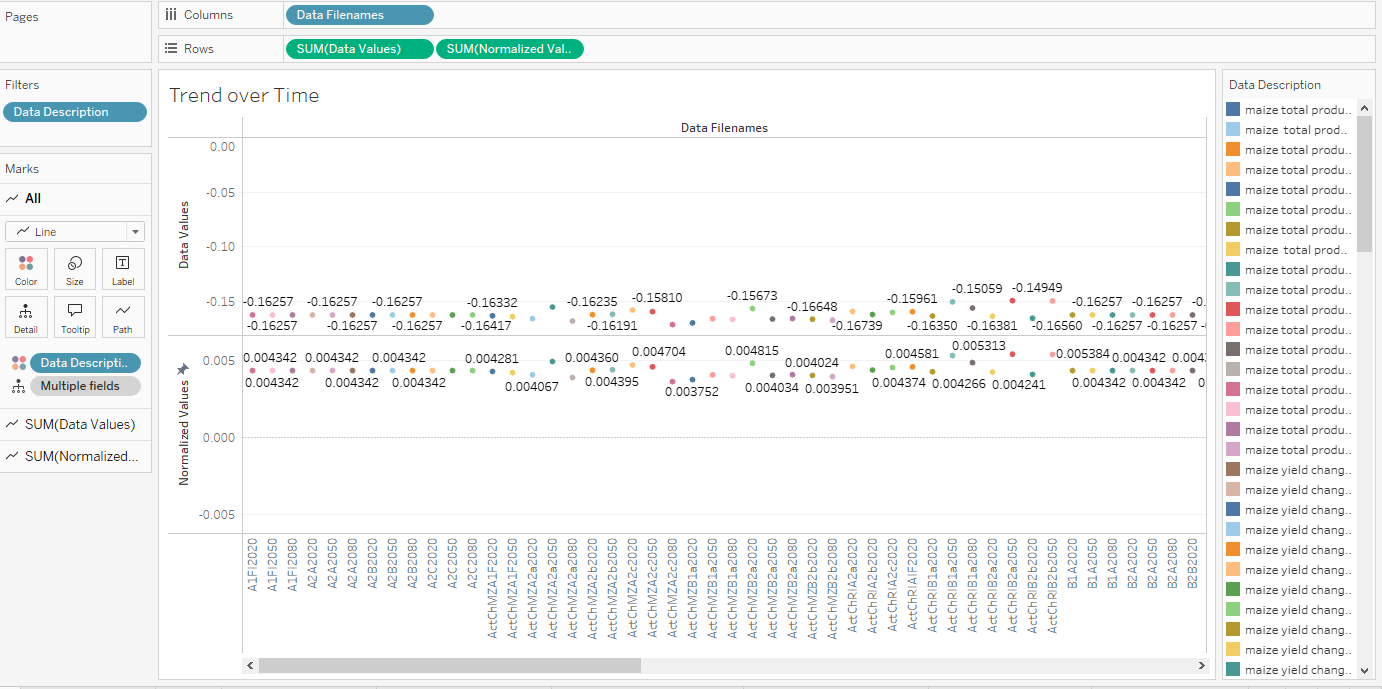
****

**Pie or Donut Chart**:

* This chart seems to represent three categories, labeled "WH%GR," "MZ%GR," and "RI%GR," with each sector accounting for 33%. It might depict the percentage distribution of different grain types, possibly "Wheat," "Maize," and "Rice," based on color legend and labels.
* The chart is likely showing the percentage of total production for each grain type.

 **Production Change Over Time (Line Chart)**:

* This chart visualizes the changes in production values over a period (likely years or specific time frames).
* The X-axis lists "Data Filenames," which could represent different scenarios or models over time.
* The Y-axis ("Data Values") shows the change in production (values are negative, which suggests a reduction in production).
* The color legend represents different types of grains or datasets related to grain production, with maize being the most notable.



**Trend Over Time (Line Chart)**:

* This chart appears similar to the previous one but with two sets of data visualized: "Data Values" and "Normalized Values."
* The top portion represents the same production changes over time as seen in the second chart.
* The lower portion ("Normalized Values") shows more subtle fluctuations, likely adjusted or normalized to remove variations and focus on trend insights.
* Each colored dot represents a specific time point for a different grain type or production model (based on the color legend).