Integrating Superset with National Platform for Crop Yield Forecasting

Nikunj Indoriya

Project Duration: August 2024 - October 2024

Institute of Data Analytics, Engineering and Science TIH, Indian Statistical Institute

Contents

List of Figures

List of Tables

Listings

1 Project Overview and Objective

In this project, we utilized Apache Superset to analyze and visualize crop production data sourced from the Ministry of Agriculture & Farmers Welfare, Department of Agriculture & Farmers Welfare (DAFW).Throughout the project, we developed two interactive dashboards to examine crop production metrics in Apache Superset, embedding them into an HTML page via iframe for accessible viewing.Additionaly, we integrated supplementary data from the *Agricultural Statistics at a Glance* report by the Ministry of Agriculture, Government of India, which includes the cost-price index for various crops.This allowed us to assess the relationship between the crop production volumes and their respective price indices. To effectively analyze these data points, we employed a range of visualizations, including bar charts, line charts, pie charts, and word cloud.

2 Dataset Overview

The dataset is provided by the Ministry of Agriculture Farmers Welfare, Department of Agriculture Farmers Welfare (DAFW) and includes time-series data on the production of various crops across different seasons, including Rabi (November to April), Kharif (June to November), and Summer, covering the years 2013-14 to 2022-23. The original dataset comprises production estimates for key crops such as rice, wheat, maize, and a range of pulses and oilseeds, reported in lakh tonnes. Additionally, it includes seasonal and annual totals, allowing for an in-depth examination of crop production trends. This data was preprocessed prior to analysis.

The dataset is organized into a series of columns that detail crop production estimates by year, season, and crop type. It contains several key columns, including *Year, Crop Type, Season*(Rabi, Kharif, and Summer), and *Production in Lakh Tonnes.* Each row represents a specific crop production metric for a given year and season. The seasonal and total production figures are reported for multiple crop categories, such as food grains, cereals, pulses, oilseeds, and commercial crops.

2.1 Preprocessing of the Data

The dataset was restructured from a horizontal layout to a vertical one, where instead of having individual rows for each crop, dedicated columns were created for *Crop, Season, Year, and Production.* This restructuring allows each row to represent a unique crop production record, detailing the crop name, season (Kharif, Rabi, or Summer), year, and production volume in lakh tonnes. Additionally, each crop was classified into broader categories such as *Cereals, Pulses, Oilseeds, and Cash Crops* based on the standards and definitions provided by the Department of Agriculture & Farmers Welfare (DA&FW). This classification enhances analytical capabilities by enabling category-specific analysis and trend identification across multiple crop types and seasons. A sample of the preprocessed dataset is shown in Table 1.

Crop	Season	Year	Production (in Lakh Tonnes)	Crop Type
Rice	Kharif	2013	914.97	Cereals
Rice	Rabi	2013	151.49	Cereals
Rice	Summer	2013	0.00	Cereals
Rice	Kharif	2014	913.92	Cereals
Rice	Rabi	2014	140.91	Cereals

Table 1: Sample Preprocessed Data

Moreover, additional data was taken from the *Agricultural Statistics at a Glance* report by the Ministry of Agriculture, Government of India, consisting of the Cost Price Index disaggregated by *Crop Type*, specifically for categories such as Sugar and Confectionery, Pulses and Products, Oils and Fats, and Cereals and Products, covering the years 2013 to 2017.This data is present in the Table 2.

Year	Sugar and Confectionary	Pulses and Products	Oils and Fats	Cereals and Products
2013	102.1	107.5	105.6	116.6
2014	101.7	116.0	108.1	122.6
2015	94.5	153.0	112.7	124.9
2016	113.0	167.2	117.3	130.2
2017	119.9	132.1	119.2	134.7

Table 2: Cost Price Index by Product Category (2013-2017)

3 Installation of Superset and SQL Workbench Using Docker

- Visit Docker installation documentation for Windows Link to document.
- Download the MySQL Installer from the Official Site of MySQL.

3.1 Installation of MySQL Server

- Download the MySQL Installer for Windows. Choose the appropriate version (e.g., MySQL Installer Community).
- Launch the MySQL Installer executable.
- Follow the setup wizard:
  - **Choose Setup Type:** Select "Server only" if you only want the server.
  - **Check Requirements:** The installer will check if any required components are missing.Install any prerequisites if prompted.
  - **Select Products and Features:** Ensure that MySQL Server is selected along with any other tools you need (e.g., MySQL Workbench).
  - **Configure MySQL Server:**
    - **Configuration Type:** Choose "Standalone MySQL Server".
    - **Connectivity:** Set the port (default is 3306). Ensure it's open if using firewall.
    - **Authentication Method:** Select the authentication method (default is "Use Legacy Authentication Method").
    - **Set Root Password:** Enter and confirm a root password. This will be used for administrative access.
    - **Create a User Account (optional):** You can create additional user accounts with specific privileges.
  - **Apply Configuration:** Review the configuration and click "Execute" to apply the settings.
- **Connect to MySQL:** Open a command prompt and connect to MySQL using mysql command line.

```
mysql -u root -p
```

Listing 1: Running MySQL

3.2 Installation of Superset

- Install Docker desktop using the downloaded Docker Desktop Installer.
- Search for superset in the Docker desktop and pull the latest version available to run it as a container.
- Note the latest tag and use in the below provided code.
- Open the command prompt and use the below provided codes to load the superset in local machine.

```
docker run -d -p 8080:8088 -e "SUPERSET_SECRET_KEY=mysuperset" --name
    superset apache/superset:latest
docker exec -it superset superset fab create-admin --username admin
    --firstname Superset --lastname Admin --email admin@superset.com
    --password admin
docker exec -it superset superset db upgrade
docker exec -it superset superset\load_examples
```

Listing 2: Superset Activation and Installation

4 Dashboard Creation and design

4.1 Data Import and setup

The Apache superset was integrated to MySQL to import the dataset using the following command after opening the command prompt

```
docker ps  #(to figure out the name of the container)
docker exec -it superset bash
pip install mysqlclient #(inside the container)
```

Listing 3: Integration of SQL to Superset

Then open the Superset and add a Database using SQLAlchemyURL format which is given below:

```
mysql://mysql_ username:mysql_password@host.docker.internal:3306/superset_db
```

Listing 4: SQLAlchemy URL

Then test and connect.

Using MySQL Command Line Client, import the reshaped data into MySQL which can be used in the Superset later to create the charts. To add the column of Crop Type we used a special SQL Query:

```
UPDATE reshaped_data
SET crop_type =
  CASE
    WHEN crop IN ('Cotton', 'Sugarcane', 'Jute', 'Mesta') THEN 'Cash Crops'
    WHEN crop IN ('Rice', 'Wheat', 'Maize', 'Barley', 'Bajra', 'Jowar',
        'Ragi', 'Small Millets', 'Shree Anna', 'Nutri/Coarse Cereals',
        'Cereals') THEN 'Cereals'
    WHEN crop IN ('Pulses', 'Gram', 'Peas', 'Tur (Arhar)', 'Gram (Chana)',
        'Urad', 'Moong', 'Lentil (Masoor)', 'Other Pulses', 'Total Pulses')
        THEN 'Pulses'
    WHEN crop IN ('Soybean', 'Groundnut', 'Sunflower', 'Mustard', 'Sesamum',
        'Rapeseed & Mustard', 'Linseed', 'Safflower', 'Castorseed',
        'Nigerseed') THEN 'Oilseeds'
    WHEN crop IN ('Tea', 'Coffee') THEN 'Plantation Crops'
    ELSE 'Unknown'
  END;
```

Listing 5: Creation of Crop Type Column

4.2 Dashboard Design

The dashboard consists of two main tabs **Crop Overview** and **Exploratory Analysis**.

Crop Overview

This tab provides an insightful analysis of crop production trends over the years, giving users a clear view of how production has shifted and evolved over time. It breaks down the data seasonally, highlighting production during the Rabi, Kharif, and Summer seasons. This detailed approach allows users to explore how each season contributes to the overall crop production and observe any patterns or fluctuations within those seasons. By examining both the long-term trends and seasonal variations, users can better understand the dynamics of crop production and its impact on annual totals.

Exploratory Analysis

This tab delves into crop production trends over years and seasons. It tries to answer the below questions using different charts as mentioned:

1. What is total amount of production for each year by crop type?
2. Production trendline which shows crop production by crop type with year.
3. Word Cloud which show which crop is produce more with bigger size.
4. What are the top 5 crops with the highest production for each year?
5. Which are top 3 years with the highest production for each crop?

Filters and Interactivity

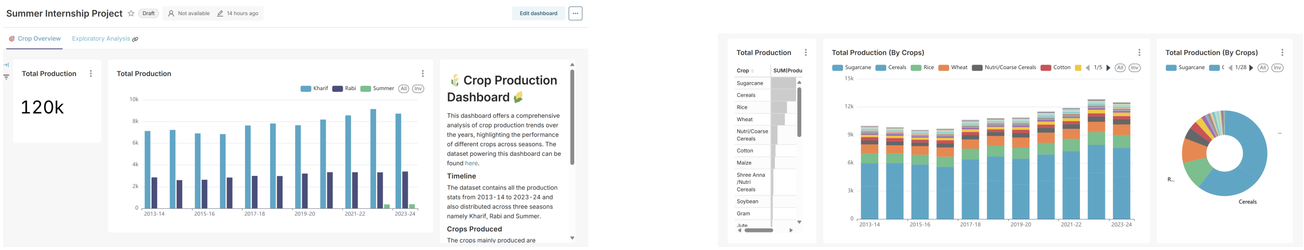
A filter was applied on the charts within the *Exploratory Overview* dashboard, enabling users to select a specific crop type for focused analysis. This interactivity allows users to customize the view according to their analytical needs, providing detailed insights into individual crop trends.
Additionally, a comprehensive filter is applied across all charts in the ”Crop” Overview” dashboard, allowing users to select specific crops, years, and seasons. This level of filtering empowers users to conduct targeted analyses on crop production patterns across different time periods and growing seasons, enhancing the flexibility and depth of insights that can be derived from the data.

List of Visualization

A comprehensive table of all charts used across the dashboards, including their visualization names, chart types, and associated dashboards, is provided in Table 3 on the following page. Moreover, screenshots of the dashboard are also attached.

Visualization Name	Chart Type	Dashboard
Gross Production	Big Number	Crop Overview
Total Production (by Season)	Bar Chart	Crop Overview
Total Production (by Crops)	Table	Crop Overview
Total Production (by Crops)	Bar Chart	Crop Overview
Total Production (by Crops)	Pie Chart	Crop Overview
Total amount of production (by Crop Type)	Bar Chart	Exploratory Analysis
Crops Produced	Word Cloud	Exploratory Analysis
Production Trendline	Area Chart	Exploratory Analysis
Top 5 crops each year (Production)	Bar Chart	Exploratory Analysis
Top 3 years for each crop (Production)	Bar Chart	Exploratory Analysis

Table 3: List of Visualisations



Crop Production Dashboard

This dashboard offers a comprehensive analysis of crop production trends over the years, highlighting the performance of different crops across seasons. The dataset powering this dashboard can be found here.

Timeline

The dataset contains all the production stats from 2013-14 to 2023-24 and also distributed across three seasons namely Kharif, Rabi and Summer.

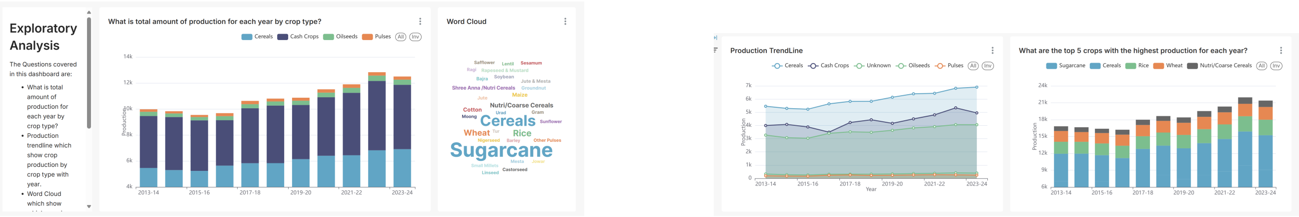
Crops Produced

The crops mainly produced are

(a) Overview Dashboard

(b) Total Production Visualisations

Figure 1: Overview Dashboard



Exploratory Analysis

The Questions covered in this dashboard are:

- What is total amount of production for each year by crop type?
- Production trendline which show crop production by crop type with year.
- Word Cloud which show

(a) Exploratory Analysis Dashboard

(b) Production Trendline and Top 5 Crops

Figure 2: Exploratory Analysis

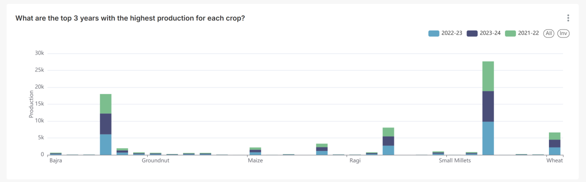


Figure 3: Top 3 years for each crop for highest production

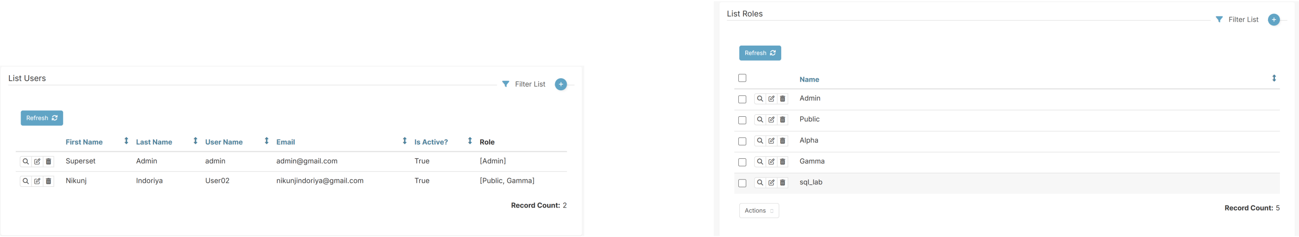
4.3 Role-Based Access

In our project, we configured a custom role called **Public** which inherits all permissions from the Gamma role and additional access specifically to the "Summer Internship" dashboard. Altogether, five roles were set up in the project: **Admin, Gamma, Alpha, SQL Lab, and Public.**

Two users were created in the system:

- The default **admin** user with full Admin access.
- A secondary user with roles similar to **Gamma** and shared ownership of the Summer Internship dashboard.

The relevant screenshots are attached herewith for reference.



List Users

	First Name	Last Name	User Name	Email	Is Active?	Role
	Superset	Admin	admin	admin@gmail.com	True	[Admin]
	Nikunj	Indoriya	User02	nikunjindoriya@gmail.com	True	[Public, Gamma]

Record Count: 2

(a) List of Users

List Roles

□	Name
□	Admin
□	Public
□	Alpha
□	Gamma
□	sql_lab

Record Count: 5

(b) List of Roles

Figure 4: List of Roles and Users

4.4 Dashboard Embedding Using iframe

- To embed the dashboard, we used the HTML iframe tag.
  Before embedding, a configuration file, superset_config.py was created which includes the required configurations and is shown below:

```
FEATURE_FLAG = {
    "EMBEDDED_SUPERSET": True,
}
ENABLE_PROXY_FIX = True
SESSION_COOKIE_SAMESITE = None

PUBLIC_ROLE_LIKE_GAMMA = True
#AUTH_ROLE_PUBLIC = 'GAMMA'

WTF_CSRF_ENABLED = False
TALISMAN_ENABLED = False

HTTP_HEADERS = {'X-Frame-Options': 'ALLOWALL'}
```

Listing 6: superset_config.py

- Mount the superset_config.py in Docker using the following command:

```
        docker cp superset_config.py
           superset:/app/pythonpath/superset_config.py
```

- To verify if it has been copied:

```
        docker exec -it superset /bin/bash
        cd/app/pythonpath/
        ls
        which typically enforcesexit
```

- Restart the superset using docker. This superset_config.py file is configured to allow the embedding of Superset dashboards in external applications by modifying security settings and access controls. Each setting in this configuration file serves a specific purposes, detailed below:

1. **FEATURE_FLAGS:** This dictionary contains feature toggles for Superset. Setting ‘”EMBEDDED_SUPERSET”: True‘ enables embedding of Superset dashboards or views within external applications or web pages.
2. **ENABLE_PROXY_FIX:** Setting this to **True** adjusts Superset to handle headers correctly when deployed behind a proxy or load balancer, ensuring that headers like the original IP address are accurately recognized.
3. **SESSION_COOKIE_SAMESITE:** Setting this to **None** removes the SameSite attribute from session cookies, allowing cross-site requests, which is often required when embedding Superset in other applications.
4. **PUBLIC_ROLE_LIKE_GAMMA:** Setting this to **True** grants anonymous (public) users the same permissions as those in the Gamma role, which typically includes basic viewing and dashboard access. This setting is helpful for embedding dashboards without requiring login.
5. **WTF_CSRF_ENABLED:** Setting this to **False** disables CSRF (Cross-Site Request Forgery) protection, which can simplify embedding Superset in other applications, though it reduces certain security protections.
6. **TALISMAN_ENABLED:** Setting this to **False** disables Talisman, which typically enforces strict HTTP security headers. Disabling it allows easier embedding but reduces protections such as clickjacking prevention.
7. **HTTP_HEADERS:** This permits the Superset instance to be embedded in iframes by any site. This setting is essential for embedding Superset dashboards but should be used with caution, as it may expose the application to clickjacking risks if not handled carefully.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>IDEAS Internship Project Dashboard</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f9f9f9;
            display: flex;
            justify-content: center;
            align-items: center;
            flex-direction: column;
        }
        .container {
            max-width: 90%;
            padding: 20px;
            background-color: #ffffff;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
            margin-top: 40px;
            text-align: center;
            border-radius: 8px;
        }
        h1 {
            color: #004080;
            margin-bottom: 20px;
        }
        p.description {
            color: #555;
            font-size: 1.1em;
            margin: 0 0 20px;
        }
        iframe {
            width: 100%;
            height: 800px;
            border: none;
            border-radius: 8px;
            margin-bottom: 20px;
        }
        footer {
            margin-top: 20px;
            font-size: 0.9em;
            color: #888;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>IDEAS Internship Project: Integrating Superset with NPCYF</h1>
        <p class="description">Welcome to my project dashboard! This
           interface showcases
 the integration of Apache Superset to visualize and analyze data for
     NPCYF.</p>
        <iframe src="http://localhost:8080/superset/dashboard/p/o7rLWk4Ln0E/">
         </iframe>
        <footer>
            <p>Created by Nikunj Indoriya | <a
                href="https://www.ideas-tih.org/">Institute of Data
                Engineering, Analytics and Sciences Foundation</a>
                Internship, October, 2024
            </p>
                </footer>
            </div>
</body>
</html>
```

Listing 7: HTML Code for embedding Superset dashboard

- The above given HTML code was used to embed the Apache Superset dashboard for the Project. It includes an iframe tag to display the dashboards and some basic styling for the page layout.

5 Conclusion and Future Exploration

This project effectively demonstrates the integration of Apache Superset for visualizing and analyzing crop production data, using MySQL as the backend and implementing secure, role-based user access. Through embedded dashboards, it provides accessible and interactive insights into crop production trends and price correlations. Future developments could explore enhancing data sources, expanding analytical capabilities, and refining the user experience with advanced filtering and customization options.

6 Appendix

SQL Queries Used to Build Charts

The SQL Queries used to build some of the charts are presented below:

1. **What is total amount of production for each year by crop type?**

```sql
SELECT `Year` AS `Year`, crop_type AS crop_type, sum(`Production`) AS
    `SUM(Production)`
FROM mysql.reshaped_data
WHERE `Season` != 'Total' AND crop_type != 'Unknown' GROUP BY `Year`,
    crop_type ORDER BY `SUM(Production)` DESC
 LIMIT 10000;
```

Listing 8: What is total amount of production for each year by crop type?

2. **Production TrendLine**

```sql
SELECT `Year` AS `Year`, crop_type AS crop_type, sum(`Production`) AS
    `SUM(Production)`
FROM mysql.reshaped_data
WHERE `Season` != 'Total' GROUP BY `Year`, crop_type ORDER BY
    `SUM(Production)` DESC
 LIMIT 10000;
```

Listing 9: Production Trendline

3. **What are the top 5 crops with the highest production for each year?**

```sql
SELECT `Year` AS `Year`, `Crop` AS `Crop`, sum(`Production`) AS
    `SUM(Production)`
FROM mysql.reshaped_data INNER JOIN (SELECT `Crop` AS `Crop__`,
    sum(`Production`) AS mme_inner__
FROM mysql.reshaped_data
WHERE `Crop` != 'Total Pulses' AND `Crop` != 'Total Food Grains' AND `Crop`
    != 'Total Oil Seeds' GROUP BY `Crop` ORDER BY mme_inner__ DESC
 LIMIT 5) AS series_limit ON `Crop` = `Crop__`
WHERE `Crop` != 'Total Pulses' AND `Crop` != 'Total Food Grains' AND `Crop`
    != 'Total Oil Seeds' GROUP BY `Year`, `Crop` ORDER BY `SUM(Production)`
    DESC
 LIMIT 10000;
```

Listing 10: What are the top 5 crops with the highest production for each year?

4. **What are the top 3 years with the highest production for each crop?**

```
SELECT `Crop` AS `Crop`, `Year` AS `Year`, sum(`Production`) AS
    `SUM(Production)`
FROM mysql.reshaped_data INNER JOIN (SELECT `Year` AS `Year__`,
    sum(`Production`) AS mme_inner__
FROM mysql.reshaped_data
WHERE `Crop` != 'Total Pulses' AND `Crop` != 'Total Food Grains' AND `Crop`
    != 'Total Oil Seeds' GROUP BY `Year` ORDER BY mme_inner__ DESC
 LIMIT 3) AS series_limit ON `Year` = `Year__`
WHERE `Crop` != 'Total Pulses' AND `Crop` != 'Total Food Grains' AND `Crop`
    != 'Total Oil Seeds' GROUP BY `Crop`, `Year` ORDER BY `SUM(Production)`
    DESC
 LIMIT 10000;
```

Listing 11: What are the top 3 years with the highest production for each crop?

5. **Total Production (By Crops)**

```
SELECT `Year` AS `Year`, `Crop` AS `Crop`, sum(`Production`) AS
    `SUM(Production)`
FROM mysql.reshaped_data
WHERE `Season` != 'Total' AND crop_type != 'Unknown' GROUP BY `Year`, `Crop`
    ORDER BY `SUM(Production)` DESC
 LIMIT 10000;
```

Listing 12: Total Production (By Crops)

Github Repository