Integrating Superset using NPCYF

Yash Burman

October 2024

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 (DA&FW). 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. Additionally, 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 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 time-series percentage change line charts.

2 Dataset overview

The dataset is provided by the Ministry of Agriculture & Farmers Welfare, Department of Agriculture & Farmers Welfare (DA&FW) 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 present in Table 1

Table 1: Sample Preprocessed Data

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

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 presented in Table 2.

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

Year	Sugar and Confectionery	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

3 Installation of Superset and SQL Workbench Using Docker

- Visit Docker installation documentation for Ubuntu: Link to Document.
- Add user to the docker group using:

```
sudo usermod -aG docker $USER
```

- We will use ‘docker-compose‘ to run different components, i.e., MySQL server, MySQL Workbench, and Superset.

Installation of MySQL Server

- As we are using ‘docker-compose‘ to run MySQL Server, we need a ‘docker-compose.yaml‘ file to set up the MySQL container.
- Visit the DockerHub documentation for MySQL Server: Link to Document.
- Locate the ”via docker-compose or docker stack deploy” section in the documentation.
- Create a directory on your local machine:

```
mkdir -p ~/projects/mysql-server
```

- Copy the ‘docker-compose.yaml‘ configuration and save it in ‘ /projects/mysql-server/docker-compose.yaml‘ using a text editor like nano.
- Add port and volume bindings in ‘docker-compose.yaml‘ for database access and data persistence.
- Run the following command to launch the MySQL server container:

```
docker compose up -d
```

Installation of MySQL Workbench

- Similar to the MySQL server setup, we need a ‘docker-compose.yaml‘ file to run MySQL Workbench.
- Visit the DockerHub documentation for MySQL Workbench: Link to Document.
- Find the ”Usage” section in the documentation.
- Create a directory on your local machine:

```
mkdir -p ~/projects/mysql-workbench
```

- Copy the ‘docker-compose.yaml‘ configuration and save it in ‘ /projects/mysql-workbench/docker-compose.yaml‘.

- Add port and volume bindings in ‘docker-compose.yaml‘ for access and data persistence.
- Run the following command to launch the MySQL Workbench container:

```
    docker compose up -d

```

Installation of Superset

- A ‘docker-compose.yaml‘ file is also needed to run Superset using Docker.
- Visit the documentation for Superset installation: Link to Document.
- Create a directory on your local machine:

```
    mkdir -p ~/projects/superset

```

- Follow the setup steps provided in the documentation.
- Run the following command to start the Superset container:

```
    docker compose -f docker-compose-image-tag.yml up -d

```

4 Dashboard Creation and design

4.1 Data Import and setup

he data preprocessing was initially performed in LibreOffice Calc, where the raw data was cleaned and prepared before being exported as a `.csv` file. Next, in SQL Workbench, a new connection titled `Crop_Production_Database` was created specifically for this project. Within this connection, a schema named `crop_production_data` was established. The datasets referenced in Section 2, namely the production data and price data, were then imported as tables named `Crop_price_data` and `crop_production_data`.

Once the data import was completed in SQL Workbench, a database connection was created in Apache Superset to connect to the SQL Workbench instance. The datasets were then successfully imported into Apache Superset. In the `crop_production_data` dataset, two calculated columns—`Crop_Type` and `Year`—were created directly in Superset.

To enable time-series analysis and the construction of line charts, the `Year` column, originally of integer type, was converted to datetime format. The SQL code snippets below detail the transformations applied.

```
CAST(CONCAT(CAST(Year AS CHAR), '-01-01') AS DATE)
```

Listing 1: Converting `Year` to a Datetime 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 2: Constructing the `Crop_Type` Column

4.2 Dashboard Design

The dashboard consists of two main tabs: **Overall Crop Production and Season-wise Analysis** and **Analysis by Crop Type**.

Overall Crop Production and Season-wise Analysis

This tab analyzes the overall trend in crop production across years, providing a comprehensive view of how crop production has changed over time. Additionally, it includes a season-wise breakdown, with production data segmented by the Rabi, Kharif, and Summer seasons. This approach allows users to observe production patterns for each season and assess the contribution of each to the annual production totals.

Analysis by Crop Type

The *Analysis by Crop Type* tab delves into production trends at the `Crop_Type` level. This dashboard includes a filter that allows users to select specific crop types for detailed analysis. Additionally, it explores the relationship between the Consumer Price Index (CPI) for product categories within each crop type and the corresponding production data, offering insights into how price changes might influence production trends.

Filters and Interactivity

A filter was applied on the charts within the *Analysis by Crop Type* 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 "Overall Crop Production and Season-wise Analysis" 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 Visualizations

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

Table 3: List of Visualizations by Dashboard

Visualization Name	**Chart Type**	**Dashboard**	**Dataset Used**
Gross Production in 2024	Big Number	Overall Crop Production and Season-wise Analysis	crop_production_data
Production Trends by Season	Area Chart	Overall Crop Production and Season-wise Analysis	crop_production_data
Top 5 Crops by Production	Treemap	Overall Crop Production and Season-wise Analysis	crop_production_data
Total Production for Each Crop Category	Stacked Bar Chart	Overall Crop Production and Season-wise Analysis	crop_production_data
Seasonal Production Breakdown by Crop Category	Bar Chart	Overall Crop Production and Season-wise Analysis	crop_production_data
Production Trend	Line Chart	Overall Crop Production and Season-wise Analysis	crop_production_data
Production Trend by Season	Line Chart	Overall Crop Production and Season-wise Analysis	crop_production_data
Crop Production by Type	Pie Chart	Analysis by Crop Type	crop_production_data
Trend of Crop Production by Crop Type Over Time	Area Chart	Analysis by Crop Type	crop_production_data
Percentage Change in Sugar Production and CPI for Sugar and Confectionery Products	Time Series Percent Change	Analysis by Crop Type	crop_production_data, Crop_price_data
Percentage Change in Pulses Production and CPI for Pulses and Related Products	Time Series Percent Change	Analysis by Crop Type	crop_production_data, Crop_price_data
Percentage Change in Oilseed Production and CPI for Oilseed Products	Time Series Percent Change	Analysis by Crop Type	crop_production_data, Crop_price_data
Percentage Change in Cereal Production and CPI for Cereal Products	Time Series Percent Change	Analysis by Crop Type	crop_production_data, Crop_price_data

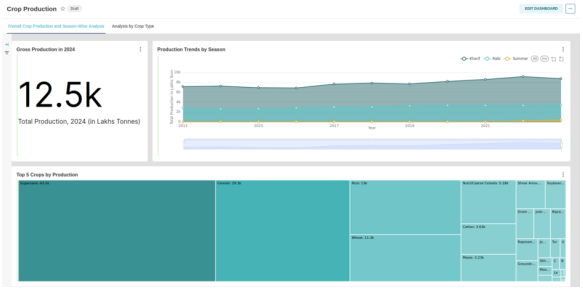

Figure 1: Overview of Crop Production Trends

Figure 2: Season-wise Crop Production Analysis

Figure 3: Crop Type Analysis Overview

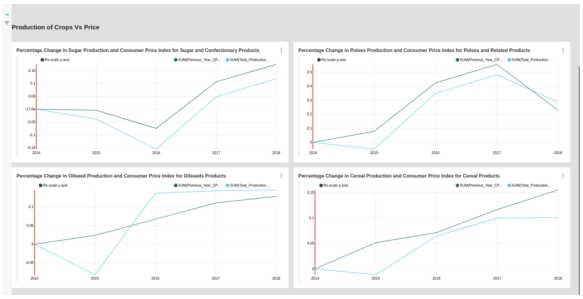

Figure 4: Price and Production Correlation by Crop Type

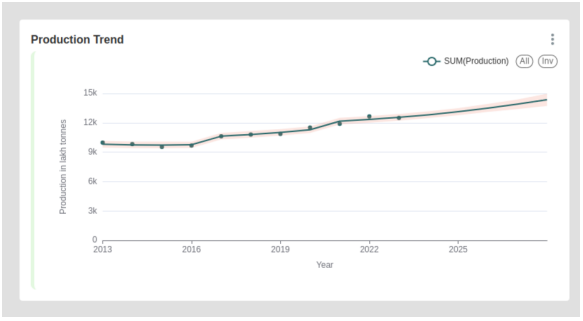

Figure 5: Production Trend by Crop

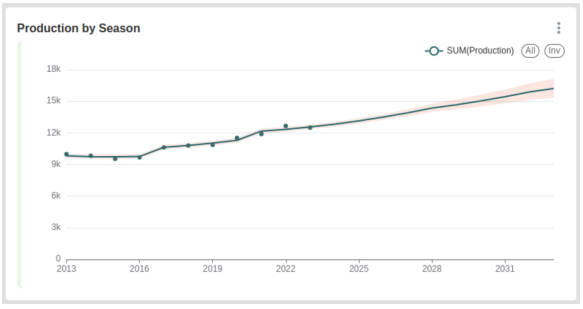

Figure 6: Production Trend by Season

4.3 Role-Based Access

In our project, we created a custom role named **Public**, which inherited all permissions from the Gamma role with additional access specifically to the "Crop Production" dashboard. In total, there were five roles configured in the project: **Admin**, **Gamma**, **Alpha**, **sql_lab**, and **Public**.

Two users were set up in the system:

1. The default **admin** user with full Admin access.
2. A secondary user with roles like **Gamma** named **Junior** and shared ownership of the `Crop_production` dashboard.

The relevant screenshots showing role configurations and user assignments are provided below.

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 and placed in the path `superset/docker/pythonpath_dev` to enable the necessary settings for embedding.

The `superset_config.py` file, which includes the required configurations, is shown below:

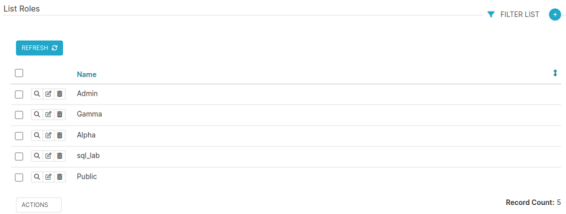

Figure 7: Screenshot of Role Configuration

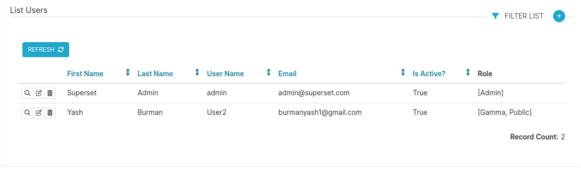

Figure 8: Screenshot of List of Users

```
FEATURE_FLAGS = {
    "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 3: superset_config.py

The `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 purpose, detailed below:

- **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.
- **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.
- **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.
- **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.
- **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.
- **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.
- **HTTP_HEADERS**: `{'X-Frame-Options': 'ALLOWALL'}` 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.

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

```
<!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:8088/superset/dashboard/p/mxVXR0kg5WJ/"></iframe>
        <footer>
            <p>Created by Yash Burman | <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 4: HTML code for embedding Superset dashboard

5 Predictive Analytics using Prophet

Prophet is an open-source forecasting tool developed by Facebook, designed specifically for forecasting time series data with daily observations. It is particularly effective for data with strong seasonal patterns and historical trends, making it ideal for agricultural production forecasts. Prophet's modeling approach is highly flexible, allowing it to handle missing data, outliers, and holiday effects with ease.

To integrate Prophet with Superset, we first install Prophet within our Docker container. The commands for this setup are given below:

```
docker exec -it superset_app bash
pip install --upgrade pip
pip install prophet
```

Listing 5: Integrating Prophet with Superset

We have applied Prophet to the *Production Trend* and *Production Trends by Season* charts to forecast crop production over the next five years, from 2024 to 2029. The forecasted trends provide valuable insights into future production levels, helping with long-term planning and resource allocation. The screenshots are attached below:

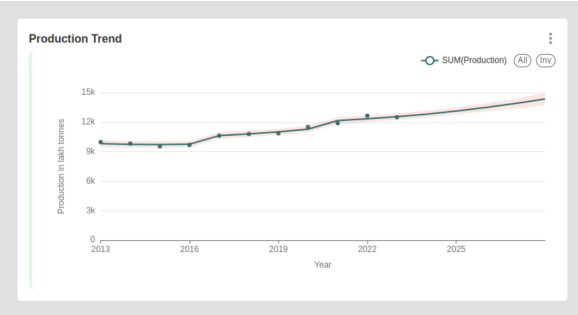

Figure 9: Predictive Analytics: Production Trend by Crop

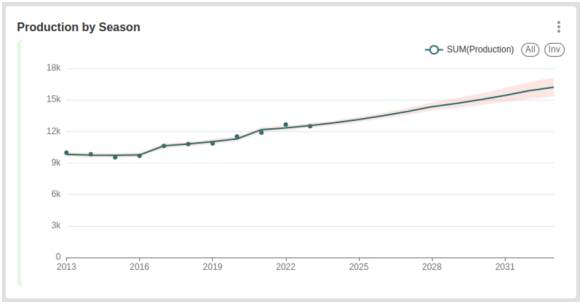

Figure 10: Predictive Analytics: Production Trend by Season

6 Conclusion and Future Exploration

This project successfully demonstrates the integration of Apache Superset to visualize and analyze crop production data, leveraging MySQL as the backend and implementing secure role-based access for users. Through embedded dashboards, it enables accessible, interactive insights into crop production trends and price correlations. Future exploration could focus on enhancing data sources, expanding analytical capabilities, and refining user experience with advanced filtering and customization options.

7 Appendix

SQL Queries Used to Build Charts

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

1. **Percentage Change in Sugar Production and Consumer Price Index for Sugar and Confectionery Products**

```
    SELECT
        CAST(CONCAT(prod.Year, '-01-01') AS DATE) AS Year,
        SUM(prod.Production) AS Total_Production,
        price.`Sugar and Confectionery` AS Previous_Year_CPI
    FROM
        crop_production_data prod
    LEFT JOIN
        Crop_price_Data price
        ON prod.Year - 1 = price.Year
    WHERE
        prod.Crop = 'Sugarcane'
        AND prod.Year BETWEEN 2014 AND 2018
    GROUP BY
        CAST(CONCAT(prod.Year, '-01-01') AS DATE), price.`Sugar and Confectionery`;

```

Listing 6: SQL query to calculate percentage change in sugar production and CPI for Sugar and Confectionery Products

2. **Percentage Change in Pulses Production and Consumer Price Index for Pulses and Related Products**

```sql
SELECT
    CAST(CONCAT(prod.Year, '-01-01') AS DATE) AS Year,
    SUM(prod.Production) AS Total_Production,
    price.`Pulses and Products` AS Previous_Year_CPI
FROM
    crop_production_data prod
LEFT JOIN
    Crop_price_Data price
    ON prod.Year - 1 = price.Year
WHERE
    prod.Crop IN ('Pulses', 'Gram', 'Peas', 'Tur (Arhar)', 'Gram (Chana)', '
Urad', 'Moong', 'Lentil (Masoor)', 'Other Pulses', 'Total Pulses')
    AND prod.Year BETWEEN 2014 AND 2018
GROUP BY
    CAST(CONCAT(prod.Year, '-01-01') AS DATE), price.`Pulses and Products`;

```

Listing 7: SQL query to calculate percentage change in pulses production and CPI for Pulses and Related Products

3. **Percentage Change in Oilseed Production and Consumer Price Index for Oilseed Products**

```sql
SELECT
    CAST(CONCAT(prod.Year, '-01-01') AS DATE) AS Year,
    SUM(prod.Production) AS Total_Production,
    price.`Oilseeds` AS Previous_Year_CPI
FROM
    crop_production_data prod
LEFT JOIN
    Crop_price_Data price
    ON prod.Year - 1 = price.Year
WHERE
    prod.Crop IN ('Soybean', 'Groundnut', 'Sunflower', 'Mustard', 'Sesamum', '
Rapeseed & Mustard', 'Linseed', 'Safflower', 'Castorseed', 'Nigerseed')
    AND prod.Year BETWEEN 2014 AND 2018
GROUP BY
    CAST(CONCAT(prod.Year, '-01-01') AS DATE), price.`Oilseeds`;

```

Listing 8: SQL query to calculate percentage change in oilseed production and CPI for Oilseed Products

4. **Percentage Change in Cereal Production and Consumer Price Index for Cereal Products**

```sql
SELECT
    CAST(CONCAT(prod.Year, '-01-01') AS DATE) AS Year,
    SUM(prod.Production) AS Total_Production,
    price.`Cereals and Products` AS Previous_Year_CPI
FROM
    crop_production_data prod
LEFT JOIN
    Crop_price_Data price
    ON prod.Year - 1 = price.Year
WHERE
    prod.Crop IN ('Rice', 'Wheat', 'Maize', 'Barley', 'Bajra', 'Jowar', 'Ragi',
 'Small Millets', 'Shree Anna', 'Nutri/Coarse Cereals', 'Cereals')
    AND prod.Year BETWEEN 2014 AND 2018
GROUP BY
    CAST(CONCAT(prod.Year, '-01-01') AS DATE), price.`Cereals and Products`;

```

Listing 9: SQL query to calculate percentage change in cereal production and CPI for Cereal Products

Link to GitHub repository