

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | ADS PROJECT REPORT | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | SHUBHDEEPBHOLE |  | | | |
|  | | | | 241099858417th November 2025—Smart Crop Recommender |  | | | |
|  | | |  | | |  | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | |  |  | | |  |
|  | INTRODUCTION | | | | | | |  |
|  |  | | |  |  | | |  |
|  |  | |  | | |  | |  |
|  |  |  | Agriculture is the backbone of the Indian economy, yet most farmers still depend on experience-based decision making rather than data-driven insights. Choosing the correct crop is influenced by multiple dynamic factors including soil nutrients, real-time weather, rainfall patterns, and volatile mandi prices. Wrong crop choice leads to reduced yield and financial losses.  **AgriSphere** is an AI-powered decision support system that provides automated, accurate, and location-based crop recommendations. The system combines:   * Machine Learning (Random Forest model) * Live weather API (OpenWeatherMap) * Real-time mandi price scraping (Agmarknet) * SQL Database logging * Power BI analytics dashboard   The goal of the project is to empower farmers with scientific and economic intelligence, enabling them to choose the most profitable and biologically suitable crop for their land. | | |  |  |  |
|  | | | | |
|  |  |  |  |
|  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decorative |  |  |  |  |
|  | MODULES | | |  |
|  |  |  |  |  |

**1. Machine Learning Recommendation Module**

* Trained on *Crop Recommendation Dataset* (NPK, temperature, humidity, pH, rainfall).
* Predicts top crops based on environmental factors.

**2. Weather API Module**

* Integrates OpenWeatherMap API.
* Fetches temperature and humidity automatically using user’s location.

**3. Soil & Location Module**

* Takes N, P, K, pH, rainfall from user or pre-defined region profiles.
* Supports auto-location using IP-based or browser-based geolocation.

**4. Web Scraping (Mandi Price) Module**

* Scrapes Agmarknet.gov.in for latest mandi prices.
* Uses BeautifulSoup with multi-strategy hardened parsing.

**5. Ranking & Reasoning Module**

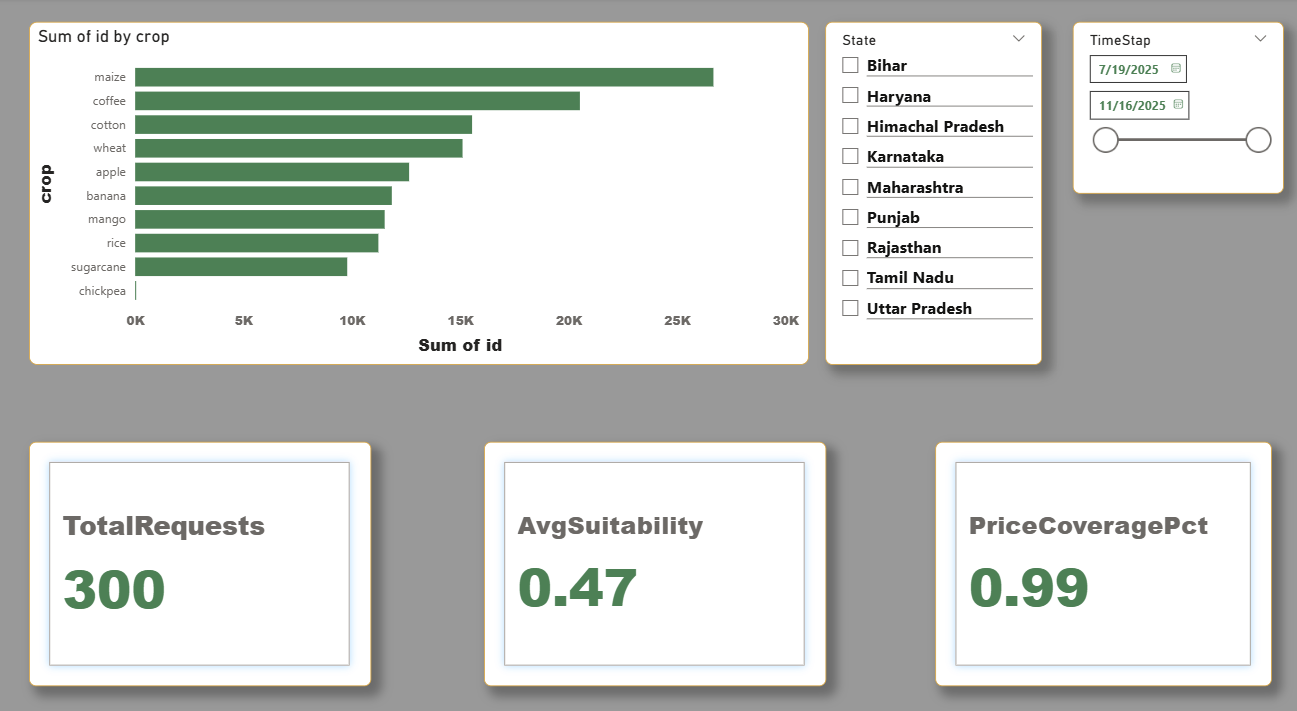
* Combines ML suitability + price normalization to compute profitability score.
* Provides explanation: *ideal vs actual values* for transparency.

**6. SQL Logging Module**

* Stores each recommendation into SQLite database.
* Tracks session inputs, predictions, and mandi pricing data.

**7. Power BI Analytics Dashboard**

* Visualizes crop trends, regional suitability, price distribution, and user locations.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decorative |  |  |  |  |
|  | 3. EXISTING SYSTEM | | |  |
|  |  |  |  |  |

In the existing agricultural system:

* Farmers rely on personal experience or advice from local dealers.
* Weather and rainfall data are not updated in real-time.
* No automated platform predicts crop suitability scientifically.
* Mandi prices vary daily, but farmers do not have instant access.
* Decision-making is manual, uncertain, and often risky.

This results in:

* Crop losses
* Low profit margins
* Wrong crop choices
* Inefficient planning

**AgriSphere** eliminates these limitations by providing a fully automated, data-driven solution.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decorative |  |  |  |  |
|  | 4. SOFTWARE REQUIREMENT SPECIFICATION (SRS) | | |  |
|  |  |  |  |  |

**Functional Requirements**

1. User can input soil parameters (N, P, K, pH, rainfall).
2. System detects user location and fetches weather automatically.
3. ML model predicts top 3 crops with suitability scores.
4. Web scraper fetches latest mandi prices.
5. System stores each recommendation in SQL database.
6. Power BI dashboard visualizes all collected data.

**Non-Functional Requirements**

* Real-time performance
* High accuracy
* Reliable API responses
* Error-handling for scraping
* Easy-to-use UI (Streamlit)

**Software Requirements**

* Python 3.x
* Streamlit
* Pandas, NumPy
* Scikit-learn
* BeautifulSoup4
* Requests
* SQLite3
* Power BI Desktop

**Hardware Requirements**

* Laptop/PC with minimum 4GB RAM
* Stable internet connection

|  |  |  |
| --- | --- | --- |
| Decorative |  |  |
|  | CODING | | |
|  |  |  |

**A. ML Model Training (app.py)**

df = pd.read\_csv("Crop\_recommendation.csv")

X = df.drop("label", axis=1)

y = df["label"]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

X, y, test\_size=0.2, random\_state=42

)

model = RandomForestClassifier(n\_estimators=200, random\_state=42)

model.fit(X\_train, y\_train)

joblib.dump(model, "model.joblib")x

B. Weather Fetching (web\_app.py)

def get\_weather(lat, lon, key):

url = f"https://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={key}&units=metric"

r = requests.get(url).json()

return r["main"]["temp"], r["main"]["humidity"]

C. Mandi Price Scraping

soup = BeautifulSoup(html, "lxml")

table = soup.find("table")

text = table.get\_text(" ")

match = re.search(r"(₹|Rs)\s\*(\d+)", text)

if match:

price = match.group(2)

D. SQL Database Logging (db.py)

def save\_recommendation(session\_info, crops, prices):

req = Request(...)

db.add(req)

db.commit()

for item in crops:

rec = Recommendation(request\_id=req.id, ...)

db.add(rec)

for p in prices:

price = Price(request\_id=req.id, ...)

db.add(price)

db.commit()

E. Final Ranking

final\_score = 0.6 \* suitability + 0.4 \* price\_normalized

F. Streamlit UI

st.title("AgriSphere – Crop Recommendation & Mandi Prices")

lat = st.number\_input("Latitude")

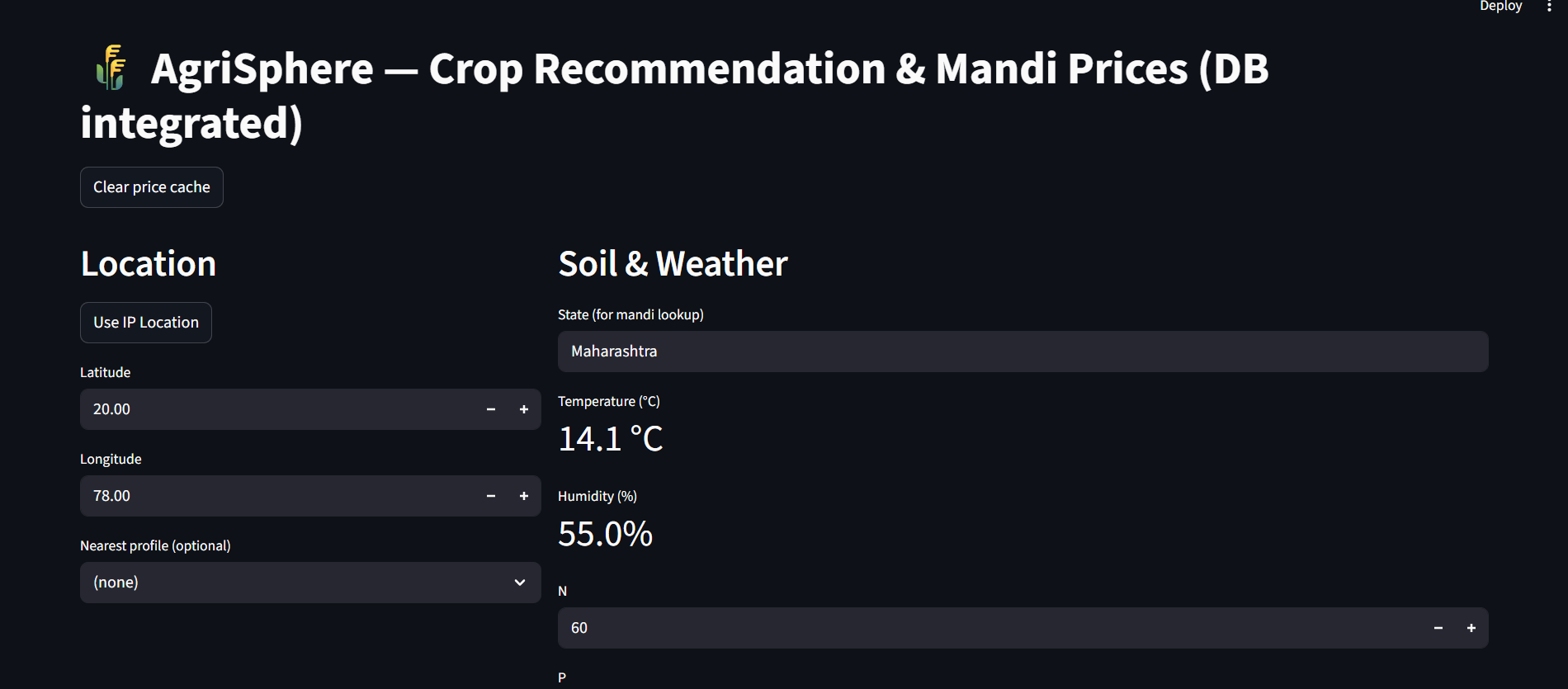
lon = st.number\_input("Longitude")

if st.button("Get Recommendations"):

temp, hum = get\_weather(lat, lon, API\_KEY)

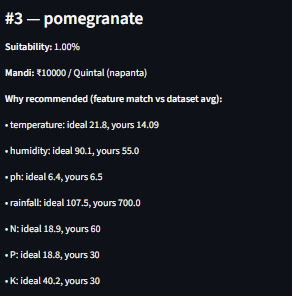
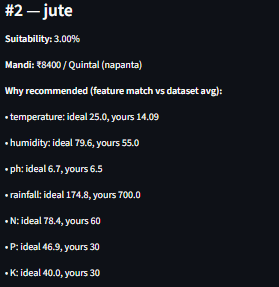
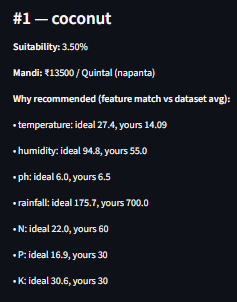
...

SCREENSHOTS WITH DESCRIPTION  
  
Home UI:



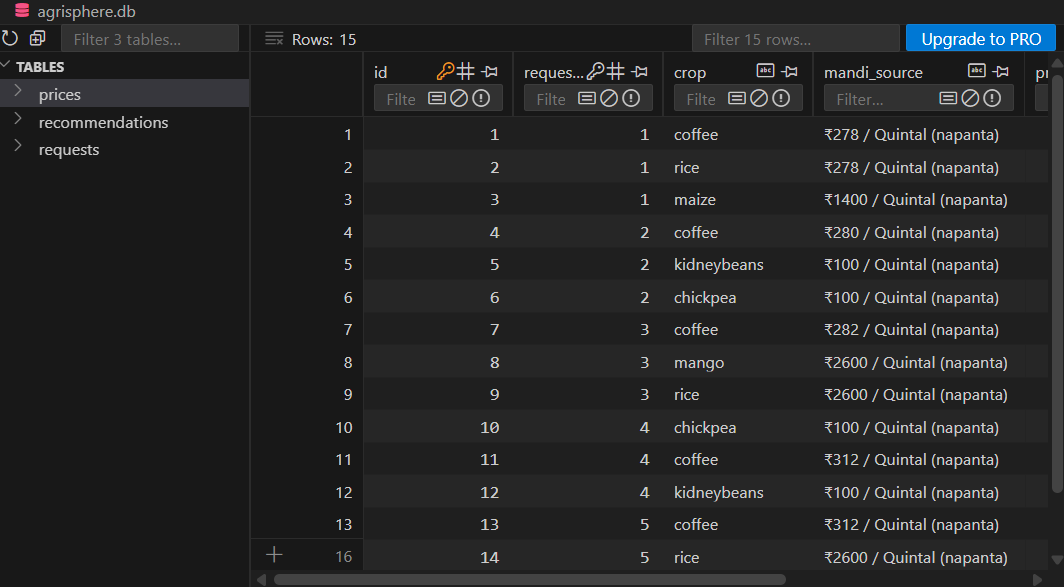
Shows the main Streamlit interface with inputs for location, weather, and soil parameters.

Top 3 Recommendations:



Displays top crops based on suitability, mandi price, and final score.

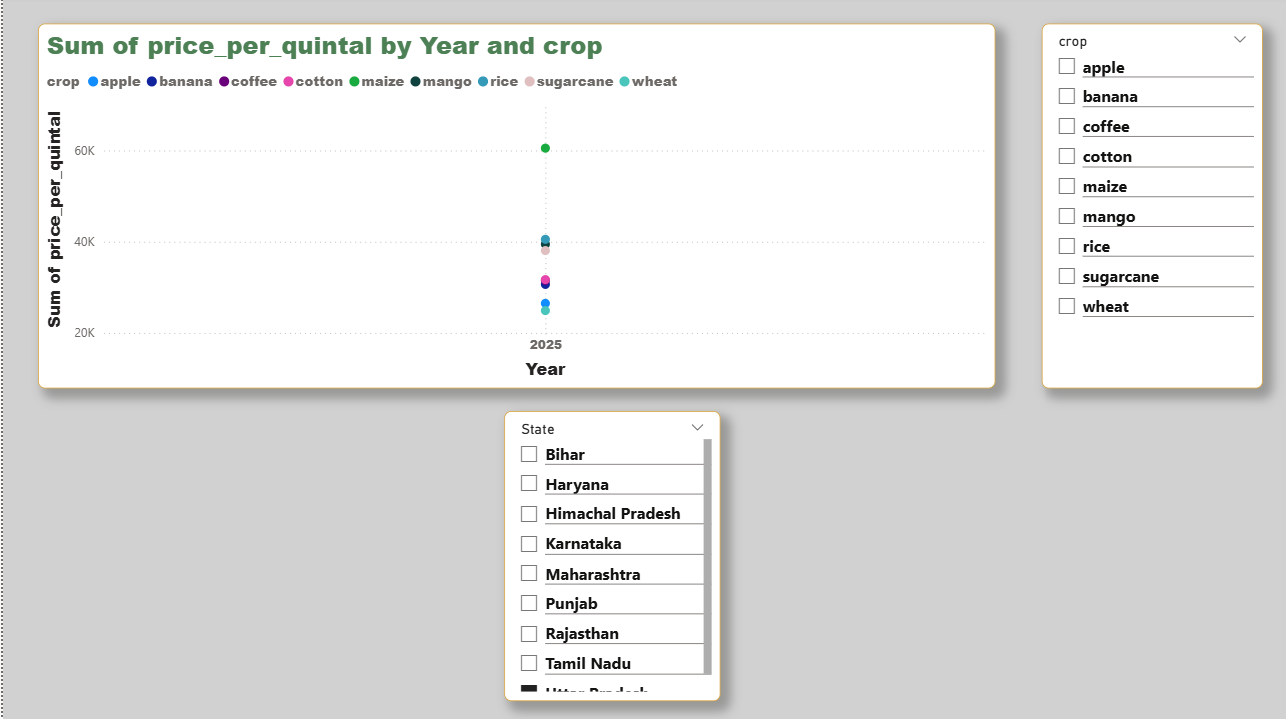
SQL Database Records:

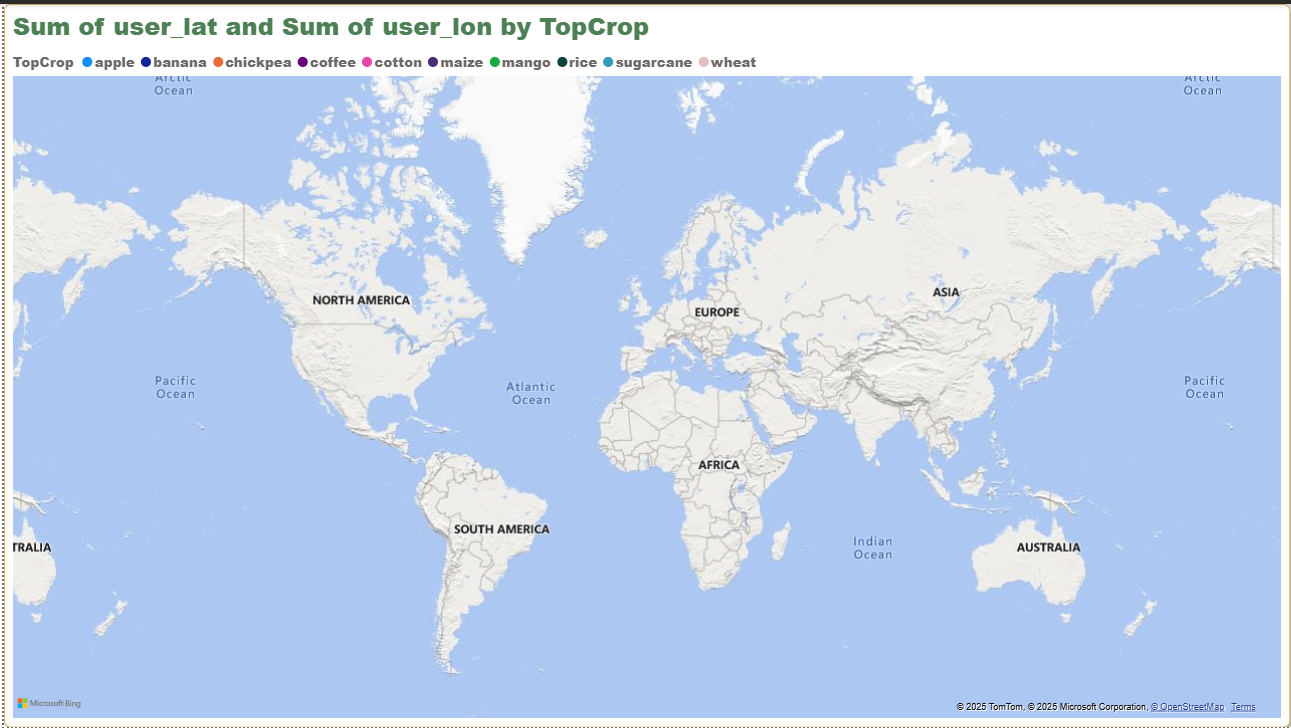


Shows how each request is stored with timestamp and details.

Power BI Dashboard:







Visual analytics including:

* Top crops
* State-wise trends
* Price trends
* User location map

1. PROJECT FLOW:

**Step-by-step Flow**

 User opens app

 Location auto-detected

 Weather fetched from API

 User inputs soil data

 ML model predicts crop probabilities

 Web scraper fetches mandi prices

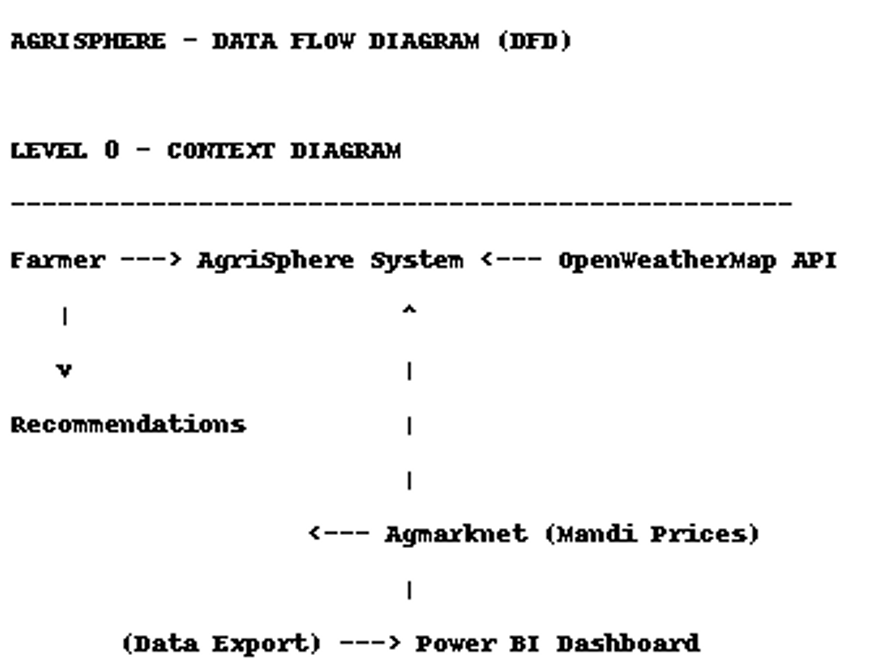
 Algorithm ranks crops based on suitability + price

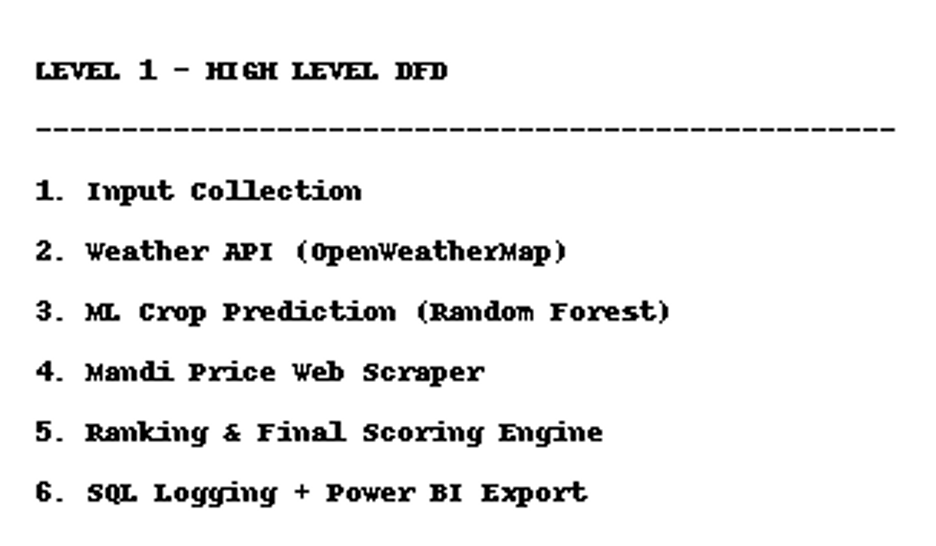
 Results displayed to user

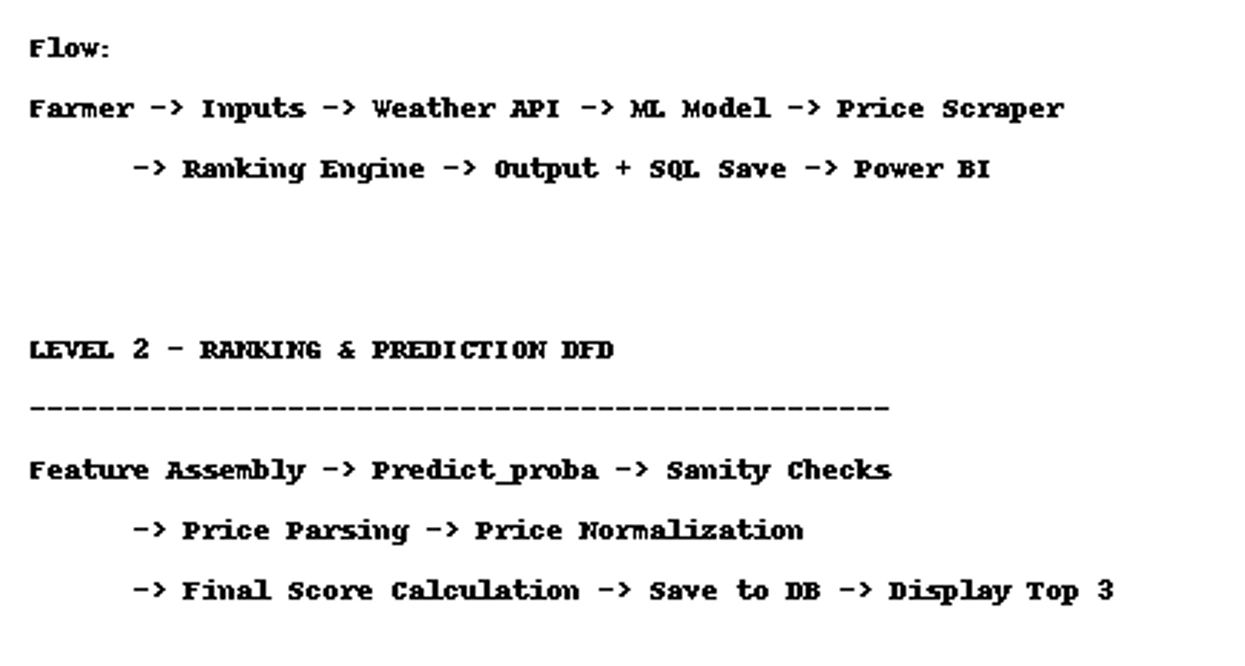
 Data stored in SQL

 Power BI reads SQL export and visualizes trends

**Data Flow Diagram (DFD)**

****

****

****

**8. Testing:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | Input | Expected Output | **Status** |
| |  | | --- | | Weather API Test |  |  | | --- | |  | | |  | | --- | | Lat/Lon |  |  | | --- | |  | | Correct temp/humidity | Pass |
| |  | | --- | | ML Prediction Test |  |  | | --- | |  | | |  | | --- | | Soil values |  |  | | --- | |  | | |  | | --- | | Top 3 crops |  |  | | --- | |  | | Pass |
| |  | | --- | | Scraper Test |  |  | | --- | |  | | |  | | --- | | Crop + State |  |  | | --- | |  | | |  | | --- | | Latest mandi price |  |  | | --- | |  | | Pass |
| |  | | --- | | DB Insert Test |  |  | | --- | |  | | |  | | --- | | Session info |  |  | | --- | |  | | |  | | --- | | Data saved |  |  | | --- | |  | | Pass |
| |  | | --- | | UI Test |  |  | | --- | |  | | |  | | --- | | Button click |  |  | | --- | |  | | |  | | --- | | Results display |  |  | | --- | |  | | Pass |

* 1. **Future Enhancements**

 Mobile app for farmers

 Local language support (Punjabi, Hindi, Marathi)

 Satellite-based soil detection

 Live rainfall forecasting

 AI-based price prediction

 Integration with government crop schemes

 Voice assistant support

* 1. **CONCLUSION:**

AgriSphere successfully demonstrates how machine learning, live APIs, web scraping, and data analytics can transform agricultural decision-making. The system provides accurate crop recommendations by combining scientific factors (soil + weather) with economic factors (mandi prices).  
The project not only benefits farmers but also showcases the power of integrating Python, SQL, and Power BI into a single intelligent system.