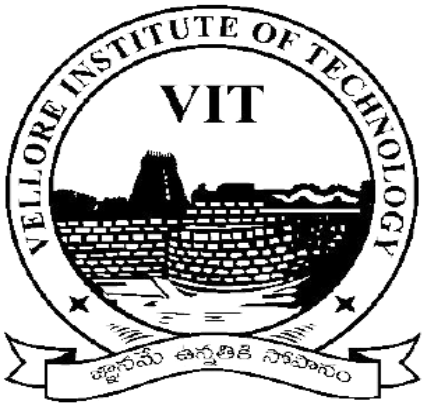


VIT®
AP

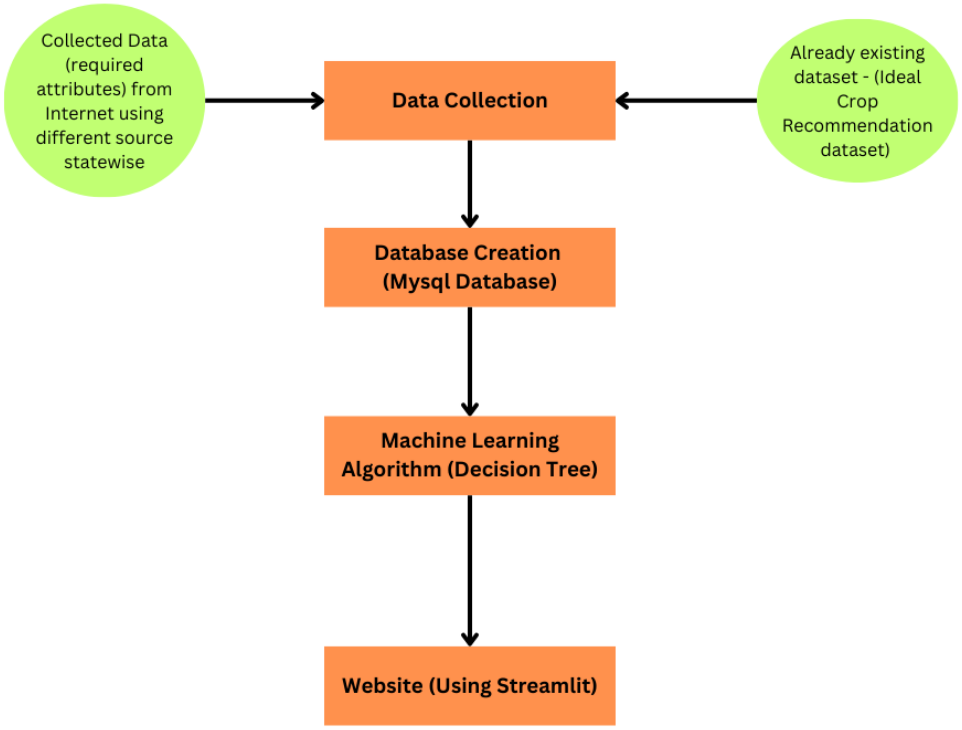
CROP RECOMMENDER SYSTEM

Capstone - Review II

Faculty:
Dr. Ravi Shankar Barpanda

Team Members: -
Tharun Kumar K – 21BCE8765
Mohajit Neog - 21BCE8170
Ankit Singh Hada – 21BCE7420
Nikesh Kumar - 21BCE8603

Overall Process (up until now):



App Stack:

1. MS Excel
2. MySQL Workbench
3. Google Collab
4. Superset
5. Power BI
6. VS Code

Data Collection:

The dataset consists of parameters like Nitrogen (N), phosphorus (P), Potassium (K), PH value of soil, Humidity, Temperature, and Rainfall.

Has **2200** instances of data that have taken from past historical data. This dataset includes **22** different crops such as rice, maize, chickpea, kidney beans, pigeon peas, moth beans, mungbean, black gram, lentil, pomegranate, banana, mango, grapes, watermelon, muskmelon, apple, orange, papaya, coconut, cotton, jute, and coffee.

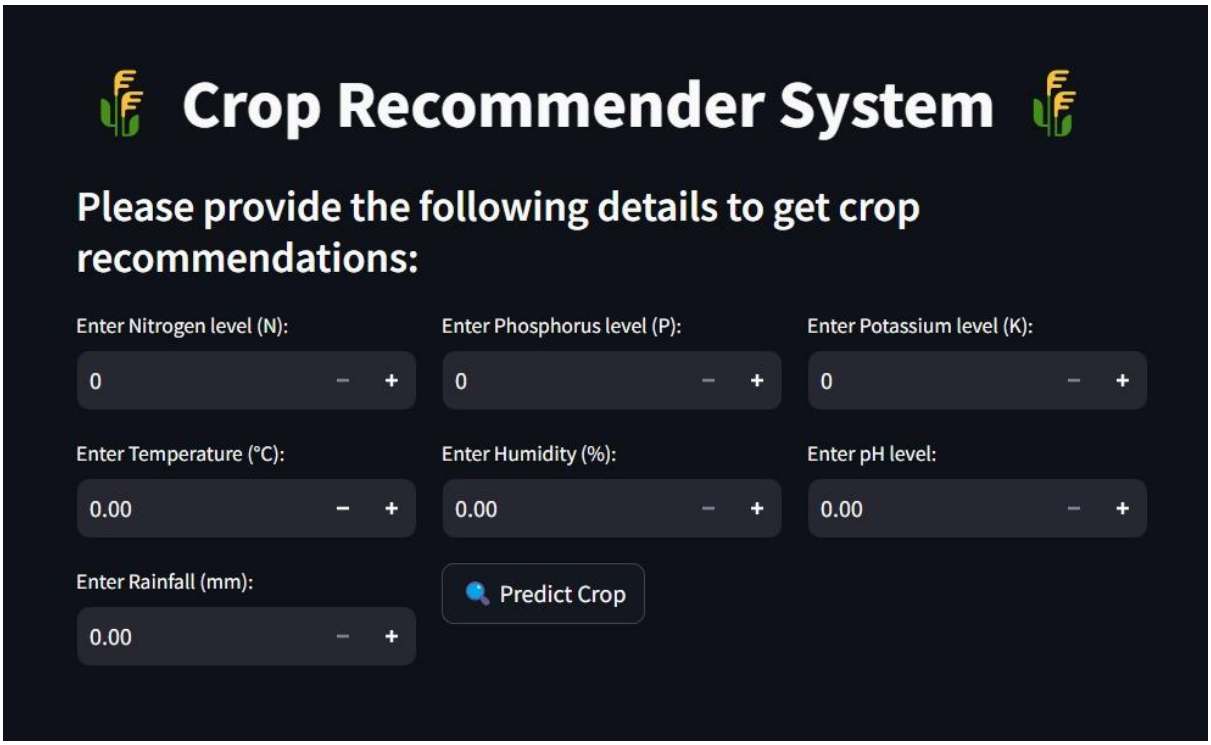
Machine Learning- Decision Tree Classifier

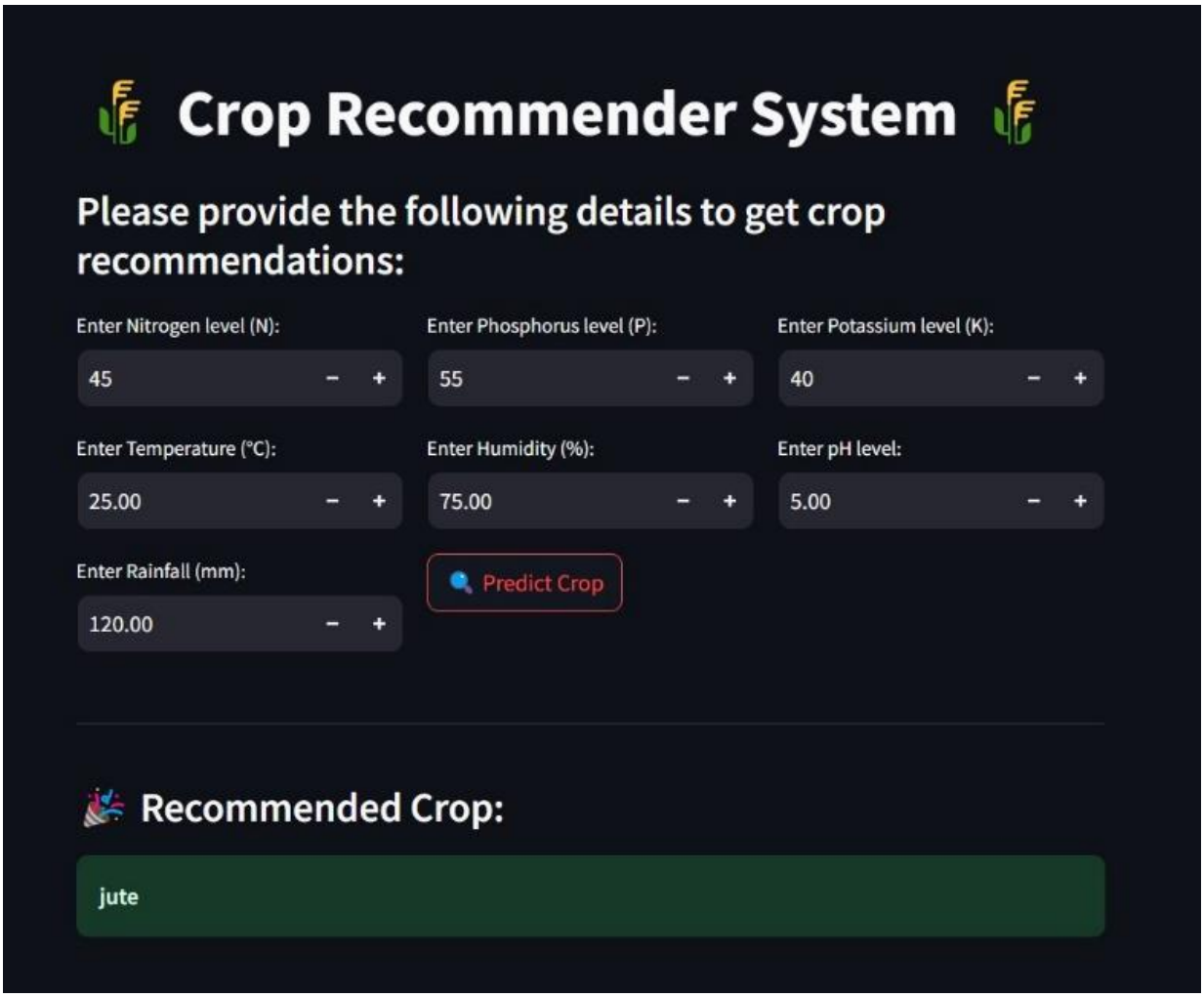
Decision tree classifiers utilize greedy methodology. It is a supervised learning algorithm where attributes and class labels are represented using a tree. The main purpose of using a Decision Tree is to form a training prototype that we can use to foresee class or value of target variables by learning decision rules deduced from previous data (training data). The Decision tree can be described by two distinct types, namely decision nodes and leaves. The leaves are the results or the end results. Each node in the tree acts as a test case for some attribute, and each edge descending from that node corresponds to one of the possible answers to the test case. This process is recursive and is repeated for every sub-tree rooted at the new nodes.

```
from sklearn.preprocessing import LabelEncoder
labelencoder_y = LabelEncoder()
dataset['label']= labelencoder_y.fit_transform(dataset['label'])
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

x = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)

from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier(criterion='entropy',random_state=0)
dtc.fit(x_train,y_train)

y_pred_dtc=dtc.predict(x_test)
y_pred_dtc

from sklearn import metrics
print('Accuracy : ',metrics.accuracy_score(y_pred_dtc,y_test))
```

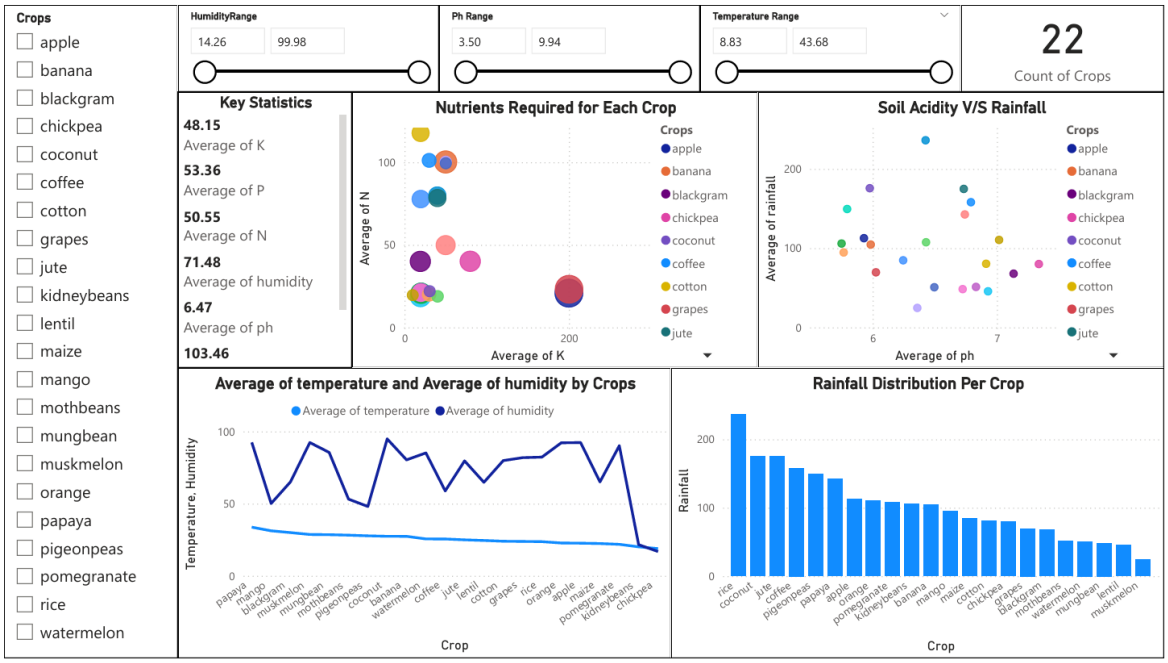
FrontEnd – Using Streamlit

Website (Intial View – 1st Look):





Dashboard



Work to be done:

1. Expand the website by adding more pages and ensure it is fully functional.
2. Integrate interactive dashboards within the website for seamless data visualization and final review.
3. Introduce a blog section to provide additional content, updates, and insights.
4. Develop a dashboard similar to Power BI using open-source BI tools such as Superset or Metabase.
5. Make the MySQL database accessible to users so they can utilize it for their own projects and research.
6. Implement state-wise data integration on the website, allowing users to select their state and enabling automatic data population for a personalized experience.