Task-2 Report

DataMites

University of Colombo School of Computing

Task

provided with a dataset containing information about crops, including their nutrient levels (N, P, K), environmental factors (temperature, humidity, pH, rainfall), and corresponding labels. Your task is to create a predictive model that recommends the best three crops based on the provided conditions. Using machine learning techniques, build a model that takes into account the input features and predicts the most suitable crops for cultivation

Approach

pThe primary goal of this report is to outline a robust approach for recommending the top 3 crops to farmers based on a comprehensive analysis of agricultural data. Rather than solely emphasizing predictive accuracy, our approach prioritizes generalization to diverse agricultural contexts.

Table of Content

1. Data Exploration

Provided dataset has 11 columns and 2200 records of tabular data.This dataset appears to contain information related to agricultural factors and perhaps crop yields or quality. Here's a brief breakdown of each column:

1.N, P, K: These seem to represent different nutrients or fertilizers used in agriculture.

2.Temperature, Humidity, pH, Rainfall: Environmental factors that can affect crop Growth.

3.Total_Nutrients: Possibly the combined value of N, P, and K.

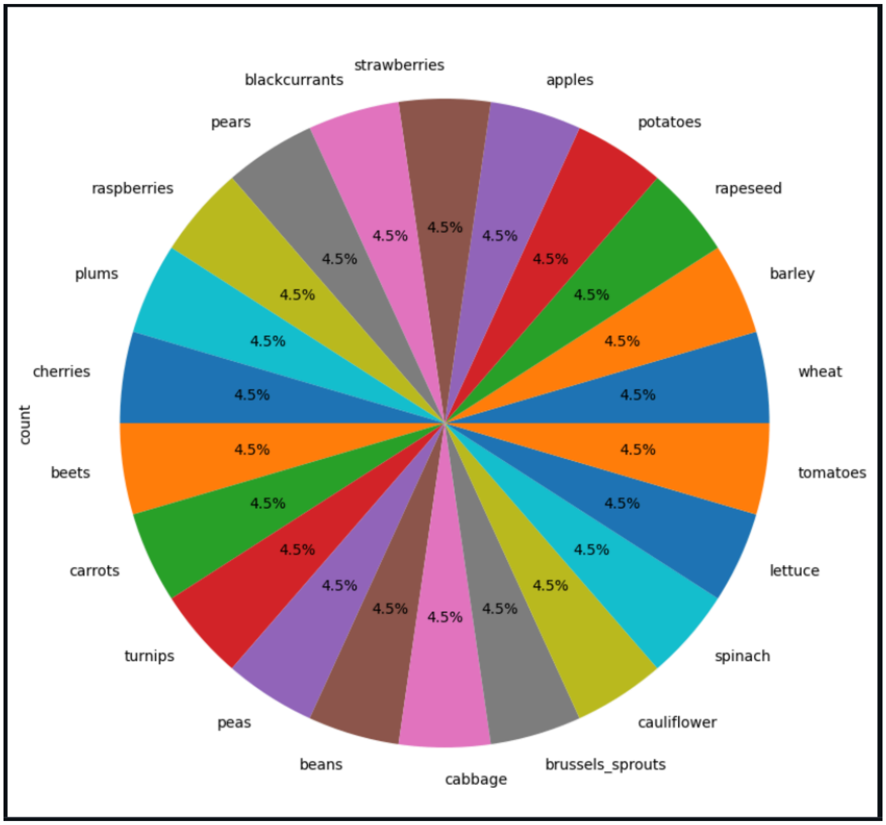
4.Temperature_Humidity: Possibly an interaction feature between temperature and humidity.

5.Log_Rainfall: Logarithm of rainfall, perhaps to normalize its distribution.

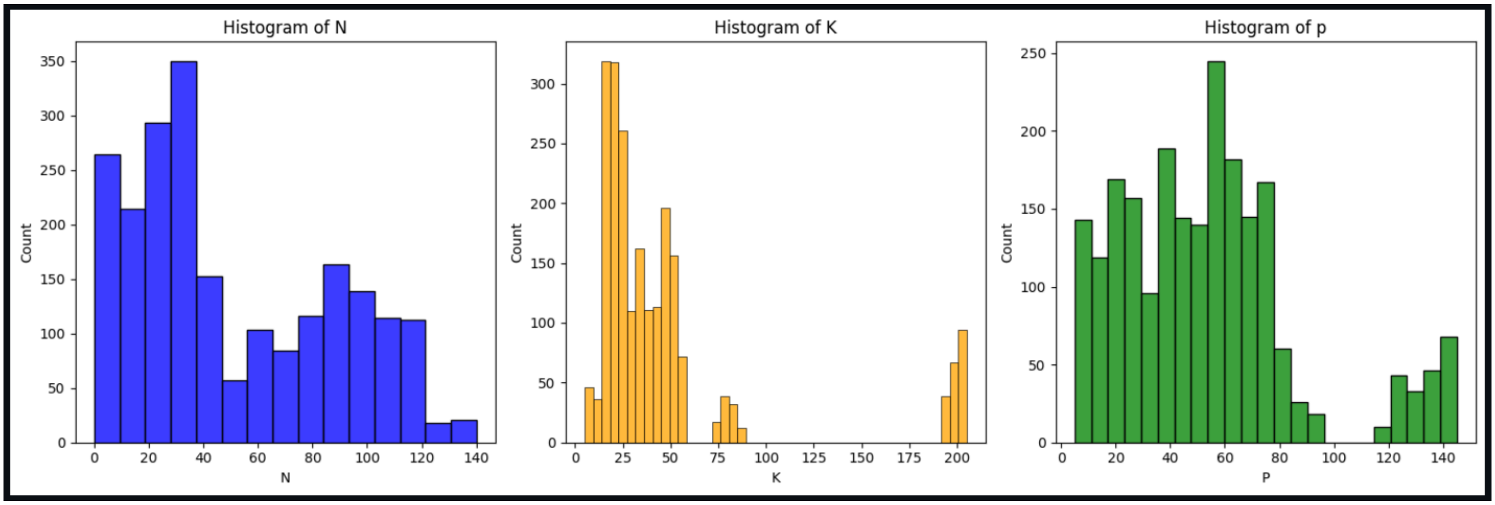
6.Label: Categorical variable indicating some outcome or classification.

7.Label_Encoded: Encoded version of the 'Label' column,

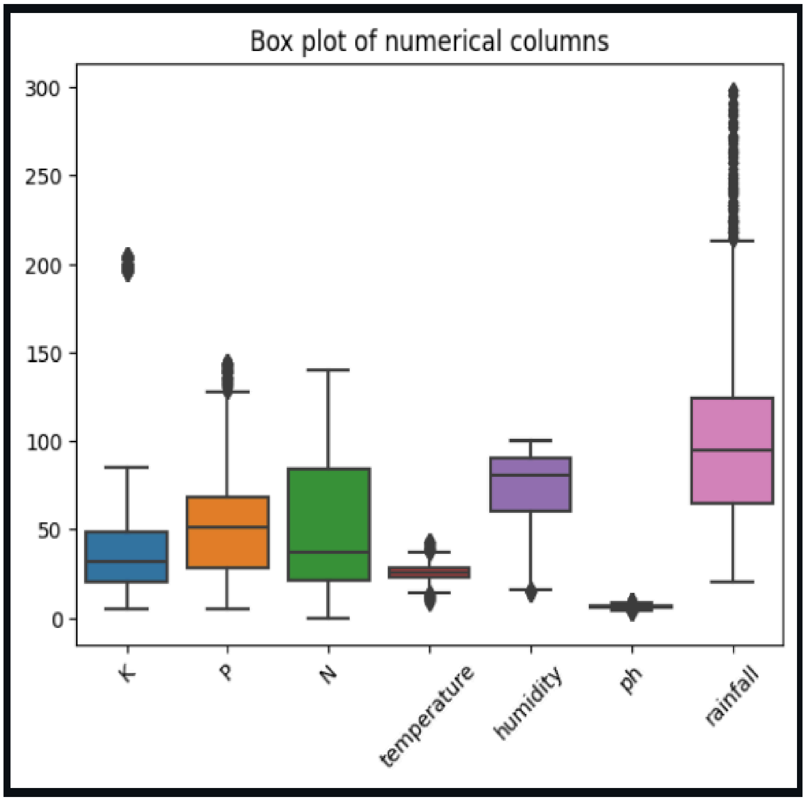
The dataset divided equally according to “Label” each type has 100 records for 22 labels.



In the features K,P,N has roughly Binomial distribution according to histplots.In feature K there is an outlier..

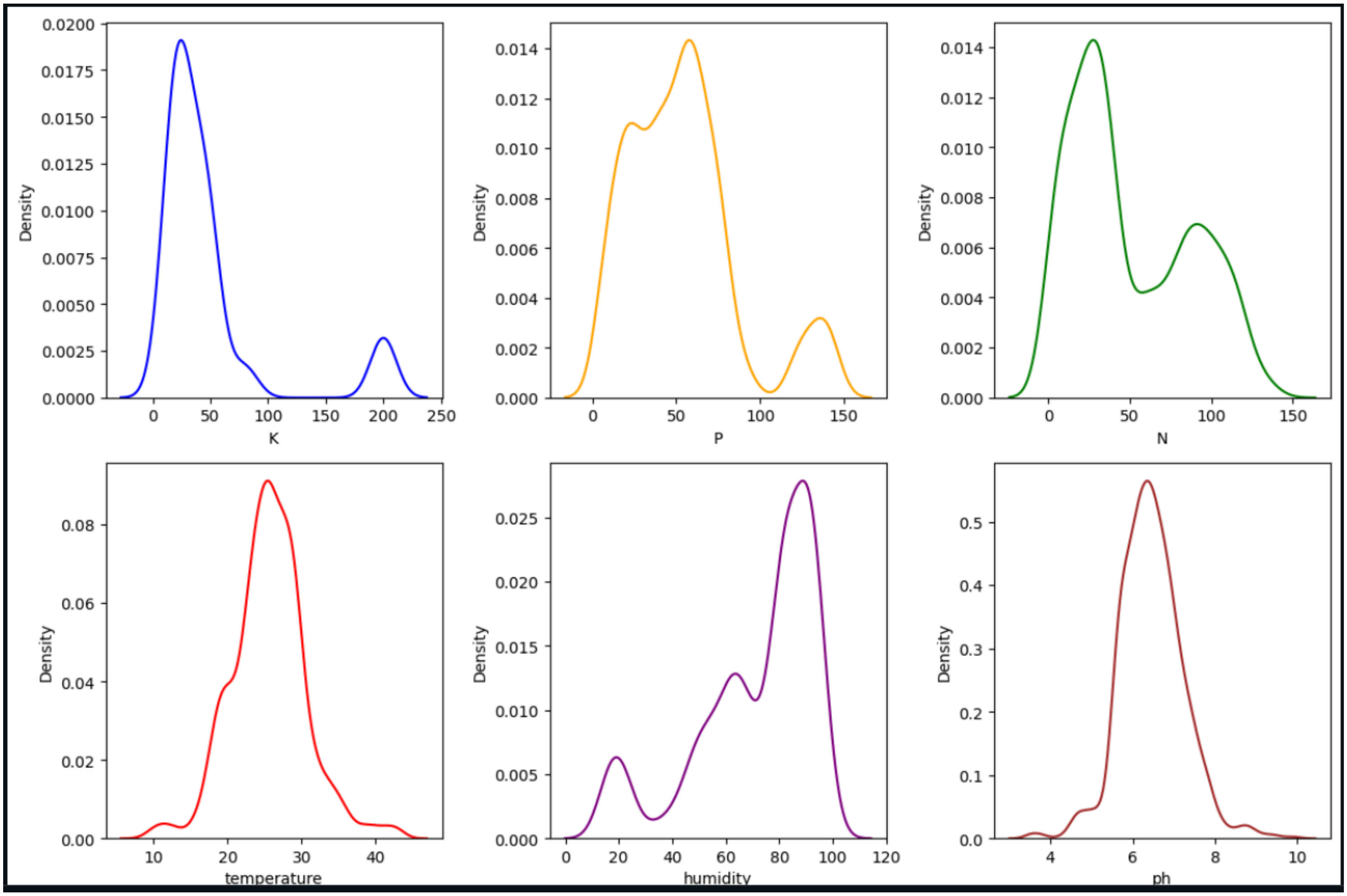


We were able to identify more outliers belonging to other factors.It clearly shows in the below box plot.



Instead of dealing with outliers we decided to put it as it is as our goal to generalize.

Below KDE Plot Binomial distribution of K , P and N shows clearly.and the temperature and ph have normal distribution.



Following findings, values show the skewness of each column.

column	**skewness**
K	**2.375167**
P	**1.010773**
N	**0.509721**
temperature	**0.184933**
humidity	**-1.091708**
ph	**0.283929**
rainfall	**0.965756**

Following are the Correlations of 'Label_Encoded' with other columns.

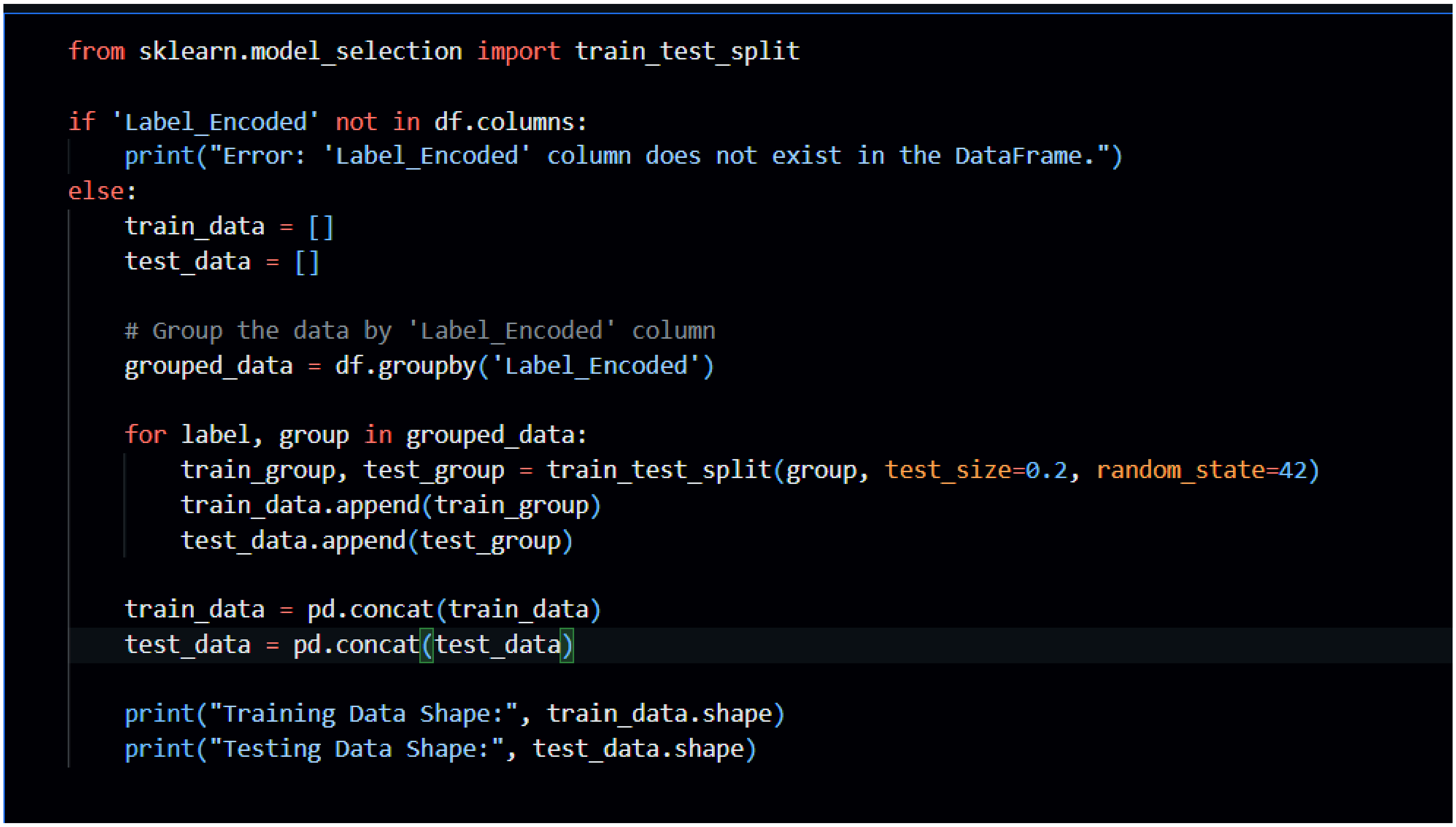
column	Correlation
K	**0.143703**
P	**-0.167951**
N	**0.282787**
temperature	**0.180571**
humidity	**0.524452**
ph	**0.052389**
rainfall	**0.121486**

By checking correlation between similar domain columns we found.

1.Correlation between humidity and rainfall is: 0.09
2.Correlation between humidity and temperature is: 0.21
3.Correlation between temperature and rainfall is: -0.03

2. Data Preprocessing

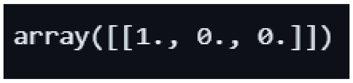
First of all we separated a random sample to use as unseen data.This sample is not going to be used for training or testing.
Instead of using “Train Test Split” we use it with a condition to balance each class and pick nearly the same amount of test values from each class. This way allow us to make proper balanced Train Test Split among classes.



```python
from sklearn.model_selection import train_test_split

if 'Label_Encoded' not in df.columns:
    print("Error: 'Label_Encoded' column does not exist in the DataFrame.")
else:
    train_data = []
    test_data = []

    # Group the data by 'Label_Encoded' column
    grouped_data = df.groupby('Label_Encoded')

    for label, group in grouped_data:
        train_group, test_group = train_test_split(group, test_size=0.2, random_state=42)
        train_data.append(train_group)
        test_data.append(test_group)

    train_data = pd.concat(train_data)
    test_data = pd.concat(test_data)

    print("Training Data Shape:", train_data.shape)
    print("Testing Data Shape:", test_data.shape)
```

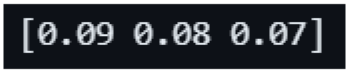
3. Model Selection and Training

According to our goal of generalizing the model, RandomForest and DecisionTree algorithms are not suitable.It predict the class with high accuracy.Using this model we cannot get top 3 recommendation.It suit for classifications.following output prove that it classify it 1 probability but it cannot identify wich should be the 2nd and 3rd.



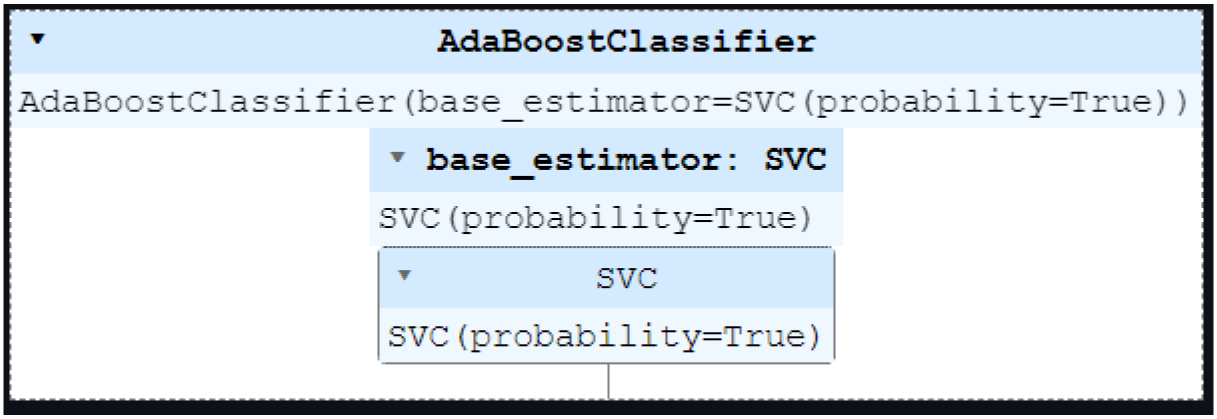
```
array([[1., 0., 0.]])
```

To address this problem we identified SupportVectorClassifier is the best solution.So we used Adaboosting with it following shows the outputs of the model.



```
[0.09 0.08 0.07]
```

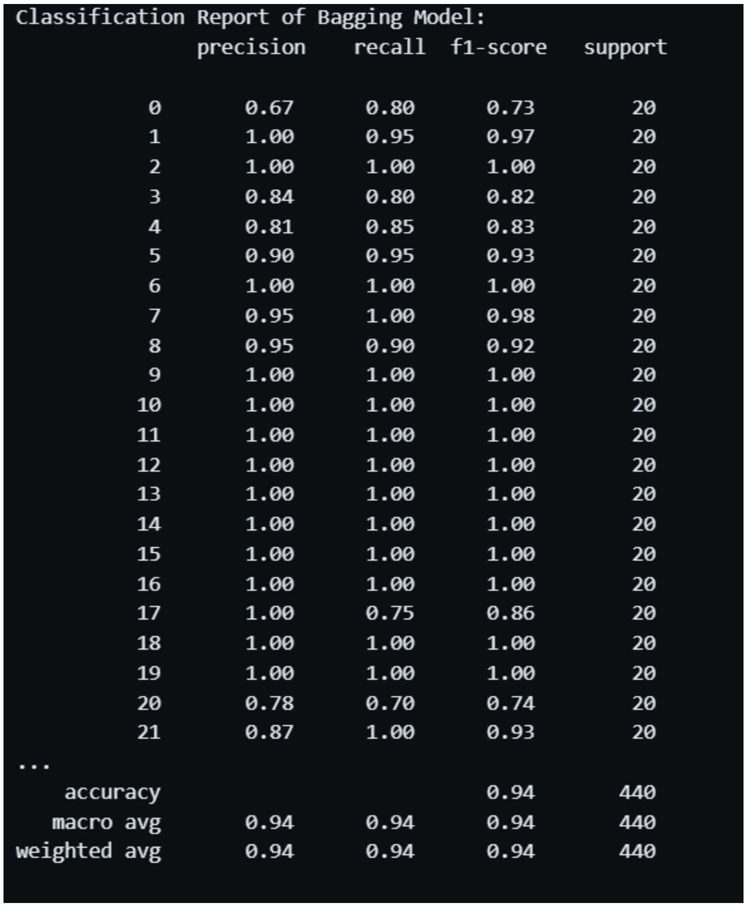
Compared to the Trees, Probability enabled SVC is the best solution to address this Problem.



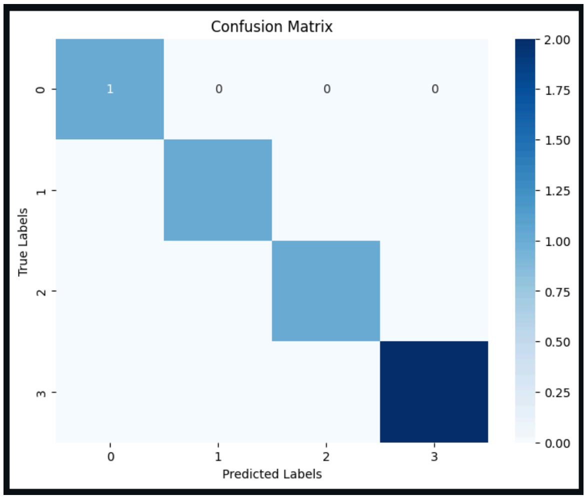
```
AdaBoostClassifier
AdaBoostClassifier(base_estimator=SVC(probability=True))
base_estimator: SVC
SVC(probability=True)
SVC
SVC(probability=True)
```

4. Evaluate the Model

Following show The classification report of the model.it has 94% f-score.

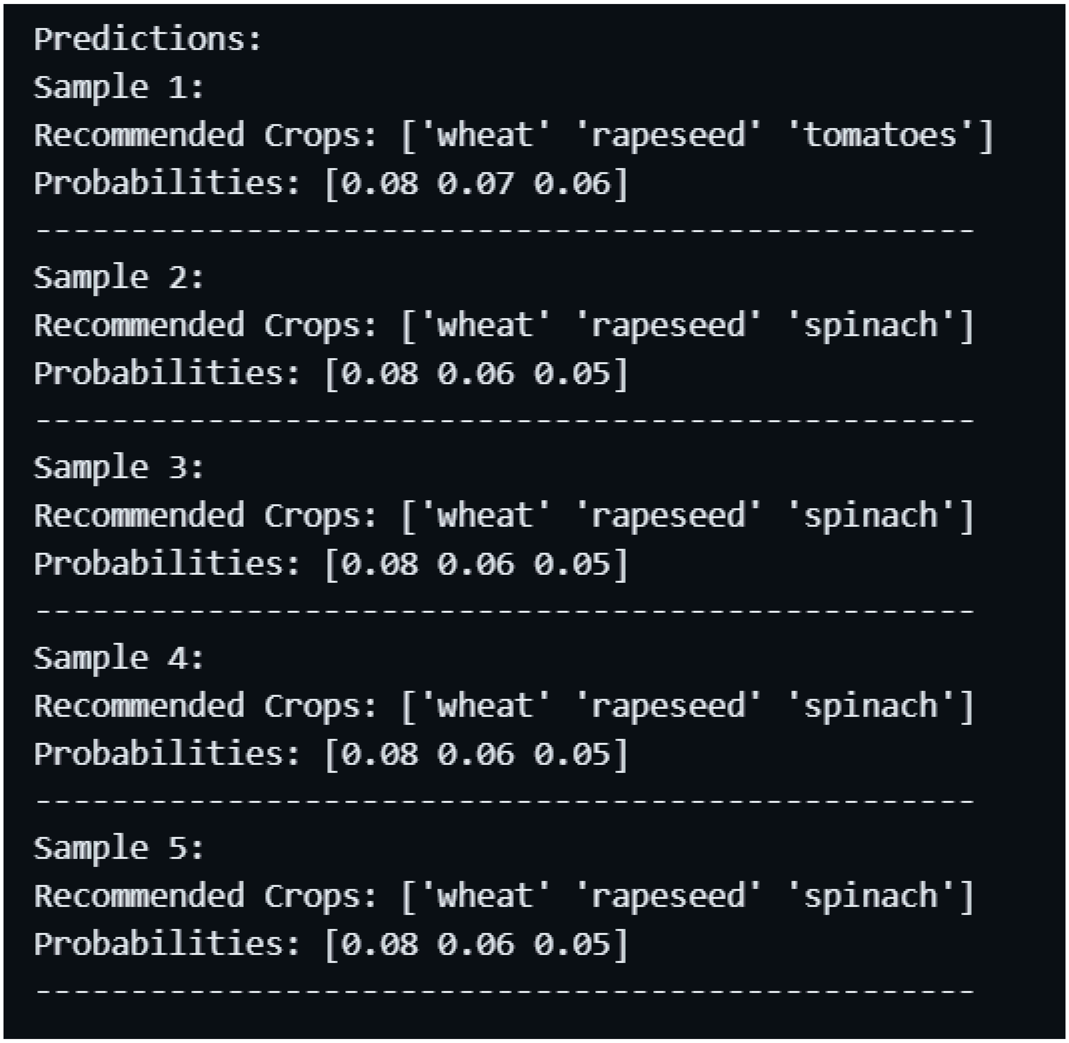


```
Classification Report of Bagging Model:
              precision    recall  f1-score   support

           0       0.67      0.80      0.73        20
           1       1.00      0.95      0.97        20
           2       1.00      1.00      1.00        20
           3       0.84      0.80      0.82        20
           4       0.81      0.85      0.83        20
           5       0.90      0.95      0.93        20
           6       1.00      1.00      1.00        20
           7       0.95      1.00      0.98        20
           8       0.95      0.90      0.92        20
           9       1.00      1.00      1.00        20
          10       1.00      1.00      1.00        20
          11       1.00      1.00      1.00        20
          12       1.00      1.00      1.00        20
          13       1.00      1.00      1.00        20
          14       1.00      1.00      1.00        20
          15       1.00      1.00      1.00        20
          16       1.00      1.00      1.00        20
          17       1.00      0.75      0.86        20
          18       1.00      1.00      1.00        20
          19       1.00      1.00      1.00        20
          20       0.78      0.70      0.74        20
          21       0.87      1.00      0.93        20
...
    accuracy                           0.94       440
   macro avg       0.94      0.94      0.94       440
weighted avg       0.94      0.94      0.94       440
```



5. Joblib Model Creation and Prediction

To reproduce the code put the dataset in same directory and install dependencies using pip after the notebook run you can use the last 2 cells make predictions without running whole notebook.



```
Predictions:
Sample 1:
Recommended Crops: ['wheat' 'rapeseed' 'tomatoes']
Probabilities: [0.08 0.07 0.06]
--------------------------------------------------
Sample 2:
Recommended Crops: ['wheat' 'rapeseed' 'spinach']
Probabilities: [0.08 0.06 0.05]
--------------------------------------------------
Sample 3:
Recommended Crops: ['wheat' 'rapeseed' 'spinach']
Probabilities: [0.08 0.06 0.05]
--------------------------------------------------
Sample 4:
Recommended Crops: ['wheat' 'rapeseed' 'spinach']
Probabilities: [0.08 0.06 0.05]
--------------------------------------------------
Sample 5:
Recommended Crops: ['wheat' 'rapeseed' 'spinach']
Probabilities: [0.08 0.06 0.05]
--------------------------------------------------
```