

**University of Information Techonology**



A project report on

# **Precision Farming and Predictive Analytics of Crop Types Based on Soil Health and Climate**

by

Fourth year, Knowledge Engineering

**TNT-886 Sein Wai Htut**

**TNT-921 May Sabal Myo**

**TNT-938 Hlaing Min Oo**

**TNT-960 Hnin Htet Htet**

**TNT-992 Aung Min Hein**

**CST – 42315, Data Analysis and Management**

**Year 2022**



# Table of Contents

|  |           |
|--|-----------|
| <b>Abstract .....</b>  | <b>3</b>  |
| <b>1. Introduction.....</b>                                  | <b>4</b>  |
| <b>2. About Crop Datasets.....</b>                           | <b>5</b>  |
| 2.1. Datasets.....   | 5         |
| 2.2. Technique used.....                                     | 6         |
| 2.2.1. Python .....  | 6         |
| 2.2.2. Tableau .....   | 6         |
| 2.3. Data Preprocessing .....                                | 7         |
| <b>Methodology .....</b>                                     | <b>10</b> |
| <b>3. Classification .....</b>                               | <b>10</b> |
| 3.1. Data preparation .....                                  | 10        |
| 3.2. Implementation with Naïve Bayes Algorithm .....         | 11        |
| 3.2.1. Modelling.....  | 12        |
| 3.2.2. Performance Analysis.....                             | 13        |
| 3.3. Implementation with K-Nearest Neighbors Algorithm ..... | 14        |
| 3.3.1 Tuning Hyperparameter .....                            | 14        |
| 3.3.2 Modelling.....   | 15        |
| 3.3.3. Performance Analysis.....                             | 16        |
| 3.4. Implementation with Decision Tree Algorithm .....       | 17        |
| 3.4.1. Tuning Hyperparameter .....                           | 18        |
| 3.4.2. Modelling.....  | 19        |
| 3.4.3. Performance Analysis.....                             | 19        |
| 3.5 Choosing the proper algorithm .....                      | 20        |
| <b>4. Clustering.....</b>                                    | <b>21</b> |
| 4.1. Implementation with K-Means.....                        | 21        |
| 4.1.1. Tuning Hyperparameter .....                           | 22        |
| 4.1.2. Modelling.....  | 23        |
| 4.1.2. Performance Analysis.....                             | 24        |
| 4.2. Usage of clustering results .....                       | 25        |
| <b>5. Conclusion .....</b>                                   | <b>26</b> |
| <b>6. References .....</b>                                   | <b>27</b> |
| <b>Appendix .....</b>  | <b>28</b> |

# Abstract

Agriculture is a major contributor to the Myanmar economy. The common problem existing among the Myanmar farmers are they don't choose the right crop based on their soil requirements and climate change. And also, they don't know how to care the soil. Due to this, they face a serious setback in productivity. This problem of the farmers has been addressed through precision agriculture.

Precision agriculture is a modern farming technique that uses research data of soil characteristics, soil types, crop yield data collection and suggests the farmers the right crop based on their site-specific parameters. This reduces the wrong choice on a crop and increases the productivity.

In this report, we are building an intelligent system, which intends to assist the Myanmar farmers in making an informed decision about which crop to grow depending on the sowing season, his farm's geographical location and soil characteristics. And based on the market trend, the farmers get to know about the other crops that can be grown on their land with the information of the climate change and soil requirements.

# 1. Introduction

Our country, Myanmar, is an agricultural country, and the agriculture sector is the backbone of its economy. FAO presents that the agriculture sector contributes to 37.8% of gross domestic product, accounts for 25% to 30% of total export earnings and employs 70% of the labor force. One major economic objective is “Development of agriculture as a base and all-round development of other sectors of the economy as well.” In the World bank, Myanmar Analysis of Farm Production Economics finds that agricultural productivity in Myanmar is low. In Ayeyarwady, farmers spend more than 100 days per hectare on monsoon rice paddy compared to 52 days in Cambodia, 22 days in Vietnam, and 11 days in Thailand. Myanmar has the lowest profits from rice production compared to those achieved by farmers in Asia’s other rice bowls.

The Soil Health is also important for farmer because the impacts of soil fertility are reflected in most of the Sustainable Development Goals. The aim of project is to bring innovative idea to be used in agriculture needs through technology solutions. It proposed to optimize the production of the crop and to improve the income of smallholder farmers in Myanmar. The technical part of this project consists of implementing and using data mining techniques to provide predictive insights to farmers, thereby helping them make an informed decision about which crop to grow.

In Myanmar, there is no record history about crop. So, in this report, we use the crop dataset from India. As India is a neighborhood country, we believe that most of all information will be the same with us. In this dataset, there are 22 types of crops with 7 features. For Models and language to be used for implementation will be python with Jupyter Notebook that is open-source particularly useful libraries that can help with our goal, then Tableau as a visualization tool for some graphical results.

## 2. About Crop Datasets

### 2.1. Datasets

The data applied for our project was obtained from Kaggle, the world's largest data science community with powerful tool and resources. It was made for crop yield prediction. we have a database of around 2200 samples this database has all different types of conditions related to farming. It can suggest about 22 different types of crops that a farmer can grow in the field.

The data has 7 features and one label.

1. N: ratio of Nitrogen content in soil
2. P: ratio of Phosphorous content in soil
3. K: ratio of Potassium content in soil
4. Temperature: it measured in Celsius scale
5. Humidity: it measured in grams of water vapor per cubic meter volume of air
6. PH: PH value of the soil
7. Rainfall: it measured in millimeters per hour in the sample

## 2.2. Technique used

In this report, we will implement Naïve Bayes, K-Nearest Neighbors and Decision Tree for classification model, and K-means for clustering using python. We use Tableau for data visualization.

### 2.2.1. Python

We use Python version 3.7.9 with Anaconda Jupyter Notebook. We use Data Science libraries that are `Pandas` and `Numpy` for manipulations. For visualizations in python, we use `matplotlib.pyplot` and `seaborn`. We use `StandardScaler` from `sklearn` for normalization and standardization in data preprocessing. For classification, we use Gaussian Naïve Bayes Classifier, K-Neighbors Classifier, Decision Tree Classifier from `scikit-learn` library. We also use `classification_report` from `scikit-learn` library for presenting classification accuracy. For clustering, we import `KMeans` from `scikit-learn`. For tuning hyperparameter in clustering, we use `silhouette` from `sklearn` and *Elbow Method*.

### 2.2.2. Tableau

*Tableau* products query relational databases, online analytical processing cubes, cloud databases, and spreadsheets to generate graph-type data visualizations. With Tableau, scatter plot for correlation, bar plot and packed bubbles for distribution are used. And also, for analytics, we use average line and median with quartiles from analytics tag.

## 2.3. Data Preprocessing

First, we load some of the data from the table as shown in Figure.

|   | N  | P  | K  | temperature | humidity  | ph       | rainfall   | label |
|---|----|----|----|-------------|-----------|----------|------------|-------|
| 0 | 90 | 42 | 43 | 20.879744   | 82.002744 | 6.502985 | 202.935536 | rice  |
| 1 | 85 | 58 | 41 | 21.770462   | 80.319644 | 7.038096 | 226.655537 | rice  |
| 2 | 60 | 55 | 44 | 23.004459   | 82.320763 | 7.840207 | 263.964248 | rice  |
| 3 | 74 | 35 | 40 | 26.491096   | 80.158363 | 6.980401 | 242.864034 | rice  |
| 4 | 78 | 42 | 42 | 20.130175   | 81.604873 | 7.628473 | 262.717340 | rice  |

Figure 1: Sample Dataset

```
rice          100
maize         100
jute          100
cotton        100
coconut       100
papaya        100
orange        100
apple         100
muskmelon     100
watermelon    100
grapes        100
mango         100
banana        100
pomegranate   100
lentil        100
blackgram     100
mungbean      100
mothbeans     100
pigeonpeas    100
kidneybeans   100
chickpea      100
coffee       100
Name: label, dtype: int64
```

Figure 2: all of the label crops and counts

We check that there is any missing data and cleaning them. After cleaning, we get 2200 records. There are 22 types of crops that are apple, banana, black gram, chickpea, coconut, coffee, cotton, grapes, jute, kidney beans, lentil, maize, mango, moth beans, mung bean, muskmelon, orange, papaya, pigeon peas, pomegranate, rice and watermelon. For each crop, there are 100 records. So, this dataset is balance.



We find about the correlation between every pair of features.

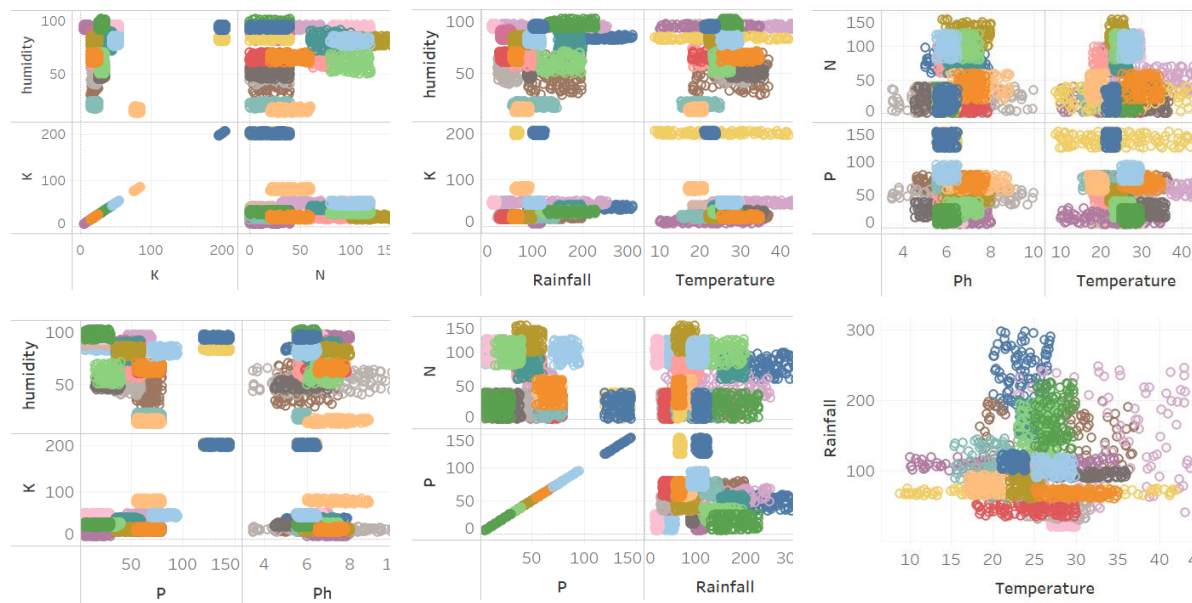
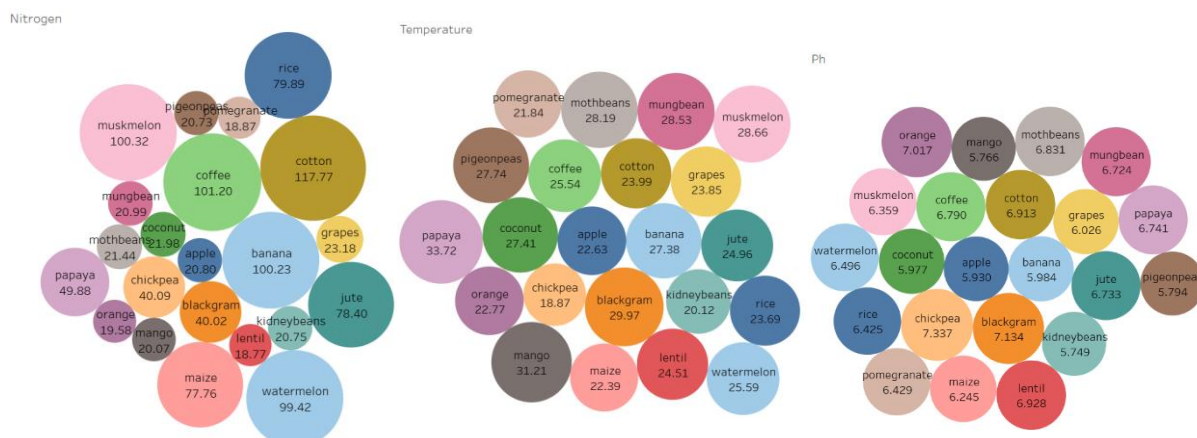


Figure 3: Correlation between each feature

Since there is no correlation between each other, we have to use all features for Data Mining Process.

We should understand about the information of each crop growing in each circumstance.



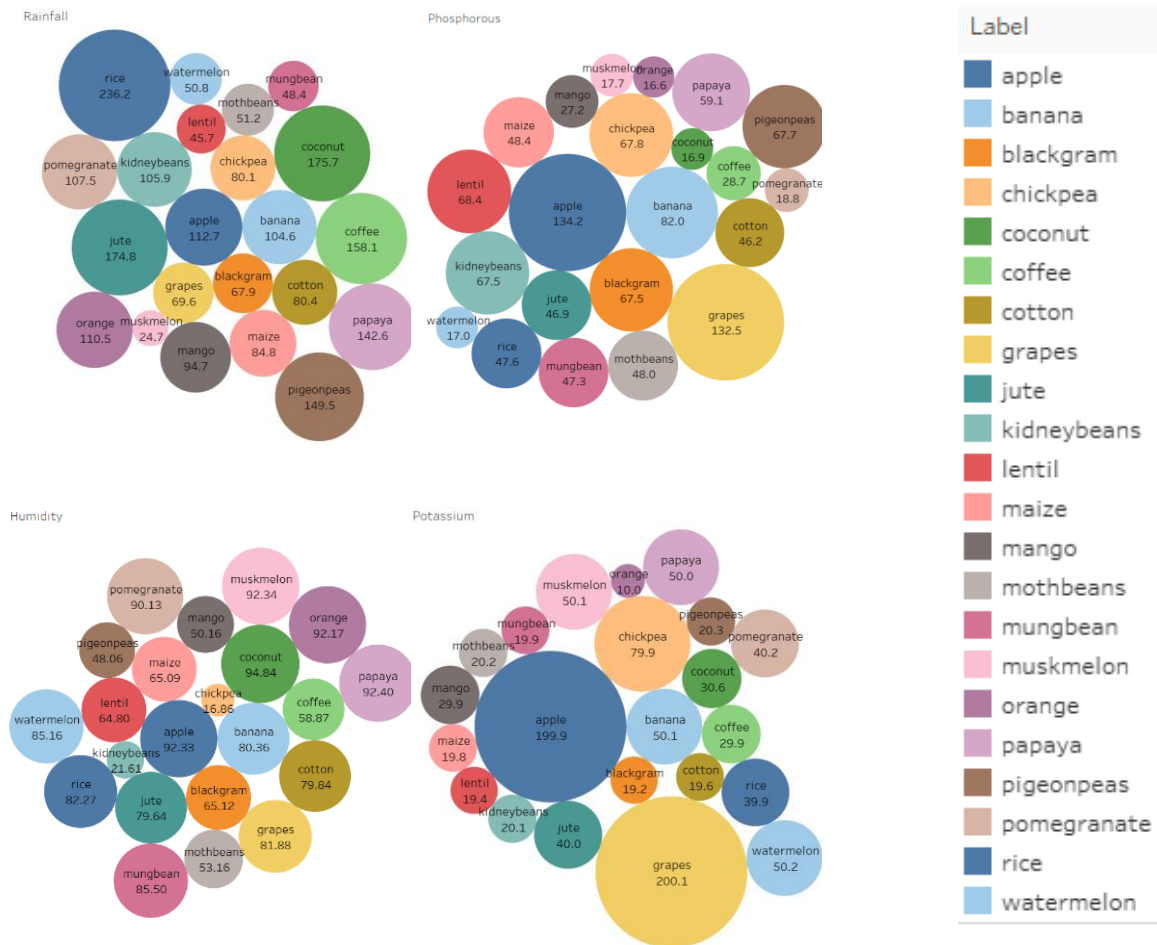


Figure 4: Features comparison for growing each type of crop

We extract the information of the distribution of each feature to more understand about the dataset.

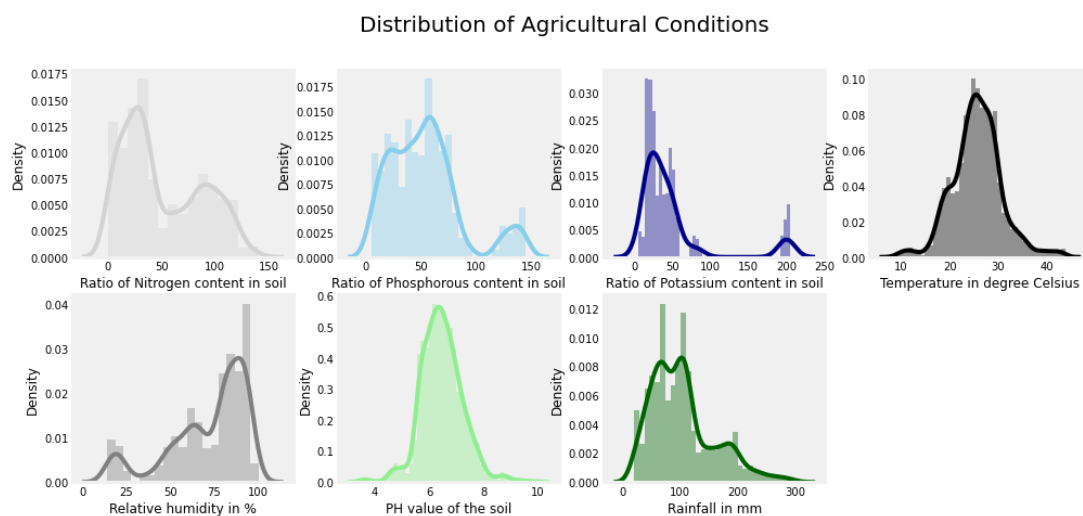


Figure 5: Distribution of all features in the dataset

According to Figure 5, we know about the most occurrences values of each feature.

# Methodology

## 3. Classification

### 3.1. Data preparation

To classify the data or to build a model for classification, we split the data into train set and test set with 80% and 20% of the original data respectively using data science python library called `sklearn`.

The following image shows the shape and size of each set.

```
The shape of x_train (1760, 7)
The shape of x_test (440, 7)
The shape of y_train (1760,)
The shape of y_test (440, 7)
```

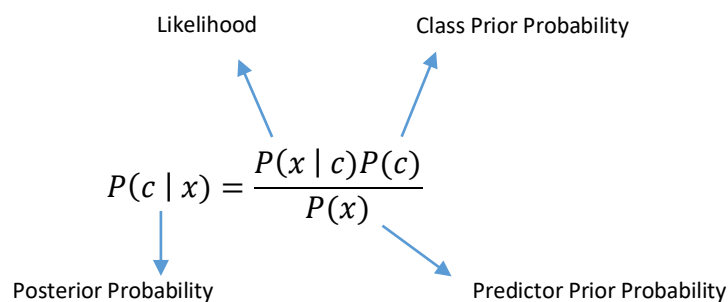
*Figure 6: Training and Testing Dataset size*

## 3.2. Implementation with Naïve Bayes Algorithm

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability  $P(c|x)$  from  $P(c)$ ,  $P(x)$  and  $P(x|c)$ . Look at the equation below:

$$P(c | x) = \frac{P(x | c)P(c)}{P(x)}$$


The diagram illustrates the components of the Naive Bayes equation. It shows the equation  $P(c | x) = \frac{P(x | c)P(c)}{P(x)}$  with four arrows pointing to its parts: 'Likelihood' points to  $P(x | c)$ , 'Class Prior Probability' points to  $P(c)$ , 'Posterior Probability' points to  $P(c | x)$ , and 'Predictor Prior Probability' points to  $P(x)$ .

$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$$

Above,

$P(c|x)$  is the posterior probability of *class* ( $c$ , *target*) given *predictor* ( $x$ , *attributes*).

$P(c)$  is the prior probability of *class*.

$P(x|c)$  is the likelihood which is the probability of *predictor* given *class*.

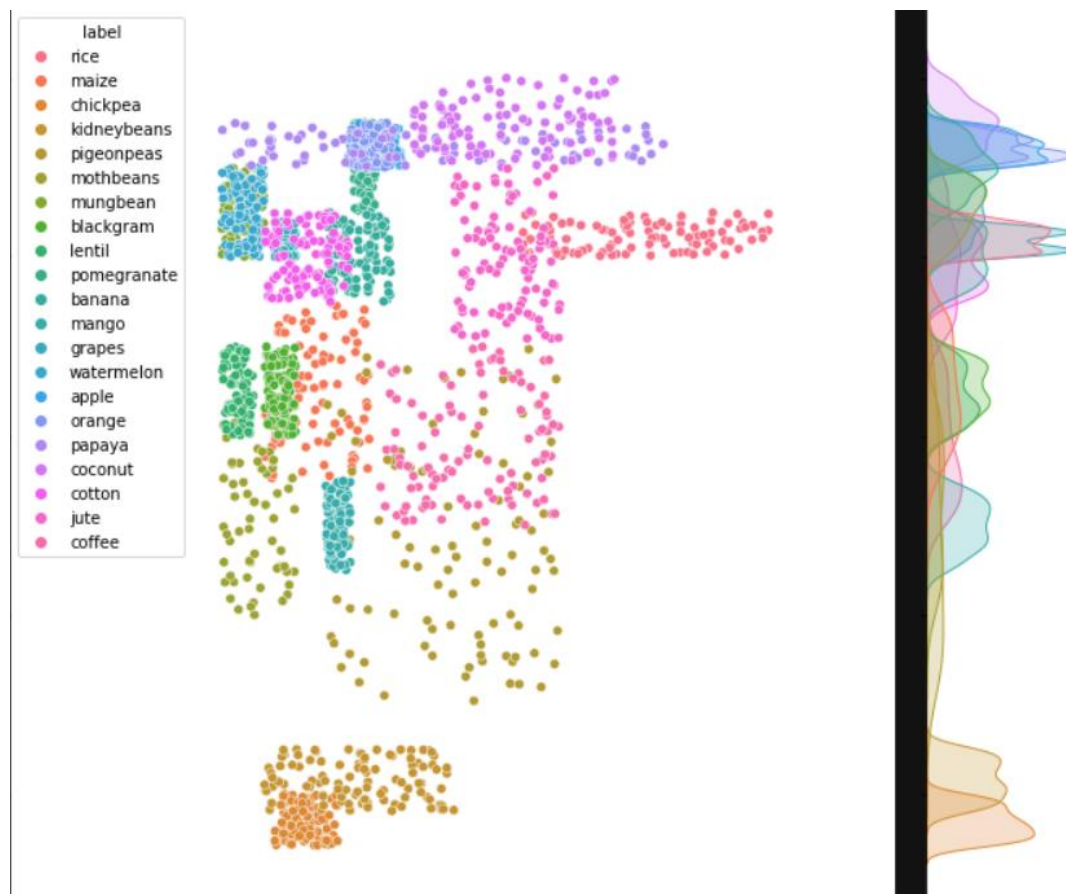
$P(x)$  is the prior probability of *predictor*.

### 3.2.1. Modelling

With our data set, we can classify the type of crop with *Naïve Bayes Algorithm*.

In this report, we use *Naïve Bayes Algorithm (Gaussian)* from `scikit-learn` library.

Before using Naïve Bayes Algorithm to classify the crops, we analyze the crop label information using scatter plot with the amount of rainfall attribute.



*Figure 7: amount of rainfall attribute*

Then we use *Gaussian Naïve Bayes Classifier Algorithm* to classify the crop labels with the help of N, P, K, Ph, temperature, rainfall and humidity attributes.

### 3.2.2. Performance Analysis

Here are the accuracy values of *Gaussian Naïve Bayes Algorithm* as shown in Figure.

```
GaussianNB()
Accuracy: 0.9931818181818182
Classification Report:
              precision    recall  f1-score   support

   apple          1.00        1.00        1.00        18
   banana          1.00        1.00        1.00        18
  blackgram          1.00        1.00        1.00        22
  chickpea          1.00        1.00        1.00        23
   coconut          1.00        1.00        1.00        15
   coffee          1.00        1.00        1.00        17
   cotton          1.00        1.00        1.00        16
   grapes          1.00        1.00        1.00        18
     jute          0.88        1.00        0.93        21
 kidneybeans          1.00        1.00        1.00        20
   lentil          1.00        1.00        1.00        17
   maize          1.00        1.00        1.00        18
   mango          1.00        1.00        1.00        21
  mothbeans          1.00        1.00        1.00        25
  mungbean          1.00        1.00        1.00        17
 muskmelon          1.00        1.00        1.00        23
   orange          1.00        1.00        1.00        23
  papaya          1.00        1.00        1.00        21
 pigeonpeas          1.00        1.00        1.00        22
 pomegranate          1.00        1.00        1.00        23
     rice          1.00        0.88        0.94        25
 watermelon          1.00        1.00        1.00        17

 accuracy                   0.99        440
  macro avg          0.99        0.99        0.99        440
  weighted avg          0.99        0.99        0.99        440
```

Figure 8: Accuracy, Precision, Recall, F1-score and Support of Gaussian Naïve Bayes model

From the above figure, we can confirm that our *Naïve Bayes* model accuracy is 0.99 so it is working well with unknown data and does not overfit the data.

### 3.3. Implementation with K-Nearest Neighbors Algorithm

The *k-nearest neighbors (KNN) algorithm* is a supervised learning algorithm that can be used to solve classification problems. The KNN algorithm assumes that similar things exist in close proximity.

#### 3.3.1 Tuning Hyperparameter

We try to find an optimal parameter by running the algorithm through  $k\_neighbors = 2$  to  $k\_neighbors = 40$ ; and plot the resulting error rate and accuracy on line charts.

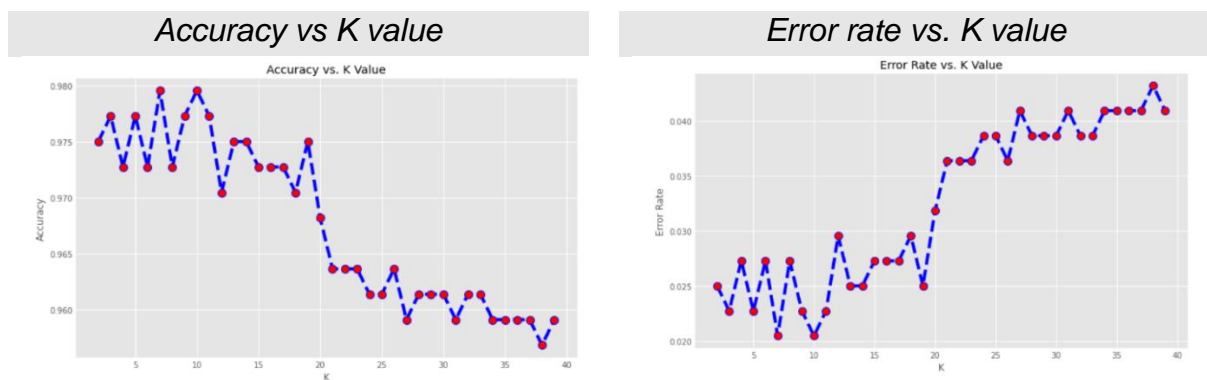


Figure 9: Accuracy and Error rate Calculated by K value

We observed a minimum error rate of 0.0205 and maximum accuracy of 0.9795 at  $k\_neighbors = 7$ . We skipped training with  $k\_neighbors = 1$  as this could lead to serious overfitting issues.

### 3.3.2 Modelling

#### Cross-validation with K-Fold

We then use K-Fold cross-validation with  $n\_splits = 10$  to estimate the skill of *the K-Nearest Neighbors model*.

In this report, we use *K-Neighbors Classifier* from `scikit-learn` library.

```
from sklearn.model_selection import KFold
k_folds = KFold(n_splits = 10, shuffle = True, random_state = 0)
for train_index, val_index in k_folds.split(x):

    X_train_KFOLD, X_val_KFOLD = x.iloc[train_index, :], x.iloc[val_index, :]
    y_train_KFOLD, y_val_KFOLD = y.iloc[train_index], y.iloc[val_index]

    knn = KNeighborsClassifier(n_neighbors=7)
    knn.fit(X_train_KFOLD, y_train_KFOLD)

    val_preds = knn.predict(X_val_KFOLD)
    val_accuracy = accuracy_score(y_val_KFOLD, val_preds)
    print(f'Accuracy Score: {val_accuracy}')
```

```
Accuracy Score: 0.9818181818181818
Accuracy Score: 0.9727272727272728
Accuracy Score: 0.9863636363636363
Accuracy Score: 0.9727272727272728
Accuracy Score: 0.9727272727272728
Accuracy Score: 0.9727272727272728
Accuracy Score: 0.9681818181818181
Accuracy Score: 0.9863636363636363
Accuracy Score: 0.9772727272727273
Accuracy Score: 0.9909090909090909
```

*Figure 10: Accuracy values for each K\_Fold*

From this, we can confirm that our model is working well with unseen data and does not overfit the data.

We choose the model with  $k\_neighbors = 7$  as this could avoid serious overfitting and get an overall best result.



### 3.3.3. Performance Analysis

Here are the accuracy values of K-Nearest Neighbor Algorithm with  $k\_neighbors = 7$  as shown in Figure.

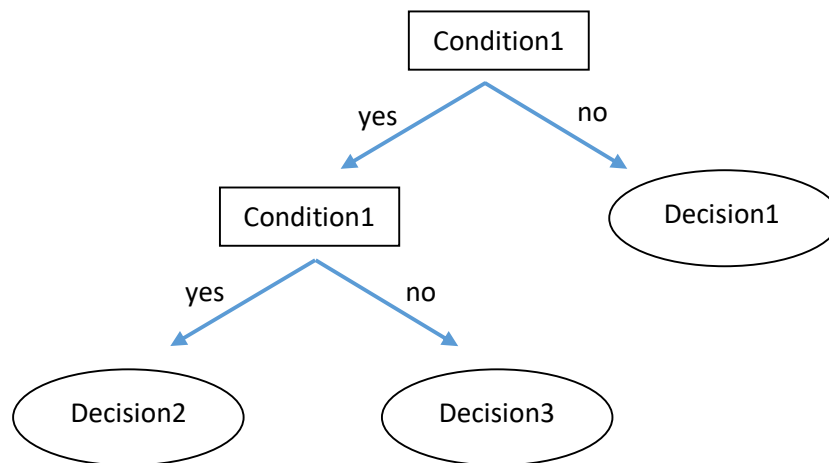
| Classification | Report:   |        |          |         |
|----------------|-----------|--------|----------|---------|
|                | precision | recall | f1-score | support |
| apple          | 1.00      | 1.00   | 1.00     | 18      |
| banana         | 1.00      | 1.00   | 1.00     | 18      |
| blackgram      | 1.00      | 1.00   | 1.00     | 22      |
| chickpea       | 1.00      | 1.00   | 1.00     | 23      |
| coconut        | 1.00      | 1.00   | 1.00     | 15      |
| coffee         | 1.00      | 1.00   | 1.00     | 17      |
| cotton         | 1.00      | 1.00   | 1.00     | 16      |
| grapes         | 1.00      | 1.00   | 1.00     | 18      |
| jute           | 0.79      | 0.90   | 0.84     | 21      |
| kidneybeans    | 0.91      | 1.00   | 0.95     | 20      |
| lentil         | 1.00      | 1.00   | 1.00     | 17      |
| maize          | 1.00      | 1.00   | 1.00     | 18      |
| mango          | 1.00      | 1.00   | 1.00     | 21      |
| mothbeans      | 1.00      | 1.00   | 1.00     | 25      |
| mungbean       | 1.00      | 1.00   | 1.00     | 17      |
| muskmelon      | 1.00      | 1.00   | 1.00     | 23      |
| orange         | 1.00      | 1.00   | 1.00     | 23      |
| papaya         | 1.00      | 1.00   | 1.00     | 21      |
| pigeonpeas     | 1.00      | 0.91   | 0.95     | 22      |
| pomegranate    | 1.00      | 1.00   | 1.00     | 23      |
| rice           | 0.91      | 0.80   | 0.85     | 25      |
| watermelon     | 1.00      | 1.00   | 1.00     | 17      |
| accuracy       |           |        | 0.98     | 440     |
| macro avg      | 0.98      | 0.98   | 0.98     | 440     |
| weighted avg   | 0.98      | 0.98   | 0.98     | 440     |

Figure 11: Accuracy, Precision, Recall, F1-score and Support of KNN for  $k\_neighbors=7$

From the above figure, we can confirm that our KNN model accuracy is 0.98 so it is working well with unknown data and does not overfit the data.

### 3.4. Implementation with Decision Tree Algorithm

A decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node or terminal node holds a class label. The topmost node in a tree is the root node.



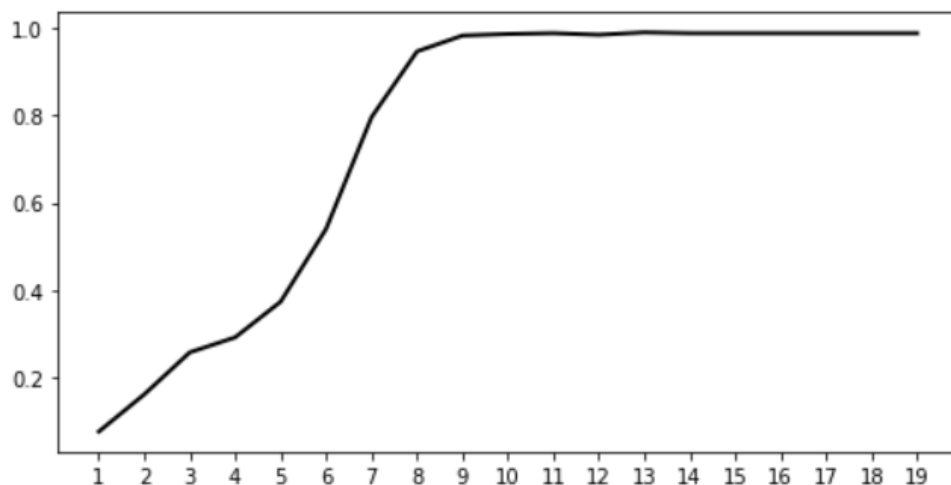
*Figure 12: Example Decision Tree*

#### Step of the Algorithm

1. Calculate the Information Gain of each feature.
2. Considering that all rows don't belong to the same class, split the dataset S into subsets using the feature for which the Information Gain is maximum.
3. Make a decision tree node using the feature with the maximum Information gain.
4. If all rows belong to the same class, make the current node as a leaf node with the class as its label.
5. Repeat for the remaining features until we run out of all features, or the decision tree has all leaf nodes.

### 3.4.1. Tuning Hyperparameter

In decision tree, number of max depths can be defined as a parameter to the model. Tree depth is a measure of how many splits it makes before coming to a prediction. A great choice of max depth depends on the accuracy of the model it can give within that depth. Too many depths or splits can over fit the training data whereas too less depth will not get good accuracy. Here we define maximum depth as an array which contain number ranging from 5 to 20 and will loop through each one, calculate the accuracy it can perform and save it in an array. Then we will choose the best max depth that can give best accuracy. Below is the figure of different accuracy calculated upon different max depth.



*Figure 13: Accuracy Calculated by Different Tree Max Depth*

Above figure explains that the highest accuracy we can achieve is at max depth 16. So, we choose 16 as our best parameter to train the model.

### 3.4.2. Modelling

In this report, we use *DecisionTreeClassifier* from `scikit-learn` library.

### 3.4.3. Performance Analysis

In analyzing the performance of the model, there are various method to see how accurately they can predict the unknown data or test data. Some of them are accuracy, confusion matrix, precision, recall, etc.

The figure below shows the precision, recall, f1-score and support the model.

|              | precision | recall | f1-score | support |
|--------------|-----------|--------|----------|---------|
| apple        | 1.00      | 1.00   | 1.00     | 24      |
| banana       | 1.00      | 1.00   | 1.00     | 24      |
| blackgram    | 1.00      | 0.96   | 0.98     | 26      |
| chickpea     | 0.93      | 1.00   | 0.97     | 28      |
| coconut      | 1.00      | 1.00   | 1.00     | 19      |
| coffee       | 1.00      | 1.00   | 1.00     | 24      |
| cotton       | 1.00      | 1.00   | 1.00     | 21      |
| grapes       | 1.00      | 1.00   | 1.00     | 24      |
| jute         | 0.96      | 0.82   | 0.88     | 28      |
| kidneybeans  | 1.00      | 0.91   | 0.95     | 23      |
| lentil       | 1.00      | 1.00   | 1.00     | 17      |
| maize        | 1.00      | 1.00   | 1.00     | 22      |
| mango        | 1.00      | 1.00   | 1.00     | 24      |
| mothbeans    | 0.97      | 1.00   | 0.98     | 29      |
| mungbean     | 1.00      | 1.00   | 1.00     | 27      |
| muskmelon    | 1.00      | 1.00   | 1.00     | 27      |
| orange       | 1.00      | 1.00   | 1.00     | 27      |
| papaya       | 0.93      | 1.00   | 0.97     | 28      |
| pigeonpeas   | 1.00      | 1.00   | 1.00     | 27      |
| pomegranate  | 1.00      | 1.00   | 1.00     | 28      |
| rice         | 0.90      | 0.97   | 0.93     | 29      |
| watermelon   | 1.00      | 1.00   | 1.00     | 24      |
| accuracy     |           |        | 0.98     | 550     |
| macro avg    | 0.99      | 0.98   | 0.98     | 550     |
| weighted avg | 0.98      | 0.98   | 0.98     | 550     |

Figure 14: Accuracy, Precision, Recall, F1-score and Support of Decision Tree model

From the above figure, we can confirm that our *Decision Tree* model accuracy is 0.98 so it is working well with unknown data and does not overfit the data.

### 3.5 Choosing the proper algorithm

We had built the classification models using *Naïve Bayes*, *K-Nearest Neighbor* and *Decision Tree algorithms* with accuracy 0.99, 0.98 and 0.98.

According to the *Figure 8*, all crops' results of precision, recall and f1-score of *Naïve Bayes algorithm* are 1, except jute and rice.

In the *Figure 11*, precision, recall and f1-score results of *K-Nearest Neighbor* of jute, kidney beans, pigeon peas and rice are not completely 1 but that of others are 1.

Shown in *Figure 14*, the results of precision, recall and f1-score of *Decision Tree* of black gram, chickpea, jute, kidney beans, moth beans, papaya and rice are not completely 1 but that of others are 1.

We have to choose the model with highest all accuracies. So, we choose *Naïve Bayes* classifier model. We can be able to use this model to predict crop label based on 7 features: N, P, K, Rainfall, Temperature, Humidity and PH value of soil.

## 4. Clustering

### 4.1. Implementation with K-Means

In unsupervised learning, clustering technique is very popular. We just start by choosing a random number of K of data points from our sample; these represent the initial centroids, the cluster centers, and their numbers equal the number of clusters. Then, we allocate each sample to the closest cluster center. This step can be done by calculating the *Euclidean distance* between all the random cluster centers and any other data point. Finally, this can be generated until there is no re-assignment of the samples to the cluster centroids.

#### Euclidean Distance between two points in space:

If  $p = (p_1, p_2)$  and  $q = (q_1, q_2)$ , then the distance is given by

$$d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$$

We then assign each data point to the cluster center closest to it.

#### Assigning each point to the nearest cluster:

If each cluster centroid is denoted by  $c_i$ , then each data point  $x$  is assigned to a cluster based on

$$\arg \min_{c_i \in C} \text{dist}(c_i, x)^2$$

Here  $\text{dist}()$  is the Euclidean distance

Finding the new centroid from the clustered group of points:

$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i$$

$S_i$  is the set of all points assigned to the  $i^{\text{th}}$ -cluster.

Since we started with choosing random of K clusters of data points, it will not give us great and efficient results. So, we repeat the process and instead of using random initial centroids from our data points, we will calculate the actual cluster centers using the following techniques to get the optimal number of clusters centers.

### 4.1.1. Tuning Hyperparameter

#### Elbow Method

This method provides us a good k number of clusters would be based on the *sum of squared distance (SSE)* between data points and their assigned clusters centroids.

#### Silhouette Analysis

We use this method to find the degree of separation between clusters. For each sample:

- Calculate the mean distance from all data points in the same cluster ( $a^i$ ).
- Calculate the mean distance from all data points in the closest cluster ( $b^i$ ).
- Compute the coefficient:

$$\frac{b^i - a^i}{\max(a^i, b^i)}$$

The coefficient can take values in the interval  $[-1, 1]$ . If:

- It equals 0  $\rightarrow$  the sample is near the neighboring clusters.
- It equals 1  $\rightarrow$  the sample is far from the neighboring clusters.
- It equals -1  $\rightarrow$  the sample is allocated in the wrong clusters.

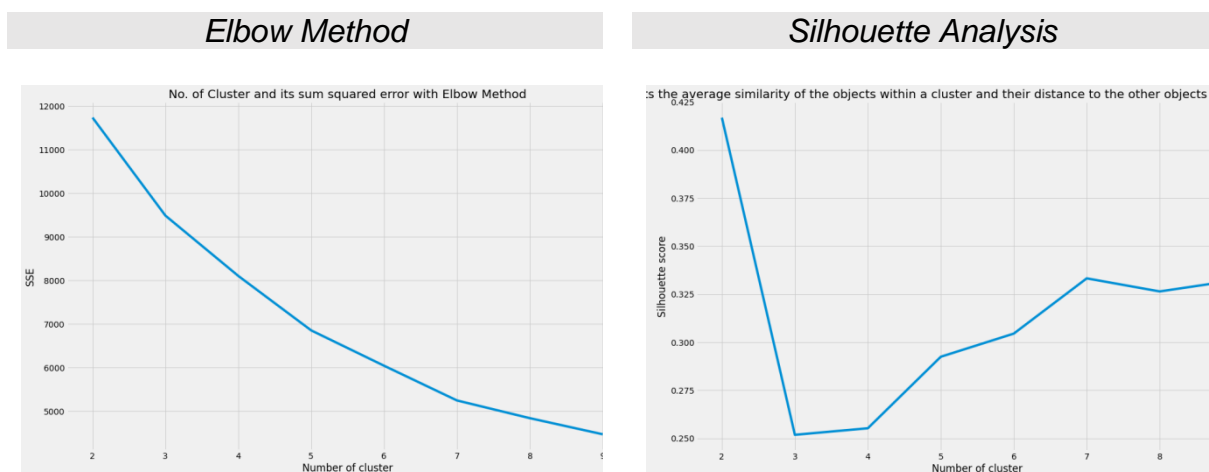


Figure 15: SSE and Silhouette score with Calculated by K value

*Elbow method* doesn't show the exact number of clusters. But from this, we choose k be 5. *Silhouette method* shows that 2 clusters will give the best quality.

We find each cluster items for  $k = 2$  and  $k = 5$ .

### 4.1.2. Modelling

We build clustering model with  $k = 2$  and  $k = 5$ .

In this report, we use *KMeans* from `scikit-learn` library.

Kmean: Number of Cluster 2

|         |  | Crops |
|---------|--|-------|
| Cluster |  |       |
| 0       | [rice, maize, chickpea, kidneybeans, pigeonpeas, mothbeans, mungbean, blackgram, lentil, pomegranate, banana, mango, watermelon, muskmelon, orange, papaya, coconut, cotton, jute, coffee] |       |
| 1       | [grapes, apple]  |       |

Kmean: Number of Cluster 5

|         |  | Crops |
|---------|--|-------|
| Cluster |  |       |
| 0       | [rice, pigeonpeas, pomegranate, orange, papaya, coconut, jute, coffee]             |       |
| 1       | [maize, banana, watermelon, muskmelon, papaya, cotton, jute, coffee]               |       |
| 2       | [maize, pigeonpeas, mothbeans, mungbean, blackgram, lentil, mango, orange, papaya] |       |
| 3       | [grapes, apple]  |       |
| 4       | [maize, chickpea, kidneybeans, pigeonpeas, lentil]                                 |       |

*Figure 16: Crops that belong to each cluster using  $k=2$  and  $k=5$*



### 4.1.2. Performance Analysis

We measure the quality of cluster no 2 and 5 with intra class similarity and inter class similarity.

Kmean: No of Cluster 2  
Intra class similarity

| Cluster | Distance    |
|---------|-------------|
| 0       | 0 98.754348 |
| 1       | 1 44.408691 |

Inter class similarity

| Cluster | Distance          |
|---------|-------------------|
| 0       | (0, 1) 206.116068 |

Figure 17: Intra class and Inter class similarity with  $k=2$

Model with  $k = 2$  is better than that with  $k = 5$  because all intra class similarities of model with  $k = 2$  are much lower than the inter class similarity of its while some of inter class similarity of model with  $k = 5$  is greater than its intra class similarity.

Kmean: No of Cluster 5  
Intra class similarity

| Cluster | Distance    |
|---------|-------------|
| 0       | 0 81.942159 |
| 1       | 1 84.386560 |
| 2       | 2 70.304776 |
| 3       | 3 44.408691 |
| 4       | 4 81.463450 |

Inter class similarity

| Cluster | Distance          |
|---------|-------------------|
| 0       | (0, 1) 99.010272  |
| 1       | (0, 2) 96.559392  |
| 2       | (0, 3) 213.548427 |
| 3       | (0, 4) 104.568930 |
| 4       | (1, 2) 95.515990  |
| 5       | (1, 3) 206.778016 |
| 6       | (1, 4) 101.022524 |
| 7       | (2, 3) 203.657457 |
| 8       | (2, 4) 73.080972  |
| 9       | (3, 4) 195.025973 |

Figure 18: Intra class and Inter class similarity with  $k=5$

Also according to the correlation *Figure 3*, we can prove that why we get this cluster.

In above figure , shows that in most correlation, apples and grapes are nearly the same and difference from others.

## **4.2. Usage of clustering results**

### **Recommendation System for farmers**

Using the results of clustering, we would help to improve the income of farmers by recommending the same groups of crops which are farmers already grown based on the market trend.

For example, support that there is a farmer who grows apples. And there is a future prediction that shows the grapes are at the high demand in this season. Then in addition to apples, the system recommends growing grapes because apples and grapes are in the same group according to soil nutrients and climate change.

### **Targeting Markets for Market Researchers**

For the market researchers, we can support that a product that is marketing in one field, can also do marketing in another fields.

For example, there is a company that manufactures a new fertilizer for apples and it started to marketing. Then we can suggest that this fertilizer can also useful for grapes. It helps researchers to try to expand the market.

The results of clustering can be used in many other applications like for marketing and sales, for crop demand predictions, etc.

## 5. Conclusion

This report helps Myanmar farmers to choose the right crop by providing insights that ordinary farmers don't keep track of thereby decreasing the chances of crop failure and increasing productivity. The productivity gets improved with sustained research in the field of spatial data mining to realize precision agriculture. It also prevents them from incurring losses. The report can be extended to the web and can be accessed by millions of farmers across the country. By using classification analysis, we could achieve the accuracy of 99.3 percent from Naïve Bayes algorithm, the accuracy of 97.4 percent from K-Nearest Neighbors Algorithm and the accuracy of 98 percent from Decision Tree Algorithm. By clustering analysis, we recommend that apple and grape can grow together for some state of soil health and weather condition.

## 6. References

- [1] Dabbura, I. (2019, September 3). K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks. Retrieved from - <https://towardsdatascience.com/kmeans-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a>
- [2] T.R. Lekhaa, "Efficient Crop Yield and Pesticide Prediction for Improving Agricultural Economy using Data Mining Techniques", International Journal of Modern Trends in Engineering and Science (IJMTES), 2016, Volume 03, Issue 10
- [3] Athvika, S. (2020, February 28). Crop Yield Prediction. Retrieved from <https://www.kaggle.com/srprojects/crop-yield-prediction>
- [4] Krishnan, M. (2018, July 7). Mathematics behind K-Mean Clustering algorithm. Retrieved from <https://muthu.co/mathematics-behind-k-mean-clustering-algorithm/>
- [5] <https://www.fao.org/myanmar/fao-in-myanmar/myanmar/en/>
- [6] <https://www.worldbank.org/en/country/myanmar/publication/myanmar-analysis-of-farm-production-economics>
- [7] <https://www.fao.org/global-soil-partnership/areas-of-work/soil-fertility/en/>
- [8] <https://www.agrocares.com/2020/11/02/npk-what-is-it-and-why-is-it-so-important/>
- [9] <https://www.dtn.com/the-role-of-weather-and-weather-forecasting-in-agriculture/>
- [10] <https://climatechange.chicago.gov/climate-impacts/climate-impacts-agriculture-and-food-supply>

# Appendix

Import pandas and Numpy for data manipulations. And then we use Matplotlib, Seaborn and Sklearn for data visualization.

|    | A  | B  | C  | D        | E        | F        | G        | H     |
|----|----|----|----|----------|----------|----------|----------|-------|
| 1  | N  | P  | K  | temperat | humidity | ph       | rainfall | label |
| 2  | 90 | 42 | 43 | 20.87974 | 82.00274 | 6.502985 | 202.9355 | rice  |
| 3  | 85 | 58 | 41 | 21.77046 | 80.31964 | 7.038096 | 226.6555 | rice  |
| 4  | 60 | 55 | 44 | 23.00446 | 82.32076 | 7.840207 | 263.9642 | rice  |
| 5  | 74 | 35 | 40 | 26.4911  | 80.15836 | 6.980401 | 242.864  | rice  |
| 6  | 78 | 42 | 42 | 20.13017 | 81.60487 | 7.628473 | 262.7173 | rice  |
| 7  | 69 | 37 | 42 | 23.05805 | 83.37012 | 7.073454 | 251.055  | rice  |
| 8  | 69 | 55 | 38 | 22.70884 | 82.63941 | 5.700806 | 271.3249 | rice  |
| 9  | 94 | 53 | 40 | 20.27774 | 82.89409 | 5.718627 | 241.9742 | rice  |
| 10 | 89 | 54 | 38 | 24.51588 | 83.53522 | 6.685346 | 230.4462 | rice  |

Figure1: CSV view of Crop dataset

```

1 def KMean_cluster(X, no_clusters_list):
2     cluster_crop = []
3     for k in no_clusters_list:
4         km = KMeans(n_clusters = k, max_iter = 300, n_init = 10, random_state = 0)
5         y_kmeans = km.fit_predict(X)
6         y_kmeans = pd.DataFrame(y_kmeans)
7         z = pd.concat([y_kmeans, X], axis = 1)
8         z = z.rename(columns = {0: 'cluster'})
9
10        cluster = []
11        for x in range(0,k):
12            rec = {
13                'Cluster': x,
14                'Crops': z[z['cluster']==x]['label'].unique()
15            }
16            cluster.append(rec)
17        cluster_df = pd.DataFrame(cluster)
18        cluster_df.set_index('Cluster', inplace=True)
19
20        cluster_crop.append(cluster_df)
21    return cluster_crop

```

```

1 cluster_crop = KMean_cluster(X_sc, k)

```

Figure2: Python using Jupyter Notebook Tool



Figure3: Tableau for Data Visualization