Crop Recommendation System

Project Report

Bachelor

in

Computer Science

By

Basireddy Sreehitha

Rajiv Gandhi University Of Knowledge And Technologies

S.M. Puram , Srikakulam -532410

Andhra Pradesh, India

Abstract

Agriculture plays a crucial role in Indian economy and employment. Half of the country's population is still employed on agriculture sector. India is one of the largest producers of agricultural products, but still it has less farm production. The most common problem faced by the farmers is that they do not opt the crop based on the necessity of soil. This problem can be solved by using crop recommendation system.

A crop recommendation system is a project that helps the farmers to enhance their crop production by recommending the best crops based on soil properties. This system uses machine learning algorithms to analyse the data collected from several sources. Based on this analysis,the system can suggest the best crop to grow in a particular area,by taking the factors such as soil type,temperature,rainfall etc. It helps to improve crop yields and reduce waste. It helps the farmers to save time and money in reducing the non suitable crop selection. It also helps the farmers to take better decisions for farming.

This system helps to implement the crop selection method so that this method helps in solving many agriculture and farmers problems. This improves our Indian economy by maximizing the yield rate of crop production.

Contents

Abstract 1

1 Introduction 4

1.1 Introduction of Crop Recommendation System 4

1.2 Application 5

1.3 Motivation Towards Crop Recommendation System 5

1.4 Problem Statement 6

2 Approach to Crop Recommendation System 7

2.1 About Crop Recommendation System 7

2.2 Data Set 8

2.3 Prediction techniques 8

2.4 Graphs 9

2.4.1 Bar Graph 9

2.4.2 Scatter Plot 10

2.4.3 Histogram 10

2.4.4 Box Plot 11

2.5 Visualization 11

2.5.1 Pair Plot 11

2.5.2 Bar Graph 12

2.5.3 Box Plot 12

2.5.4 3D Graph 13

2.5.5 Pie Chart 14

3 Code **15**

3.1 Code Explanation 15

3.1.1 Importing Necessary Libraries: 15

3.1.2 Data Understanding: 15

3.1.3 Data Preprocessing: 18

4 Conclusion and Future Work **25**

4.0.1 Conclusion 25

4.0.2 Future Work 26

Chapter 1

Introduction

1.1 Introduction of Crop Recommendation System

In India,farming is not considered as a business but also has a huge impact on the social life of people which are associated to it. There are many festivals and social gatherings are celebrated in accordance with the different seasons and practices involved in farming. Hence large part of population is dependent on agriculture field directly or indirectly.Most of the times it is observed that farmers tend to saw the crop according to its market value and possible financial profits rather than taking factors like soil conditions,weather conditions etc., into the account.This may lead to undesirable results for farmers and for the nature of soil too.

In today's time,technologies like machine learning and deep learning can become the game changers in such fields if they are used in proper manner.This project aim is to recommend the most suitable crop based on input parameters like Nitrogen(N),Phosphorus(P), Potassium(K),pH value of the soil,Humidity,Temperature and Rainfall.This project predicts the accuracy of the future prodution of nine different crops such as pomegranate,banana, mango,grapes,watermelon,muskmelon,apple,orange,papaya crops using various supervised machine learning approches and recommends the most suitable crop.

This dataset contains various parameters like Nitrogen(N),Phosphorus(P),Potassium(K), PH value of the soil,Humidity,Temperature and Rainfall. This proposed system applied different kinds of machine learning algorithms like K-Nearest Neighbours(K-NN),Decision tree,Support Vector Machine(SVM) to predict the suitable crop.

1.2 Application

A crop recommedation system is a valuable application that can help farmers in making better decisions about which crops to grow based on various factors such as soil quality and climate conditions. It can suggest suitable crop for a specific region or field based on the parameters it helps farmers select crops that have higher chances of success and profitability. By recommending crops that are well suited to the conditions farmers can maximize crop productivity. It leads to better yeilds.It helps farmers optimize the use of fetilizers,water etc.,and reducing wastage.

1.3 Motivation Towards Crop Recommendation System

Agriculture plays a crucial role in Indian economy and employment. Half of the country's population is still employed on agriculture sector. India is one of the largest producers of agricultural products, but still it has less farm production. The most common problem faced by the farmers is that they do not opt the crop based on the necessity of soil. This problem can be solved by using crop recommendation system.

A crop recommendation system project aim is to recommend the most suitable crop based on input parameters like Nitrogen(N),Phosphorus(P), Potassium(K),pH value of the soil,Humidity,Temperature and Rainfall.This project predicts the accuracy of the future prodution of nine different crops such as pomegranate,banana,mango,grapes,watermelon, muskmelon,apple,orange,papaya crops using various supervised machine learning approches and recommends the most suitable crop.

1.4 Problem Statement

There are very few platforms that helps farmers with their farming strategy and crop recommedation is one of them. Farmer understimate the fertility of the soil on their farms.The most common problem faced by the farmers is that they do not opt the crop based on the necessity of soil. This problem can be solved by using crop recommendation system.Using appropriate parameters like rainfall,temperature,humidity,nutrient levels etc.This system can predict the suitable crop for a specific field.

Chapter 2

Approach to Crop Recommendation System

2.1 About Crop Recommendation System

Agriculture plays a major role in Indian economy and employment. India is one of the largest producers of agricultural products, but still it has less farm production. The most common problem faced by the farmers is that they do not opt the crop based on the necessity of soil. This problem can be solved by using crop recommendation system.

A crop recommendation system helps the farmers to enhance their crop production by recommending the best crops based on soil properties. This system uses machine learning algorithms to analyse the data collected from several sources. Based on this analysis,the system can suggest the best crop to grow in a particular area,by taking the factors such as nutrient levels,temperature,rainfall,pH value,humidity etc. It helps to improve crop yields and reduce waste. It helps the farmers to save time and money in reducing the non suitable crop selection. It also helps the farmers to take better decisions for farming. This improves our Indian economy by maximizing the yield rate of crop production.

2.2 Data Set

The dataset consists of parameters like Nitrogen(N),Phosphorus(P), Potassium(K),pH value of the soil,Humidity,Temperature and Rainfall. The dataset have been obtained from the Kaggle website. The dataset has 2708 instance or data that have taken from the past historic data. This dataset includes nine different crops such as pomegranate,banana, mango,grapes,watermelon,muskmelon,apple,orange and papaya.

Features of dataset

N: ratio of Nitrogen content in soil

P: ratio of Phosphorus content in soil

K: ratio of Potassium content in soil

Temperature: temperature in degree celsius

Humidity: relative humidity in %

pH: pH value of the soil

rainfall: rainfall in mm

Data Cleaning and preprocessing

The first step is to make sure that the dataset we are using is accurate. The dataset should not have any missing values and if the dataset have missing values,they should be replaced by appropriate values by using mean,median or mode of the entire column.

2.3 Prediction techniques

K-Nearest Neighbour(KNN)

K-Nearest Neighbour is one of the simplest machine learning algorithms based on supervised learning technique.K-NN algorithm assumes the similarity between the new case and available cases and puts the new case into category that is most similar to available categories.K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.It is also called as a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification,it performs an action on the dataset.

Support Vector Machines(SVM)

Support Vector Machine(SVM) is one of the most popular Supervised Learning algorithms,which is used for Classification as well as Regression problems.It is used mostly for Classification problems in Machine Learning.The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future.

2.4 Graphs

2.4.1 Bar Graph

```
x=df["label"]
y=df["rainfall"]
fig = plt.figure(figsize=(10,10))
plt.xlabel("label")
plt.ylabel("rainfall")
plt.title("Bar_Graph")
plt.bar(x,y,width=0.7)
plt.show()
```

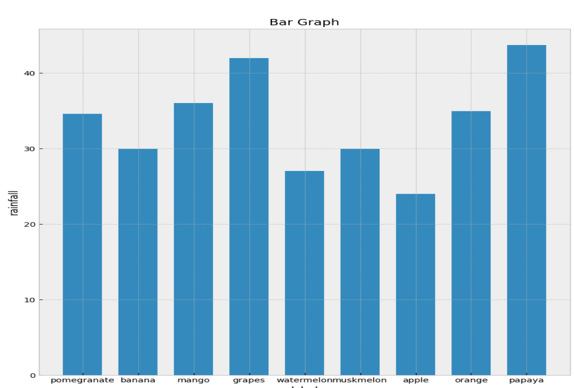


Figure 2.1: Bar Graph

2.4.2 Scatter Plot

```
a=df["rainfall"]
b=df["humidity"]
plt.xlabel("rainfall")
plt.ylabel("humidity")
plt.title("Scatter Plot")
plt.scatter(a,b,color="green")
```

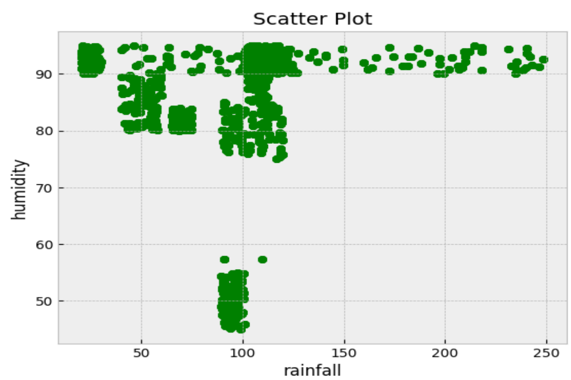


Figure 2.2: Scatter Plot

2.4.3 Histogram

```
plt.xlabel("temperature")
plt.title("Histogram")
df["temperature"].plot(kind="hist")
```

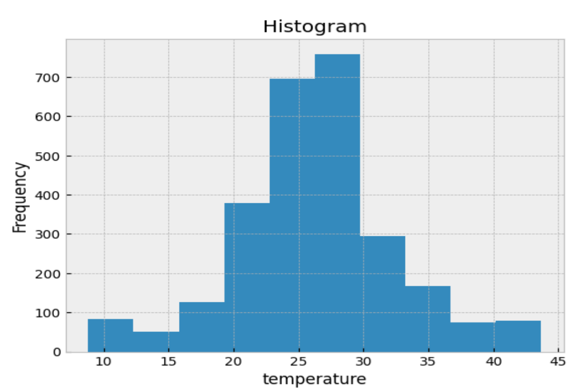


Figure 2.3: Histogram

2.4.4 Box Plot

```
x=df["N"]
plt.xlabel("N")
plt.title("Box Plot")
plt.boxplot(x,notch=True,patch_artist=True)
```

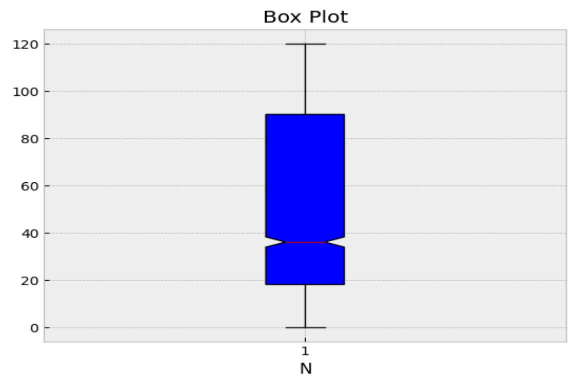


Figure 2.4: Box Plot

2.5 Visualization

2.5.1 Pair Plot

```
import seaborn as sns
sns.pairplot(df)
```

Figure 2.5: Pair Plot

2.5.2 Bar Graph

```
fig=plt.figure(figsize=(10,6))
ax=fig.add_axes([0,0,1,1])
sns.barplot(x="label",y="rainfall",data=df)
plt.title("graph showing the average rainfall for each label")
```

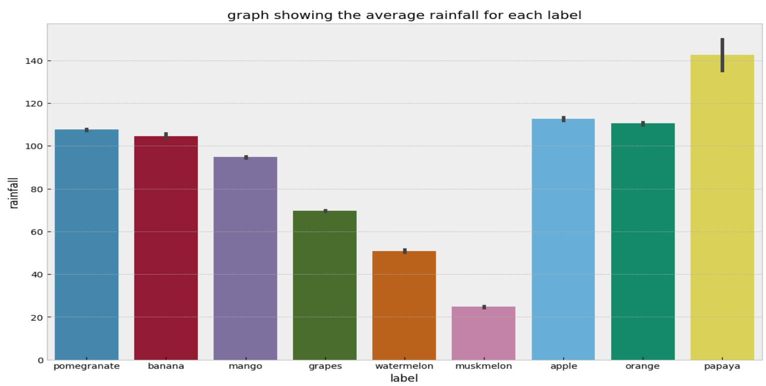


Figure 2.6: Bar Graph

2.5.3 Box Plot

```
fig = plt.figure(figsize=(20,15))
ax=fig.subplots(3,3)
sns.boxplot(x="label",y="N",data=df,ax=ax[0,0])
ax[0,0].set_xticklabels(ax[0,0].get_xticklabels(), rotation=90)

sns.boxplot(x="label",y="P",data=df,ax=ax[0,1])
ax[0,1].set_xticklabels(ax[0,1].get_xticklabels(), rotation=90)

sns.boxplot(x="label",y="K",data=df,ax=ax[0,2])
ax[0,2].set_xticklabels(ax[0,2].get_xticklabels(), rotation=90)

sns.boxplot(x="label",y="temperature",data=df,ax=ax[1,0])
ax[1,0].set_xticklabels(ax[1,0].get_xticklabels(), rotation=90)

sns.boxplot(x="label",y="humidity",data=df,ax=ax[1,1])
ax[1,1].set_xticklabels(ax[1,1].get_xticklabels(), rotation=90)
```

```
sns.boxplot(x="label",y="ph",data=df,ax=ax[1,2])
ax[1,2].set_xticklabels(ax[1,2].get_xticklabels(), rotation=90)

sns.boxplot(x="label",y="temperature",data=df,ax=ax[2,0])
ax[2,0].set_xticklabels(ax[2,0].get_xticklabels(), rotation=90)

plt.delaxes(ax[2,1])
plt.delaxes(ax[2,2])
plt.show()
```

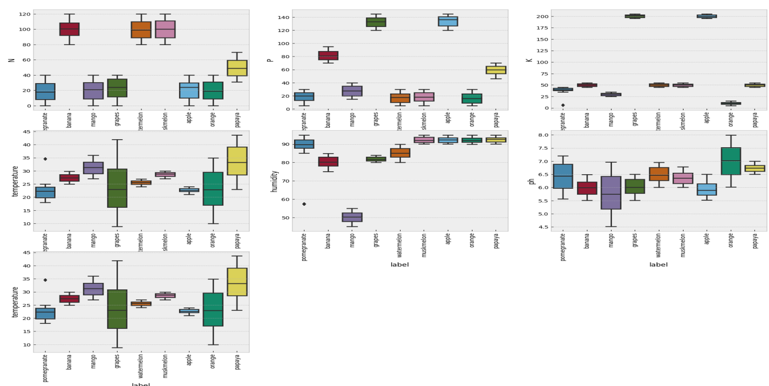


Figure 2.7: Box Plot

2.5.4 3D Graph

```
data = df.sort_values(["temperature"])
x,y,z=data["temperature"],data["rainfall"],data["humidity"]
fig=plt.figure(figsize=(5,6))
ax2=fig.add_axes([0,0,1,1],projection="3d")
ax2.scatter3D(x,y,z,alpha=0.4)
ax2.set_xlabel("temperature")
ax2.set_ylabel("rainfall")
ax2.set_zlabel("humidity")
plt.show()
ax2.view_init(0,40)
```

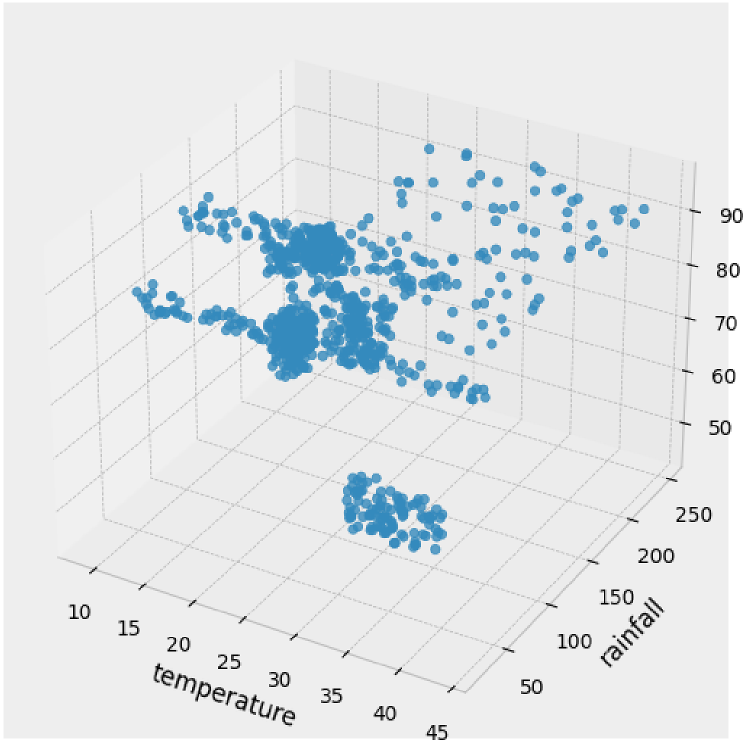


Figure 2.8: 3D Graph

2.5.5 Pie Chart

```
explode=[0]*9
explode[0]=0.2
plt.pie(data["rainfall"],shadow=True,explode=explode,labels=data["label"],autopct="%1.0f%%",
textprops={"color":"white"})
plt.legend(bbox_to_anchor=[1,0,0.5,1])
plt.show()
```

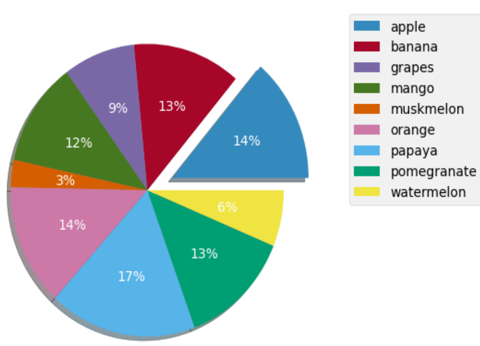


Figure 2.9: Pie Chart

Chapter 3

Code

3.1 Code Explanation

Crop Recommendation System

3.1.1 Importing Necessary Libraries:

```
import pandas as pd
import matplotlib.pyplot as plt
import sklearn
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.svm import SVC
import seaborn as sns
```

3.1.2 Data Understanding:

```
df=pd.read_csv("cropdata.csv")
df
```

Figure 3.1: Loading the dataset

	N	P	K	temperature	humidity	ph	rainfall	label
0	NaN	NaN	NaN	34.567890	57.402477	NaN	NaN	pomegranate
1	NaN	25.445813	NaN	NaN	57.402477	NaN	NaN	pomegranate
2	NaN	NaN	6.497367	NaN	57.402477	NaN	NaN	pomegranate
3	NaN	NaN	NaN	NaN	57.402477	NaN	109.416919	pomegranate
4	2.0	24.000000	38.000000	24.559816	91.635362	5.922936	111.968462	pomegranate
...	...	...	...	...	...	...	...	...
2703	42.0	59.000000	55.000000	40.102077	94.351102	6.979102	149.119999	papaya
2704	43.0	64.000000	47.000000	38.589545	91.580765	6.825665	102.270823	papaya
2705	35.0	67.000000	49.000000	41.313301	91.150880	6.617067	239.742755	papaya
2706	56.0	59.000000	55.000000	37.035519	91.794302	6.551893	188.518142	papaya
2707	39.0	64.000000	53.000000	23.012402	91.073555	6.598860	208.335798	papaya

2708 rows × 8 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2708 entries, 0 to 2707
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   N            2700 non-null   float64
 1   P            2702 non-null   float64
 2   K            2702 non-null   float64
 3   temperature  2702 non-null   float64
 4   humidity     2708 non-null   float64
 5   ph           2700 non-null   float64
 6   rainfall     2702 non-null   float64
 7   label        2708 non-null   object
dtypes: float64(7), object(1)
memory usage: 169.4+ KB
```

Figure 3.2: Data info

dtype: float64(7), object(1))

Dataframe contained 8 null values and the label is object

```
df.corr()
```

It calculates the relationship between each column.

	N	P	K	temperature	humidity	ph	rainfall
N	1.000000	-0.288102	-0.325028	0.129913	0.144062	0.032565	-0.460399
P	-0.288102	1.000000	0.860912	-0.128504	0.049062	-0.339588	0.255382
K	-0.325028	0.860912	1.000000	-0.248526	0.125457	-0.320848	-0.019322
temperature	0.129913	-0.128504	-0.248526	1.000000	-0.147927	0.030738	0.118974
humidity	0.144062	0.049062	0.125457	-0.147927	1.000000	0.446184	0.003235
ph	0.032565	-0.339588	-0.320848	0.030738	0.446184	1.000000	0.077294
rainfall	-0.460399	0.255382	-0.019322	0.118974	0.003235	0.077294	1.000000

```
df.shape()
```

Output:

(2708, 8)

The dataset contains 2708 rows and 8 columns

```
df.describe()
```

It returns the description of the data in the dataframe

	N	P	K	temperature	humidity	ph	rainfall
count	2700.000000	2702.000000	2702.000000	2702.000000	2708.000000	2700.000000	2702.000000
mean	50.261111	56.089523	75.563285	26.411413	84.023964	6.305336	90.869908
std	38.267360	47.057459	67.742092	6.151605	13.059421	0.558612	40.550861
min	0.000000	5.000000	5.000000	8.825675	45.022364	4.507524	20.211267
25%	18.000000	18.000000	38.000000	22.918432	81.521204	5.931906	65.800060
50%	36.000000	30.000000	49.000000	26.288460	89.901470	6.293941	98.540477
75%	90.000000	88.000000	55.000000	29.457717	92.132298	6.677046	110.251110
max	120.000000	145.000000	205.000000	43.675493	94.998975	7.995849	248.859299

```
df.head()
```

It returns first five values

]:

	N	P	K	temperature	humidity	ph	rainfall	label
0	NaN	NaN	NaN	34.567890	57.402477	NaN	NaN	pomegranate
1	NaN	25.445813	NaN	NaN	57.402477	NaN	NaN	pomegranate
2	NaN	NaN	6.497367	NaN	57.402477	NaN	NaN	pomegranate
3	NaN	NaN	NaN	NaN	57.402477	NaN	109.416919	pomegranate
4	2.0	24.000000	38.000000	24.559816	91.635362	5.922936	111.968462	pomegranate

3.1.3 Data Preprocessing:

```
df.duplicated()
```

Checking of duplicate values

```
: 4       False
  5       False
  6       False
  7       False
  8       False
          ...
  2703     True
  2704     True
  2705     True
  2706     True
  2707     True
  Length: 2700, dtype: bool
```

It shows that our dataset contain some duplicate values.Next process is to replace the null values

Replacing the null values

```
x=df["N"].mean()
df["N"].fillna(x,inplace=True)
x=df["P"].mean()
df["P"].fillna(x,inplace=True)
x=df["K"].mean()
```

```
df["K"].fillna(x,inplace=True)
x=df["temperature"].mean()
df["temperature"].fillna(x,inplace=True)
x=df["humidity"].mean()
df["humidity"].fillna(x,inplace=True)
x=df["ph"].mean()
df["ph"].fillna(x,inplace=True)
x=df["rainfall"].mean()
df["rainfall"].fillna(x,inplace=True)
print(df.to_string())
```

:

	N	P	K	temperature	humidity	ph	rainfall	label
0	50.261111	56.089523	75.563285	34.567890	57.402477	6.305336	90.869908	pomegranate
1	50.261111	25.445813	75.563285	26.411413	57.402477	6.305336	90.869908	pomegranate
2	50.261111	56.089523	6.497367	26.411413	57.402477	6.305336	90.869908	pomegranate
3	50.261111	56.089523	75.563285	26.411413	57.402477	6.305336	109.416919	pomegranate
4	2.000000	24.000000	38.000000	24.559816	91.635362	5.922936	111.968462	pomegranate
...	...	...	...	...	...	...	...	...
2703	42.000000	59.000000	55.000000	40.102077	94.351102	6.979102	149.119999	papaya
2704	43.000000	64.000000	47.000000	38.589545	91.580765	6.825665	102.270823	papaya
2705	35.000000	67.000000	49.000000	41.313301	91.150880	6.617067	239.742755	papaya
2706	56.000000	59.000000	55.000000	37.035519	91.794302	6.551893	188.518142	papaya
2707	39.000000	64.000000	53.000000	23.012402	91.073555	6.598860	208.335798	papaya

2708 rows × 8 columns

It indicates that the null values are replaced by mean value of the entire column of each parameter in the dataset.

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2708 entries, 0 to 2707
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   N            2708 non-null   float64
 1   P            2708 non-null   float64
 2   K            2708 non-null   float64
 3   temperature  2708 non-null   float64
 4   humidity     2708 non-null   float64
 5   ph           2708 non-null   float64
 6   rainfall     2708 non-null   float64
 7   label        2708 non-null   object 
dtypes: float64(7), object(1)
memory usage: 169.4+ KB
```

```
x=df["label"].nunique()
print("No. of crops:",x)
```

Output:

No.of crops: 9

It represents the no.of crops present in the dataset.

```
df=df.rename(columns={"label":"crop"})
df
```

:

	N	P	K	temperature	humidity	ph	rainfall	crop
0	50.261111	56.089523	75.563285	34.567890	57.402477	6.305336	90.869908	pomegranate
1	50.261111	25.445813	75.563285	26.411413	57.402477	6.305336	90.869908	pomegranate
2	50.261111	56.089523	6.497367	26.411413	57.402477	6.305336	90.869908	pomegranate
3	50.261111	56.089523	75.563285	26.411413	57.402477	6.305336	109.416919	pomegranate
4	2.000000	24.000000	38.000000	24.559816	91.635362	5.922936	111.968462	pomegranate
...	...	...	...	...	...	...	...	...
2703	42.000000	59.000000	55.000000	40.102077	94.351102	6.979102	149.119999	papaya
2704	43.000000	64.000000	47.000000	38.589545	91.580765	6.825665	102.270823	papaya
2705	35.000000	67.000000	49.000000	41.313301	91.150880	6.617067	239.742755	papaya
2706	56.000000	59.000000	55.000000	37.035519	91.794302	6.551893	188.518142	papaya
2707	39.000000	64.000000	53.000000	23.012402	91.073555	6.598860	208.335798	papaya

2708 rows × 8 columns

From the above, the column name changes from label to crop

```
print("Crop names:")
for crop in df["label"].unique():
    print(crop)
```

```
Crop names:
pomegranate
banana
mango
grapes
watermelon
muskmelon
apple
orange
papaya
```

We can print the particular crop data.

```
crop_name="papaya"
filtered_df=df[df["crop"]==crop_name]
filtered_df
```

]:

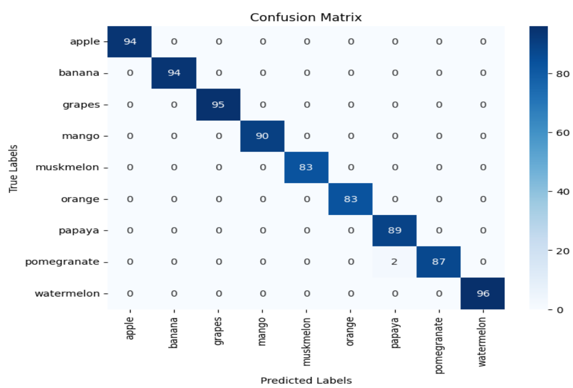
	N	P	K	temperature	humidity	ph	rainfall	crop
804	61.0	68.0	50.0	35.214628	91.497251	6.793245	243.074507	papaya
805	58.0	46.0	45.0	42.394134	90.790281	6.576261	88.466075	papaya
806	45.0	47.0	55.0	38.419163	91.142204	6.751453	119.265388	papaya
807	39.0	65.0	53.0	35.332949	92.115086	6.560743	235.613359	papaya
808	31.0	68.0	45.0	42.923253	90.076005	6.938313	196.240824	papaya
...	...	...	...	...	...	...	...	...
2703	42.0	59.0	55.0	40.102077	94.351102	6.979102	149.119999	papaya
2704	43.0	64.0	47.0	38.589545	91.580765	6.825665	102.270823	papaya
2705	35.0	67.0	49.0	41.313301	91.150880	6.617067	239.742755	papaya
2706	56.0	59.0	55.0	37.035519	91.794302	6.551893	188.518142	papaya
2707	39.0	64.0	53.0	23.012402	91.073555	6.598860	208.335798	papaya

Evolution measures

Using Support Vector Machine(SVM) Algorithm:

```
X = data.drop('label', axis=1)
y = data['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)
svm = SVC()
svm.fit(X_train, y_train)
```

```
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d", xticklabels=
svm.classes_ ,yticklabels=svm.classes_)
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()
```




```
y_pred = svm.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

Output:

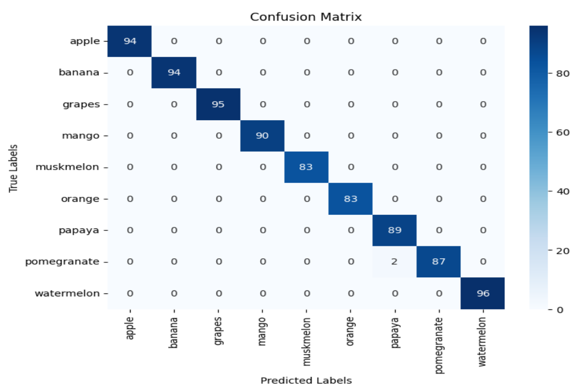
Accuracy: 0.997539975399754

Using K-Nearest Neighbour(KNN) Algorithm:

```
X = data.drop('label', axis=1)
y = data['label']
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.3,
random_state=42)
k = 19
knn = KNeighborsClassifier(n_neighbors=k)
knn.fit(X_train, y_train)
```

```
cm = confusion_matrix(y_test, y_pred)
```

```python
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, cmap="Blues", fmt="d", xticklabels=
knn.classes_, yticklabels=knn.classes_)
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()
```




```python
y_pred = knn.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

Output:

Accuracy: 0.9975399753997540

Predictions by the model

Using KNN Algorithm:

```python
new_data = {
    'N': 50,
    'P': 30,
    'K': 20,
    'temperature': 28,
    'humidity': 70,
    'ph': 6.5,
```

```
    'rainfall': 100
}
new_input = pd.DataFrame(new_data, index=[0])
predicted_label = knn.predict(new_input)
print("Predicted label:", predicted_label[0])
```

Output:

Predicted label: mango

Using SVM Algorithm:

```
new_data = {
    'N': 50,
    'P': 30,
    'K': 20,
    'temperature': 28,
    'humidity': 70,
    'ph': 6.5,
    'rainfall': 100
}
new_input = pd.DataFrame(new_data, index=[0])
predicted_label = svm.predict(new_input)
print("Predicted label:", predicted_label[0])
```

Output:

Predicted label: mango

Chapter 4

Conclusion and Future Work

4.0.1 Conclusion

Predictions by the model

Using KNN Algorithm:

```
new_data = {
    'N': 75,
    'P': 69,
    'K': 47,
    'temperature': 25,
    'humidity': 80,
    'ph': 6.3,
    'rainfall': 105
}
new_input = pd.DataFrame(new_data, index=[0])
```

```
predicted_label = svm.predict(new_input)
print("Predicted label:", predicted_label[0])
```

Output:

Predicted label: banana

Using SVM Algorithm:

```
predicted_label = knn.predict(new_input)
print("Predicted label:", predicted_label[0])
```

Output:

Predicted label: banana

The new data includes nitrogen(N)=75,phosphorus(P)=69,potassium(K)=47, temperature=25, humidity=80,pH=6.3 and rainfall=105,and the system predicts the label to be "banana".This prediction suggests that these parameter values check with the characteristics given the growth and cultivation of bananas.

In crop recommendation project,we built a system that can predict the type of crop based on certain parameters such as amount of nitrogen,phosphorus,potassium,temperature, humidity,pH,and rainfall.We used two different machine learning algorithms namely like k-Nearest Neighbors(KNN) and Support Vector Machines(SVM), to make these predictions.Our system performed well,by achieving an accuracy of 99.75%. This means the syatem correctly predict the type of crop in almost all cases. To check the performance of our models,we also created confusion matrix.

In conclusion,our project has successfully developed a crop prediction system using machine learning algorithms.By considering the given parameters,our system predicts the type of crop accurately.It helps farmers in making better decisions about crop selection.

4.0.2 Future Work

A crop recommedation system is a valuable application that can help farmers in making better decisions about which crops to grow based on various factors such as soil quality and climate conditions. It can suggest suitable crop for a specific region or field based on the parameters it helps farmers select crops that have higher chances of success and profitability. By recommending crops that are well suited to the conditions farmers can maximize crop productivity. It leads to better yeilds.It helps farmers optimize the use of fetilizers,water etc.,and reducing wastage.