In [3]:

```
#Importing Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
```

In [4]:

```
#Importing and reading Dataset
df=pd.read_csv("Crop_recommendation.csv")
```

In [5]:

```
#Reading data from Dataset
print("Top rows from Data are:\n",df.head(5),"\n")
print("Last rows from Data are:\n",df.tail(5),"\n")
```

```
Top rows from Data are:
     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 

Last rows from Data are:
         N   P   K  temperature   humidity        ph    rainfall   label
2195  107  34  32    26.774637  66.413269  6.780064  177.774507  coffee
2196   99  15  27    27.417112  56.636362  6.086922  127.924610  coffee
2197  118  33  30    24.131797  67.225123  6.362608  173.322839  coffee
2198  117  32  34    26.272418  52.127394  6.758793  127.175293  coffee
2199  104  18  30    23.603016  60.396475  6.779833  140.937041  coffee 
```

In [6]:

```
#checking for null values
print("No. of null values in Data are:\n",df.isnull().sum(),"\n")
```

```
No. of null values in Data are:
 N              0
P              0
K              0
temperature    0
humidity       0
ph             0
rainfall       0
label          0
dtype: int64 
```

In [7]:

```
#Column Separation
x=df.drop("label", axis=1)
y=df["label"]
print("Data without Target set is:\n",x,"\n")
print("Target Set is:\n",y,"\n")
```

```
Data without Target set is:
         N   P   K  temperature   humidity        ph    rainfall
0      90  42  43    20.879744  82.002744  6.502985  202.935536
```

```
1      85  58  41    21.770462  80.319644  7.038096  226.655537
2      60  55  44    23.004459  82.320763  7.840207  263.964248
3      74  35  40    26.491096  80.158363  6.980401  242.864034
4      78  42  42    20.130175  81.604873  7.628473  262.717340
...   ...  ..  ..          ...        ...       ...         ...
2195  107  34  32    26.774637  66.413269  6.780064  177.774507
2196   99  15  27    27.417112  56.636362  6.086922  127.924610
2197  118  33  30    24.131797  67.225123  6.362608  173.322839
2198  117  32  34    26.272418  52.127394  6.758793  127.175293
2199  104  18  30    23.603016  60.396475  6.779833  140.937041

[2200 rows x 7 columns]

Target Set is:
 0         rice
1         rice
2         rice
3         rice
4         rice
         ...
2195    coffee
2196    coffee
2197    coffee
2198    coffee
2199    coffee
Name: label, Length: 2200, dtype: object
```

In [8]:

```
#creating list for comparsion of model accuracy and corresponding model name
acc=[]
model=[]
```

In [9]:

```
#Divide Data into Train and Test
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y,test_size=0.2, random_state=15

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

In [10]:

```
#Logistic Regression
#Train the Model
from sklearn.linear_model import LogisticRegression
LogReg=LogisticRegression()
result1=LogReg.fit(x_train,y_train)

#Test the Model
predictions=result1.predict(x_test)

x=accuracy_score(y_test, predictions)
acc.append(x)
model.append("Logistic Regression")
print("Logistic Regression's Accuracy is ",x)
print("Classification Report is \n",classification_report(y_test, predictions))
```

```
Logistic Regression's Accuracy is  0.975
Classification Report is
```

```
               precision    recall  f1-score   support

       apple       1.00      1.00      1.00        15
      banana       1.00      1.00      1.00        21
   blackgram       0.91      1.00      0.95        20
    chickpea       1.00      1.00      1.00        27
     coconut       0.94      1.00      0.97        17
      coffee       1.00      1.00      1.00        24
      cotton       1.00      1.00      1.00        22
      grapes       1.00      1.00      1.00        20
        jute       0.84      0.89      0.86        18
 kidneybeans       0.95      1.00      0.98        20
      lentil       0.94      1.00      0.97        16
       maize       1.00      1.00      1.00        15
       mango       1.00      1.00      1.00        17
   mothbeans       1.00      0.86      0.93        29
    mungbean       1.00      1.00      1.00        23
   muskmelon       1.00      1.00      1.00        21
      orange       1.00      0.93      0.97        15
      papaya       0.91      1.00      0.95        20
  pigeonpeas       0.96      0.96      0.96        23
 pomegranate       1.00      1.00      1.00        21
        rice       1.00      0.86      0.92        21
  watermelon       1.00      1.00      1.00        15

    accuracy                           0.97       440
   macro avg       0.98      0.98      0.98       440
weighted avg       0.98      0.97      0.97       440
```

In [44]:

```python
#Decision Tree Classifier
#Train the Model
from sklearn.tree import DecisionTreeClassifier
DecTree=DecisionTreeClassifier(random_state=123)
result2=DecTree.fit(x_train,y_train)

#Test the Model
predictions=result2.predict(x_test)

x=accuracy_score(y_test, predictions)
acc.append(x)
model.append("Decision Tree")
print("Decision Tree Classifier's Accuracy is ",x)
print("Classification Report is \n",classification_report(y_test, predictions))
```

```
Decision Tree Classifier's Accuracy is  0.9931818181818182
Classification Report is 
               precision    recall  f1-score   support

       apple       1.00      1.00      1.00        15
      banana       1.00      1.00      1.00        21
   blackgram       1.00      1.00      1.00        20
    chickpea       1.00      1.00      1.00        27
     coconut       1.00      1.00      1.00        17
      coffee       0.96      1.00      0.98        24
      cotton       1.00      1.00      1.00        22
      grapes       1.00      1.00      1.00        20
        jute       0.89      0.94      0.92        18
 kidneybeans       1.00      1.00      1.00        20
      lentil       1.00      1.00      1.00        16
       maize       1.00      1.00      1.00        15
       mango       1.00      1.00      1.00        17
   mothbeans       1.00      1.00      1.00        29
```

```
    mungbean       1.00      1.00      1.00        23
   muskmelon       1.00      1.00      1.00        21
      orange       1.00      1.00      1.00        15
      papaya       1.00      1.00      1.00        20
  pigeonpeas       1.00      1.00      1.00        23
 pomegranate       1.00      1.00      1.00        21
        rice       1.00      0.90      0.95        21
  watermelon       1.00      1.00      1.00        15

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

In [88]:

```
#Random Tree Classifier
#Train the Model
from sklearn.ensemble import RandomForestClassifier
RanTree=RandomForestClassifier(n_estimators = 5, criterion = 'entropy', random_state =
result3=RanTree.fit(x_train,y_train)

#Test the Model
predictions=result3.predict(x_test)

x=accuracy_score(y_test, predictions)
acc.append(x)
model.append("Random Tree")
print("Random Tree Classifier's Accuracy is ",x)
print("Classification Report is \n",classification_report(y_test, predictions))
```

```
Random Tree Classifier's Accuracy is  0.990909090909091
Classification Report is
               precision    recall  f1-score   support

       apple       1.00      1.00      1.00        15
      banana       1.00      1.00      1.00        21
   blackgram       1.00      1.00      1.00        20
    chickpea       1.00      1.00      1.00        27
     coconut       1.00      1.00      1.00        17
      coffee       1.00      1.00      1.00        24
      cotton       1.00      1.00      1.00        22
      grapes       1.00      1.00      1.00        20
        jute       0.94      0.94      0.94        18
 kidneybeans       1.00      1.00      1.00        20
      lentil       0.94      1.00      0.97        16
       maize       1.00      1.00      1.00        15
       mango       1.00      1.00      1.00        17
   mothbeans       1.00      0.93      0.96        29
    mungbean       1.00      1.00      1.00        23
   muskmelon       1.00      1.00      1.00        21
      orange       1.00      1.00      1.00        15
      papaya       1.00      1.00      1.00        20
  pigeonpeas       0.96      1.00      0.98        23
 pomegranate       1.00      1.00      1.00        21
        rice       0.95      0.95      0.95        21
  watermelon       1.00      1.00      1.00        15

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

In [13]:

```
#Support Vector machine
```

```
#Train the Model
from sklearn.svm import SVC
SVM=SVC(kernel='rbf', random_state=0)
result4=SVM.fit(x_train,y_train)

#Test the Model
predictions=result4.predict(x_test)

x=accuracy_score(y_test, predictions)
acc.append(x)
model.append("SVM")
print("SVM's Accuracy is ",x)
print("Classification Report is \n",classification_report(y_test, predictions))
```

```
SVM's Accuracy is  0.990909090909091
Classification Report is 
               precision    recall  f1-score   support

       apple       1.00      1.00      1.00        15
      banana       1.00      1.00      1.00        21
   blackgram       1.00      1.00      1.00        20
    chickpea       1.00      1.00      1.00        27
     coconut       1.00      1.00      1.00        17
      coffee       1.00      1.00      1.00        24
      cotton       1.00      1.00      1.00        22
      grapes       1.00      1.00      1.00        20
        jute       0.86      1.00      0.92        18
 kidneybeans       0.95      1.00      0.98        20
      lentil       1.00      1.00      1.00        16
       maize       1.00      1.00      1.00        15
       mango       1.00      1.00      1.00        17
   mothbeans       1.00      1.00      1.00        29
    mungbean       1.00      1.00      1.00        23
   muskmelon       1.00      1.00      1.00        21
      orange       1.00      1.00      1.00        15
      papaya       1.00      1.00      1.00        20
  pigeonpeas       1.00      0.96      0.98        23
 pomegranate       1.00      1.00      1.00        21
        rice       1.00      0.86      0.92        21
  watermelon       1.00      1.00      1.00        15

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

In [22]:

```
#KNN
#Train the Model
from sklearn.neighbors import KNeighborsClassifier
KNN=KNeighborsClassifier(n_neighbors=50)
result5=KNN.fit(x_train,y_train)

#Test the Model
predictions=result5.predict(x_test)

x=accuracy_score(y_test, predictions)
acc.append(x)
model.append("KNN")
print("KNN's Accuracy is ",x)
print("Classification Report is \n",classification_report(y_test, predictions))
```

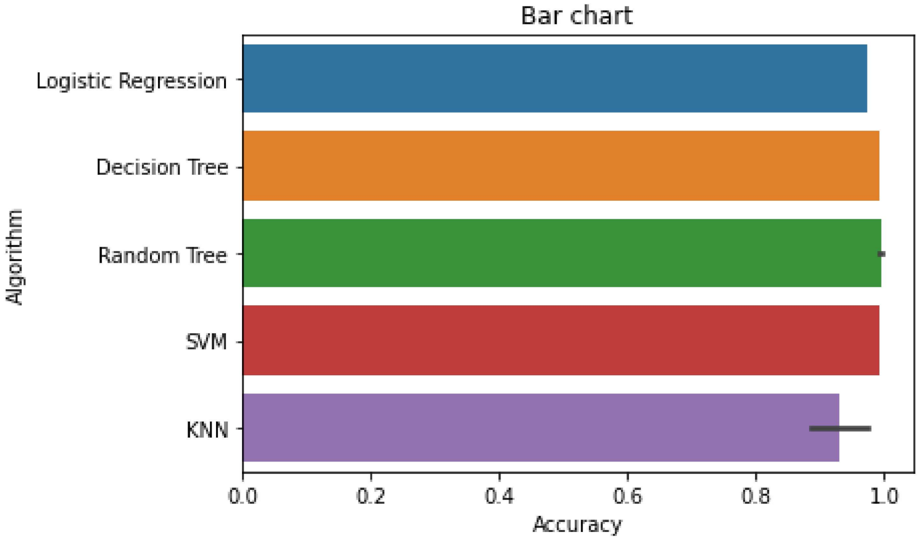
```
KNN's Accuracy is  0.8863636363636364
Classification Report is
               precision    recall  f1-score   support

       apple       0.68      1.00      0.81        15
      banana       1.00      1.00      1.00        21
   blackgram       0.80      1.00      0.89        20
    chickpea       1.00      1.00      1.00        27
     coconut       0.89      1.00      0.94        17
      coffee       1.00      1.00      1.00        24
      cotton       1.00      1.00      1.00        22
      grapes       1.00      0.65      0.79        20
        jute       0.75      1.00      0.86        18
 kidneybeans       0.74      1.00      0.85        20
      lentil       0.73      1.00      0.84        16
       maize       1.00      1.00      1.00        15
       mango       0.65      1.00      0.79        17
   mothbeans       1.00      0.55      0.71        29
    mungbean       0.88      1.00      0.94        23
   muskmelon       1.00      1.00      1.00        21
      orange       1.00      0.60      0.75        15
      papaya       1.00      0.75      0.86        20
  pigeonpeas       1.00      0.39      0.56        23
 pomegranate       0.84      1.00      0.91        21
        rice       0.94      0.76      0.84        21
  watermelon       1.00      1.00      1.00        15

    accuracy                           0.89       440
   macro avg       0.91      0.90      0.88       440
weighted avg       0.91      0.89      0.88       440
```

In [87]:

```
#Accuarcy comparision for different Algorithms
plt.title("Bar chart")
plt.xlabel("Accuracy")
plt.ylabel("Algorithm")
sns.barplot(x=acc, y=model)
plt.show()
```



In [86]:

```
#making a Predictions
new_pred=result3.predict([[0,5,36,24,90,6,105]])
print("Recommended Crop is ",new_pred)
```

```
Recommended Crop is  ['apple']
```

In [82]:

```
#making a Predictions
new_pred=result3.predict([[51,57,55,24,90,6,108]])
print("Recommended Crop is ",new_pred)
```

```
Recommended Crop is  ['papaya']
```

In [ ]: