```
**OPEN ENDED ASSIGNMENT**

BY

HARSHALI BORHADE

UEC2022016
```

OPEN ENDED ASSIGNMENT

BY

HARSHALI BORHADE

UEC2022016

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

Importing necessary libraries

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import sklearn as sl
```

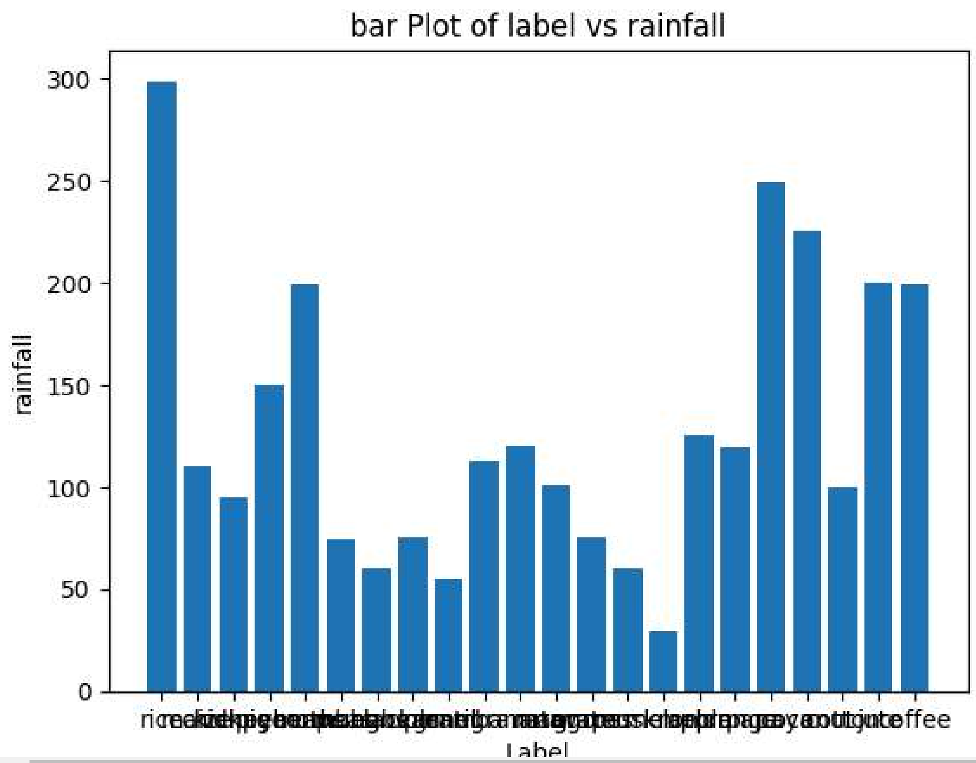
```
Start coding or generate with AI.
```

```
df=pd.read_csv('/content/Crop_recommendation.csv')
print(df)
```

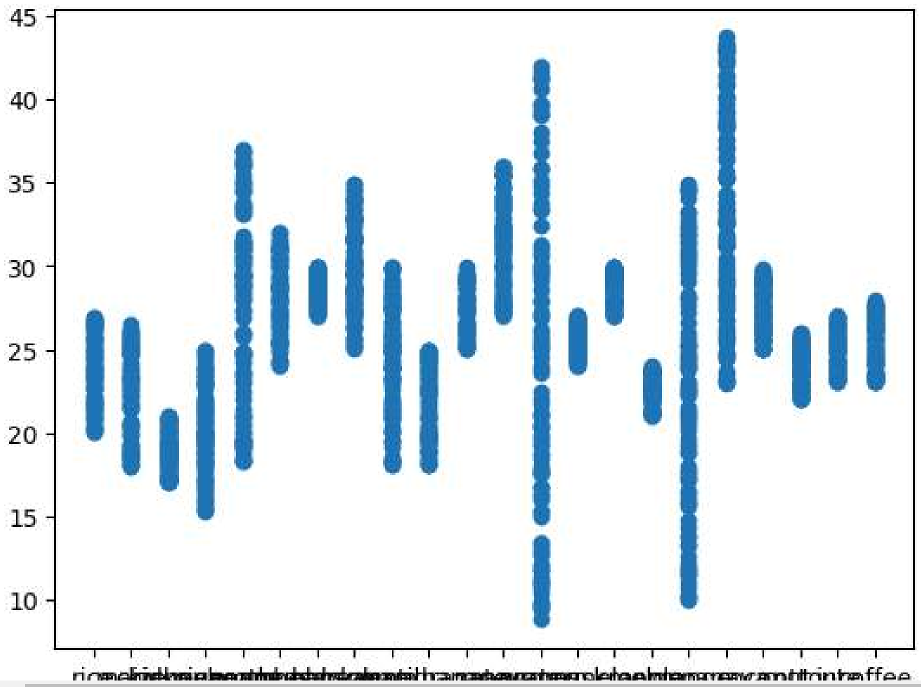
```
        N   P   K  temperature   humidity        ph    rainfall   label
0      90  42  43    20.879744  82.002744  6.502985  202.935536    rice
1      85  58  41    21.770462  80.319644  7.038096  226.655537    rice
2      60  55  44    23.004459  82.320763  7.840207  263.964248    rice
3      74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

[2200 rows x 8 columns]
```

```
plt.bar(df.label, df.rainfall)
plt.xlabel('Label')
plt.ylabel('rainfall')
plt.title('bar Plot of label vs rainfall')
plt.show()
```



```
import matplotlib.pyplot as plt
plt.scatter(df.label,df.temperature)
plt.show()
```



Split the data

```
from sklearn.model_selection import train_test_split
x=df.iloc[:,0:7]
y=df.iloc[:,7]
```

```
print(x)
print(y)
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3)
print(xtrain,xtest,ytrain,ytest)
```

```
3         rice
4         rice
         ...
2195    coffee
2196    coffee
2197    coffee
2198    coffee
2199    coffee
Name: label, Length: 2200, dtype: object
        N   P   K  temperature   humidity        ph    rainfall
1859   37  10  32    28.963183  95.163337  6.165085  222.803013
107    89  60  19    25.191924  66.690290  5.913665   78.066396
1972  111  39  22    22.603616  80.350905  6.135025   88.573955
728    32  66  17    34.946616  65.267740  7.162358   70.141514
1480   82  20  54    29.340336  90.015064  6.541150   21.445329
...   ...  ..  ..          ...        ...       ...         ...
355    22  71  17    18.153002  19.386021  5.509295  107.690796
526     8  60  18    31.216300  46.018682  3.808429   53.120528
635    14  48  21    29.245990  84.800841  6.991242   53.432289
2133   82  24  33    26.535432  67.096081  6.809594  120.649443
974    15   6  41    19.008707  88.837681  6.897368  108.679398

[1540 rows x 7 columns]         N    P    K  temperature   humidity        ph    rainfall
207    59   70   84    17.334868  18.749270  7.550808   82.617347
2074   73   43   42    26.583610  78.007748  6.310700  154.823886
1600   22   30   12    15.781442  92.510777  6.354007  119.035002
1259   17  136  195    41.207336  81.610510  6.389783   65.902275
1587    1  135  203    22.778565  92.701240  5.624203  113.775922
...   ...  ...  ...          ...        ...       ...         ...
682     6   47   18    29.161746  80.280381  6.715277   40.165460
1369  113   19   46    25.418640  81.121230  6.286388   49.523207
800    32   76   15    28.051536  63.498022  7.604110   43.357954
639    14   57   15    29.875702  83.147963  6.623438   40.120442
623    31   37   21    27.239250  86.404241  6.713411   37.312369

[660 rows x 7 columns] 1859        coconut
107           maize
1972         cotton
728       blackgram
1480      muskmelon
            ...
355     kidneybeans
526       mothbeans
635        mungbean
2133         coffee
974     pomegranate
Name: label, Length: 1540, dtype: object 207       chickpea
2074          jute
1600        orange
1259        grapes
1587         apple
            ...
682       mungbean
1369    watermelon
800         lentil
639       mungbean
623       mungbean
Name: label, Length: 660, dtype: object
```

Build Logistic Regression Model using training data

```
from sklearn.linear_model import LogisticRegression
mymodel=LogisticRegression()
mymodel.fit(xtrain,ytrain)
```

```
;r/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfg:
)P: TOTAL NO. of ITERATIONS REACHED LIMIT.

:rease the number of iterations (max_iter) or scale the data as shown in:
  https://scikit-learn.org/stable/modules/preprocessing.html
ease also refer to the documentation for alternative solver options:
  https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
ı_iter_i = _check_optimize_result(
.ogisticRegression
gisticRegression()
```

Testing the model

```
ypred=mymodel.predict(xtest)
print(ypred)
y1=mymodel.predict_proba(xtest)
print(y1)
```

```
['chickpea' 'jute' 'orange' 'grapes' 'apple' 'coffee' 'chickpea' 'maize'
 'coffee' 'cotton' 'mango' 'banana' 'blackgram' 'coffee' 'grapes'
 'coconut' 'jute' 'kidneybeans' 'lentil' 'orange' 'lentil' 'apple' 'mango'
 'pomegranate' 'mothbeans' 'grapes' 'grapes' 'lentil' 'papaya'
 'pigeonpeas' 'orange' 'papaya' 'kidneybeans' 'blackgram' 'watermelon'
 'grapes' 'apple' 'coconut' 'grapes' 'jute' 'kidneybeans' 'chickpea'
 'pomegranate' 'banana' 'mothbeans' 'grapes' 'lentil' 'chickpea' 'coconut'
 'banana' 'rice' 'watermelon' 'jute' 'coconut' 'pomegranate' 'coffee'
 'mungbean' 'coffee' 'kidneybeans' 'pigeonpeas' 'mungbean' 'papaya'
 'lentil' 'blackgram' 'pomegranate' 'mango' 'blackgram' 'blackgram'
 'kidneybeans' 'cotton' 'mango' 'mungbean' 'pigeonpeas' 'kidneybeans'
 'maize' 'orange' 'pomegranate' 'banana' 'grapes' 'watermelon' 'grapes'
 'coconut' 'coffee' 'papaya' 'pomegranate' 'maize' 'pigeonpeas' 'orange'
 'mango' 'grapes' 'papaya' 'mango' 'kidneybeans' 'papaya' 'cotton'
 'banana' 'coffee' 'papaya' 'cotton' 'banana' 'mango' 'coconut' 'mungbean'
 'jute' 'pomegranate' 'mungbean' 'jute' 'papaya' 'cotton' 'banana' 'apple'
 'papaya' 'mothbeans' 'pomegranate' 'jute' 'blackgram' 'rice' 'grapes'
 'watermelon' 'apple' 'papaya' 'jute' 'banana' 'mungbean' 'banana' 'rice'
 'mango' 'coffee' 'coconut' 'watermelon' 'watermelon' 'mungbean' 'lentil'
 'lentil' 'muskmelon' 'mungbean' 'kidneybeans' 'watermelon' 'pomegranate'
 'grapes' 'mango' 'kidneybeans' 'muskmelon' 'blackgram' 'jute'
 'pomegranate' 'watermelon' 'coffee' 'cotton' 'papaya' 'orange' 'chickpea'
 'mungbean' 'papaya' 'papaya' 'cotton' 'maize' 'pigeonpeas' 'apple'
 'coffee' 'kidneybeans' 'cotton' 'mothbeans' 'blackgram' 'orange'
 'mungbean' 'mango' 'apple' 'mango' 'mango' 'mungbean' 'papaya' 'coconut'
 'orange' 'coffee' 'muskmelon' 'chickpea' 'lentil' 'apple' 'jute' 'papaya'
 'orange' 'blackgram' 'papaya' 'grapes' 'pomegranate' 'pigeonpeas'
 'muskmelon' 'rice' 'mango' 'lentil' 'cotton' 'pomegranate' 'blackgram'
 'maize' 'mungbean' 'mango' 'apple' 'mango' 'orange' 'rice' 'muskmelon'
 'banana' 'pomegranate' 'papaya' 'coconut' 'coconut' 'blackgram' 'jute'
 'pomegranate' 'apple' 'maize' 'cotton' 'papaya' 'kidneybeans'
 'pigeonpeas' 'grapes' 'papaya' 'orange' 'maize' 'mothbeans' 'mothbeans'
 'banana' 'rice' 'apple' 'grapes' 'watermelon' 'jute' 'maize'
 'pomegranate' 'watermelon' 'coconut' 'maize' 'pomegranate' 'mango'
 'cotton' 'pigeonpeas' 'papaya' 'coconut' 'pomegranate' 'maize' 'jute'
 'rice' 'mothbeans' 'pomegranate' 'orange' 'watermelon' 'watermelon'
 'rice' 'coffee' 'watermelon' 'maize' 'apple' 'jute' 'cotton' 'cotton'
 'coffee' 'mothbeans' 'banana' 'apple' 'rice' 'papaya' 'pigeonpeas'
 'muskmelon' 'coconut' 'rice' 'mungbean' 'pomegranate' 'mungbean'
 'pigeonpeas' 'mothbeans' 'mothbeans' 'rice' 'mungbean' 'watermelon'
 'kidneybeans' 'blackgram' 'jute' 'jute' 'pomegranate' 'kidneybeans'
 'muskmelon' 'muskmelon' 'mungbean' 'chickpea' 'banana' 'jute' 'maize'
 'cotton' 'lentil' 'watermelon' 'coffee' 'jute' 'lentil' 'watermelon'
 'banana' 'banana' 'mothbeans' 'pomegranate' 'lentil' 'maize' 'banana'
 'watermelon' 'pigeonpeas' 'pomegranate' 'watermelon' 'blackgram'
 'mungbean' 'chickpea' 'papaya' 'jute' 'pigeonpeas' 'watermelon'
 'muskmelon' 'watermelon' 'banana' 'lentil' 'blackgram' 'muskmelon'
 'muskmelon' 'cotton' 'grapes' 'coffee' 'cotton' 'mothbeans' 'chickpea'
 'chickpea' 'banana' 'papaya' 'pomegranate' 'pomegranate' 'mango' 'grapes'
 'watermelon' 'kidneybeans' 'lentil' 'watermelon' 'watermelon' 'muskmelon'
 'papaya' 'apple' 'maize' 'pigeonpeas' 'mango' 'mungbean' 'blackgram'
 'mothbeans' 'orange' 'mango' 'pomegranate' 'grapes' 'cotton' 'mango'
 'jute' 'muskmelon' 'coffee' 'maize' 'papaya' 'muskmelon' 'coconut'
 'mungbean' 'grapes' 'mothbeans' 'kidneybeans' 'mothbeans' 'mango'
 'lentil' 'maize' 'coffee' 'orange' 'pomegranate' 'mango' 'lentil'
 'papaya' 'grapes' 'mango' 'apple' 'papaya' 'jute' 'orange' 'chickpea'
 'lentil' 'orange' 'blackgram' 'watermelon' 'grapes' 'mothbeans'
 'watermelon' 'apple' 'lentil' 'maize' 'grapes' 'chickpea' 'rice' 'jute'
```

```
from sklearn.metrics import confusion_matrix
confmat=confusion_matrix(ytest,ypred)
print(confmat)
```

```
[[26  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0 33  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0 23  0  0  0  0  0  0  0  0  2  0  3  0  0  0  0  0  0  0  0]
 [ 0  0  0 23  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0 26  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0 35  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0 31  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0 33  0  0  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0 31  0  0  0  0  0  0  0  0  0  0  0  1  0]
 [ 0  0  0  0  0  0  0  0  0 20  0  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0 32  0  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  1  1  0  0  0  0 28  0  0  0  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 33  0  0  0  0  0  0  0  0  0]
 [ 0  0  2  0  0  0  0  0  0  0  0  0  0 28  1  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  1  0  0  0 31  0  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 25  0  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 24  0  0  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 36  0  0  0  0]
 [ 0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 30  0  0  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 34  0  0]
 [ 0  0  0  0  0  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0 21  0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 38]]
```

```
from sklearn.metrics import classification_report
print(classification_report(ytest,ypred))
```

```
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        26
      banana       1.00      1.00      1.00        33
   blackgram       0.88      0.82      0.85        28
    chickpea       1.00      1.00      1.00        23
     coconut       1.00      1.00      1.00        26
      coffee       0.97      1.00      0.99        35
      cotton       0.97      0.97      0.97        32
      grapes       1.00      1.00      1.00        33
        jute       0.86      0.97      0.91        32
 kidneybeans       1.00      1.00      1.00        20
      lentil       0.97      1.00      0.98        32
       maize       0.90      0.93      0.92        30
       mango       1.00      1.00      1.00        33
   mothbeans       0.90      0.90      0.90        31
    mungbean       0.97      0.97      0.97        32
   muskmelon       1.00      1.00      1.00        25
      orange       1.00      1.00      1.00        24
      papaya       1.00      1.00      1.00        36
  pigeonpeas       1.00      0.97      0.98        31
 pomegranate       1.00      1.00      1.00        34
        rice       0.95      0.81      0.88        26
  watermelon       1.00      1.00      1.00        38

    accuracy                           0.97       660
   macro avg       0.97      0.97      0.97       660
weighted avg       0.97      0.97      0.97       660
```