In [ ]:

```python
import pandas as pd
import numpy as np #working with multi dimensional arrays
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report# used for classification and r
from sklearn import metrics
```

In [ ]:

```python
df = pd.read_csv('../Data-processed/crop_recommendation.csv')
```

In [ ]:

```python
df.head()
```

Out[ ]:

	N	P	K	temperature	humidity	ph	rainfall	label
0	77	49	35	20.879744	82.002744	6.502985	202.935536	rice
1	60	56	41	21.770462	80.319644	7.038096	226.655537	rice
2	62	51	35	23.004459	82.320763	7.840207	263.964248	rice
3	88	47	38	26.491096	80.158363	6.980401	242.864034	rice
4	65	45	38	20.130175	81.604873	7.628473	262.717340	rice

In [ ]:

```python
df.tail()
```

Out[ ]:

	N	P	K	temperature	humidity	ph	rainfall	label
2195	95	18	35	26.774637	66.413269	6.780064	177.774507	coffee
2196	80	40	34	27.417112	56.636362	6.086922	127.924610	coffee
2197	88	22	31	24.131797	67.225123	6.362608	173.322839	coffee
2198	93	20	28	26.272418	52.127394	6.758793	127.175293	coffee
2199	100	32	29	23.603016	60.396475	6.779833	140.937041	coffee

In [ ]:

```python
df.shape
```

Out[ ]:

```
(2200, 8)
```

In [ ]:

```python
df.columns
```

Out[ ]:

```
Index(['N', 'P', 'K', 'temperature', 'humidity', 'ph', 'rainfall', 'label'], dt
ype='object')
```

In [ ]:

```python
df['label'].unique()
```

Out[ ]:

```
array(['rice', 'maize', 'chickpea', 'kidneybeans', 'pigeonpeas',
       'mothbeans', 'mungbean', 'blackgram', 'lentil', 'pomegranate',
       'banana', 'mango', 'grapes', 'watermelon', 'muskmelon', 'apple',
       'orange', 'papaya', 'coconut', 'cotton', 'jute', 'coffee'],
      dtype=object)
```

In [ ]:

```python
df.dtypes
```

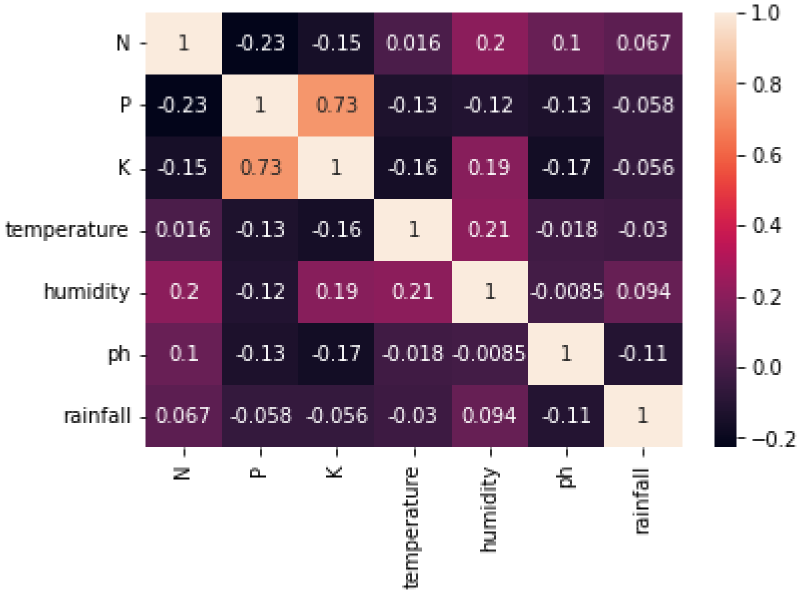
```
Out[ ]: N                int64
        P                int64
        K                int64
        temperature    float64
        humidity       float64
        ph             float64
        rainfall       float64
        label           object
        dtype: object
```

```
In [ ]: df['label'].value_counts()
```

```
Out[ ]: 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
```

```
In [ ]: sns.heatmap(df.corr(),annot=True)
```

```
Out[ ]: <AxesSubplot:>
```



Separating features and target label

In [ ]:

```
features = df[['N', 'P','K','temperature', 'humidity', 'ph', 'rainfall']]
target = df['label']
labels = df['label']
```

In [ ]:

```
acc = []
model = []
```

In [ ]:

```
from sklearn.model_selection import train_test_split
Xtrain, Xtest, Ytrain, Ytest = train_test_split(features,target,test_size = 0.2,
```

Random Forest

In [ ]:

```
from sklearn.ensemble import RandomForestClassifier

RF = RandomForestClassifier(n_estimators=20, random_state=0)
RF.fit(Xtrain,Ytrain)

predicted_values = RF.predict(Xtest)

x = metrics.accuracy_score(Ytest, predicted_values)
acc.append(x)
model.append('RF')
print("RF's Accuracy is: ", x)

print(classification_report(Ytest,predicted_values))
```

```
RF's Accuracy is:  0.9931818181818182
              precision    recall  f1-score   support

       apple       1.00      1.00      1.00        13
      banana       1.00      1.00      1.00        17
   blackgram       1.00      1.00      1.00        16
    chickpea       1.00      1.00      1.00        21
     coconut       1.00      1.00      1.00        21
      coffee       1.00      1.00      1.00        22
      cotton       1.00      1.00      1.00        20
      grapes       1.00      1.00      1.00        18
        jute       0.90      1.00      0.95        28
 kidneybeans       1.00      1.00      1.00        14
      lentil       1.00      1.00      1.00        23
       maize       1.00      1.00      1.00        21
       mango       1.00      1.00      1.00        26
   mothbeans       1.00      1.00      1.00        19
    mungbean       1.00      1.00      1.00        24
   muskmelon       1.00      1.00      1.00        23
      orange       1.00      1.00      1.00        29
      papaya       1.00      1.00      1.00        19
  pigeonpeas       1.00      1.00      1.00        18
 pomegranate       1.00      1.00      1.00        17
        rice       1.00      0.81      0.90        16
  watermelon       1.00      1.00      1.00        15

    accuracy                           0.99       440
   macro avg       1.00      0.99      0.99       440
weighted avg       0.99      0.99      0.99       440
```

```
In [ ]: score = cross_val_score(RF,features,target,cv=5)
        score
```

```
Out[ ]: array([0.99545455, 0.99545455, 0.99545455, 0.99090909, 0.98409091])
```

Saving trained Random Forest model

```
In [ ]: import pickle
        RF_pkl_filename = '../models/RandomForest.pkl'
        RF_Model_pkl = open(RF_pkl_filename, 'wb')
        pickle.dump(RF, RF_Model_pkl)
        RF_Model_pkl.close()
```

Making a prediction

```
In [ ]: data = np.array([[104,18, 30, 23.603016, 60.3, 6.7, 140.91]])
        prediction = RF.predict(data)
        print(prediction)
```

```
['coffee']
c:\Users\hp\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\bas
e.py:450: UserWarning: X does not have valid feature names, but RandomForestClass
ifier was fitted with feature names
  warnings.warn(
```

```
In [ ]: data = np.array([[83, 45, 60, 28, 70.3, 7.0, 150.9]])
        prediction = RF.predict(data)
        print(prediction)
```

```
['coffee']
c:\Users\hp\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\bas
e.py:450: UserWarning: X does not have valid feature names, but RandomForestClass
ifier was fitted with feature names
  warnings.warn(
```