project-crop

January 5, 2024

[1]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
gnb=GaussianNB()
dtc=DecisionTreeClassifier(criterion='entropy',max_depth=3)
rfc=RandomForestClassifier(n_estimators=700, criterion='entropy')
import warnings
```

[2]:
```
df = pd.read_csv("C:/Users/ASUS/OneDrive/Desktop/Crop_recommendation.csv")
```

[3]:
```
df.head()
```

[3]:

	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

[4]:
```
df.dtypes
```

[4]:
```
N                int64
P                int64
K                int64
temperature    float64
humidity       float64
ph             float64
rainfall       float64
label           object
dtype: object
```

[5]:
```
x = df.drop(columns=["label"])
y = df["label"]
```

[6]:
```
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2,␣
↪random_state=42)
```

[7]:
```
gnb.fit(xtrain,ytrain)
```

[7]:
```
GaussianNB()
```

[8]:
```
ypred_gnb=gnb.predict(xtest)
```

[9]:
```
print(f"The testing set of Naive Baye's model is {gnb.score(xtest,ytest)}")
print(f"The testing set of Naive Baye's is {gnb.score(xtrain,ytrain)}")
```

```
The testing set of Naive Baye's model is 0.9954545454545455
The testing set of Naive Baye's is 0.9948863636363636
```

[10]:
```
from sklearn.metrics import accuracy_score
```

[11]:
```
print(f"The accuracy score id {accuracy_score(ytest,ypred_gnb)}")
```

```
The accuracy score id 0.9954545454545455
```

[12]:
```
class_report = classification_report(ytest, ypred_gnb)
print("Classification Report:\n", class_report)
```

```
Classification Report:
               precision    recall  f1-score   support

       apple       1.00      1.00      1.00        23
      banana       1.00      1.00      1.00        21
   blackgram       1.00      1.00      1.00        20
    chickpea       1.00      1.00      1.00        26
     coconut       1.00      1.00      1.00        27
      coffee       1.00      1.00      1.00        17
      cotton       1.00      1.00      1.00        17
      grapes       1.00      1.00      1.00        14
        jute       0.92      1.00      0.96        23
 kidneybeans       1.00      1.00      1.00        20
      lentil       1.00      1.00      1.00        11
       maize       1.00      1.00      1.00        21
       mango       1.00      1.00      1.00        19
   mothbeans       1.00      1.00      1.00        24
    mungbean       1.00      1.00      1.00        19
   muskmelon       1.00      1.00      1.00        17
      orange       1.00      1.00      1.00        14
      papaya       1.00      1.00      1.00        23
  pigeonpeas       1.00      1.00      1.00        23
 pomegranate       1.00      1.00      1.00        23
        rice       1.00      0.89      0.94        19
```

```
  watermelon       1.00      1.00      1.00        19

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

[13]:
```
dtc.fit(xtrain,ytrain)
```

[13]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=3)
```

[14]:
```
ypred_dtc=dtc.predict(xtest)
```

[15]:
```
print(f"The testing set of Decision Tree model is {dtc.score(xtest,ytest)}")
print(f"The testing set of Decision Tree is {dtc.score(xtrain,ytrain)}")
```

```
The testing set of Decision Tree model is 0.3
The testing set of Decision Tree is 0.37670454545454546
```

[16]:
```
print(f"The accuracy score is {accuracy_score(ytest,ypred_dtc)}")
```

```
The accuracy score is 0.3
```

[25]:
```
class_report = classification_report(ytest, ypred_dtc)
print("Classification Report:\n", class_report)
warnings.filterwarnings('ignore')
```

```
Classification Report:
              precision    recall  f1-score   support

       apple       0.00      0.00      0.00        23
      banana       0.34      1.00      0.51        21
   blackgram       0.00      0.00      0.00        20
    chickpea       0.00      0.00      0.00        26
     coconut       0.00      0.00      0.00        27
      coffee       0.45      1.00      0.62        17
      cotton       0.47      1.00      0.64        17
      grapes       0.38      1.00      0.55        14
        jute       0.00      0.00      0.00        23
 kidneybeans       0.00      0.00      0.00        20
      lentil       0.09      1.00      0.17        11
       maize       0.00      0.00      0.00        21
       mango       0.00      0.00      0.00        19
   mothbeans       0.00      0.00      0.00        24
    mungbean       0.00      0.00      0.00        19
   muskmelon       0.47      1.00      0.64        17
      orange       0.29      1.00      0.44        14
      papaya       0.00      0.00      0.00        23
```

```
  pigeonpeas       0.00      0.00      0.00        23
 pomegranate       0.32      0.91      0.47        23
        rice       0.00      0.00      0.00        19
  watermelon       0.00      0.00      0.00        19

    accuracy                           0.30       440
   macro avg       0.13      0.36      0.18       440
weighted avg       0.11      0.30      0.16       440
```

[18]:
```
rfc.fit(xtrain,ytrain)
```

[18]:
```
RandomForestClassifier(criterion='entropy', n_estimators=700)
```

[19]:
```
ypred_rfc=rfc.predict(xtest)
```

[20]:
```
print(f"The testing set of Random forest model is {rfc.score(xtest,ytest)}")
print(f"The testing set of Random forest is {rfc.score(xtrain,ytrain)}")
```

```
The testing set of Random forest model is 0.9931818181818182
The testing set of Random forest is 1.0
```

[21]:
```
print(f"The accuracy score id {accuracy_score(ytest,ypred_rfc)}")
```

```
The accuracy score id 0.9931818181818182
```

[22]:
```
class_report = classification_report(ytest, ypred_rfc)
print("Classification Report:\n", class_report)
```

```
Classification Report:
               precision    recall  f1-score   support

       apple       1.00      1.00      1.00        23
      banana       1.00      1.00      1.00        21
   blackgram       1.00      1.00      1.00        20
    chickpea       1.00      1.00      1.00        26
     coconut       1.00      1.00      1.00        27
      coffee       1.00      1.00      1.00        17
      cotton       1.00      1.00      1.00        17
      grapes       1.00      1.00      1.00        14
        jute       0.92      1.00      0.96        23
 kidneybeans       1.00      1.00      1.00        20
      lentil       0.92      1.00      0.96        11
       maize       1.00      1.00      1.00        21
       mango       1.00      1.00      1.00        19
   mothbeans       1.00      0.96      0.98        24
    mungbean       1.00      1.00      1.00        19
   muskmelon       1.00      1.00      1.00        17
```

```
      orange       1.00      1.00      1.00        14
      papaya       1.00      1.00      1.00        23
  pigeonpeas       1.00      1.00      1.00        23
 pomegranate       1.00      1.00      1.00        23
        rice       1.00      0.89      0.94        19
  watermelon       1.00      1.00      1.00        19

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

0.0.1 Using Naive Baye's model since the accuracy is greater than other classifiers.

[23]:
```
#Testing using random data
data = np.array([[104,18, 30, 23.603016, 60.3, 6.7, 140.91]])
prediction = gnb.predict(data)
print(prediction)
```

```
['coffee']
```

[24]:
```
data = np.array([[83, 45, 60, 28, 70.3, 7.0, 150.9]])
prediction = gnb.predict(data)
print(prediction)
```

```
['jute']
```