random_forest_classification

January 27, 2022

[ ]:
```
import seaborn as sns
import pandas as pd
import numpy as np
from sklearn.tree import plot_tree
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

df= sns.load_dataset("iris")
X=df.iloc[:,:-1]
y=df.iloc[:,-1:]
```

[ ]:
```
model = RandomForestClassifier(n_estimators=100)
model.fit(X,y)
model.predict([[2,3,4,5]])
```

```
C:\Users\dell7450\AppData\Local\Temp/ipykernel_11344/1312009047.py:2:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples,), for example using
ravel().
  model.fit(X,y)
C:\Users\dell7450\AppData\Local\Programs\Python\Python310\lib\site-
packages\sklearn\base.py:450: UserWarning: X does not have valid feature names,
but RandomForestClassifier was fitted with feature names
  warnings.warn(
```

[ ]:
```
array(['virginica'], dtype=object)
```

[ ]:
```
X_train,X_test,y_train,y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=0)
#model.fit(X,y)
```

0.0.1 Accuracy (X-test,y_test)

[ ]:
```
score = model.score(X_test,y_test)
print("The accuracy score with 80-20 (X_test) and (y_test) is ",score)
```

```
The accuracy score with 80-20 (X_test) and (y_test) is  1.0
```

0.0.2 Metric Accuracy (y_test, predictions)

[ ]:
```
predicted_values = model.predict(X_test)
print("The predicted values from 20% of test input is",predicted_values,"\n")

superscore = accuracy_score(y_test,predicted_values)
print("The accuracy score of model when compared with twenty percent original␣
 ↪test values is",superscore)
```

```
The predicted values from 20% of test input is ['virginica' 'versicolor'
'setosa' 'virginica' 'setosa' 'virginica'
 'setosa' 'versicolor' 'versicolor' 'versicolor' 'virginica' 'versicolor'
 'versicolor' 'versicolor' 'versicolor' 'setosa' 'versicolor' 'versicolor'
 'setosa' 'setosa' 'virginica' 'versicolor' 'setosa' 'setosa' 'virginica'
 'setosa' 'setosa' 'versicolor' 'versicolor' 'setosa']

The accuracy score of model when compared with twenty percent original test
values is 1.0
```

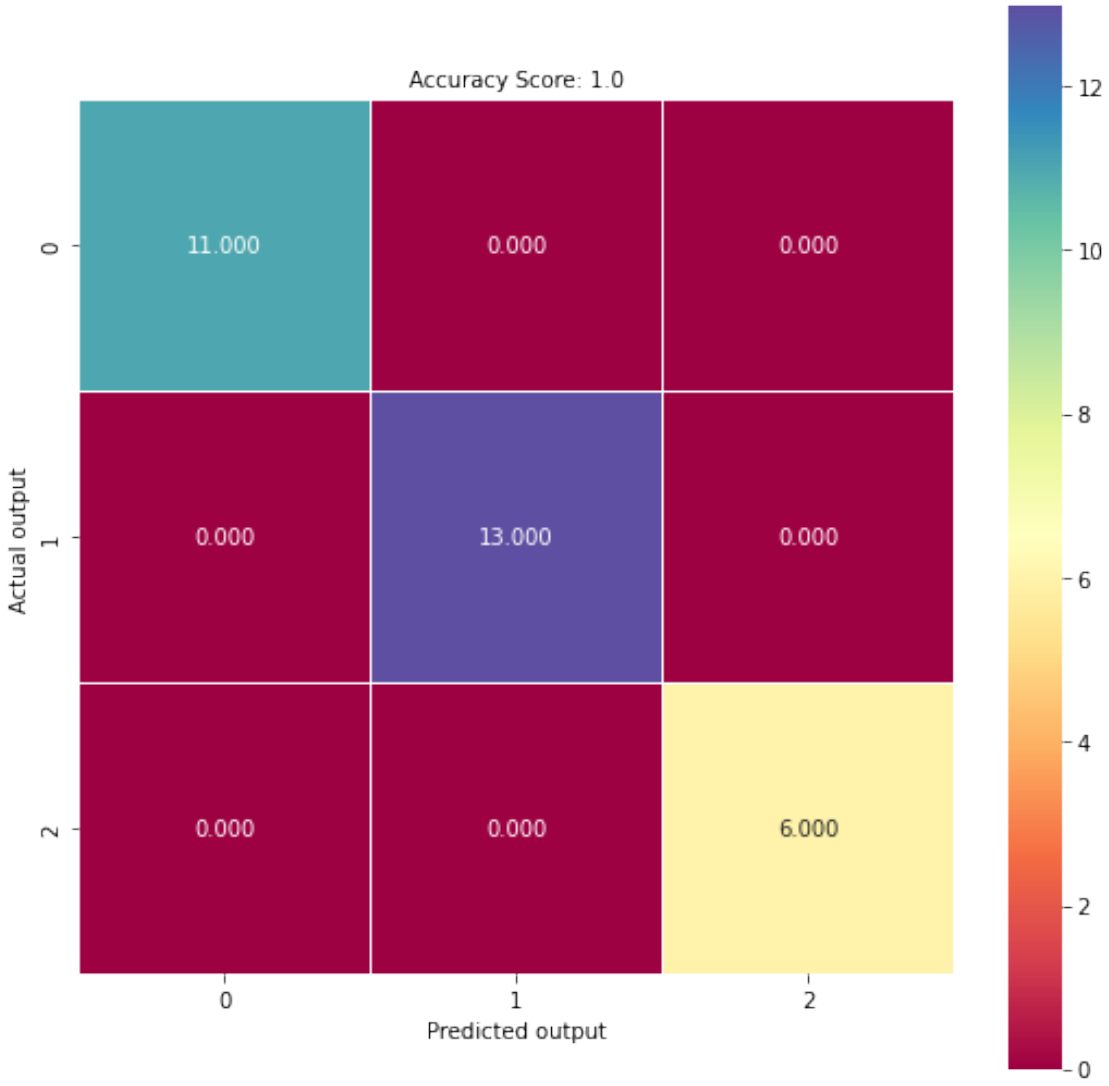
0.0.3 Confusion Matrix (y_test,predictions)

[ ]:
```
from sklearn.metrics import confusion_matrix
predictions = model.predict(X_test)
cm= confusion_matrix(y_test, predictions)
cm
```

[ ]:
```
array([[11,  0,  0],
       [ 0, 13,  0],
       [ 0,  0,  6]], dtype=int64)
```

[ ]:
```
# Heatmap to visualize COnfusion Matrix
plt.figure(figsize=(9,9))
sns.heatmap(cm, annot=True, fmt=".3f", linewidths=.5, square=True,␣
 ↪cmap='Spectral')
plt.ylabel('Actual output')
plt.xlabel('Predicted output')
all_sampletitle= 'Accuracy Score: {0}'.format(score)
plt.title(all_sampletitle,size =10)
```

[ ]:
```
Text(0.5, 1.0, 'Accuracy Score: 1.0')
```


Accuracy Score: 1.0
11.000
0.000
0.000
0.000
13.000
0.000
0.000
0.000
6.000
Actual output
Predicted output
0
1
2
12
10
8
6
4
2
0

[ ]: