assn6

May 22, 2023

[3]:
```
#Aishwarya kelgandre Roll no.73 batch T3
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
s1 =pd.Series(range(1,10,1))
s1
import pandas as pd
from matplotlib import pyplot as plt


df = pd.read_csv("E:\\TRINITY ACADEMY OF ENGINEERING PUNE\\TE␣
 ↪2022-23\\assignment\\dsbda\\csv\\iris.csv")
df.head(10)
```

[3]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa
5	6	5.4	3.9	1.7	0.4	Iris-setosa
6	7	4.6	3.4	1.4	0.3	Iris-setosa
7	8	5.0	3.4	1.5	0.2	Iris-setosa
8	9	4.4	2.9	1.4	0.2	Iris-setosa
9	10	4.9	3.1	1.5	0.1	Iris-setosa

[4]:
```
X=df.iloc[:,0:4]
y=df.iloc[:,-1]
y
```

[4]:
```
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
            …
145    Iris-virginica
146    Iris-virginica
```

```
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Species, Length: 150, dtype: object
```

[5]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,train_size=0.
 ↪8,random_state=1)
X_test
```

[5]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm
14	15	5.8	4.0	1.2
98	99	5.1	2.5	3.0
75	76	6.6	3.0	4.4
16	17	5.4	3.9	1.3
131	132	7.9	3.8	6.4
56	57	6.3	3.3	4.7
141	142	6.9	3.1	5.1
44	45	5.1	3.8	1.9
29	30	4.7	3.2	1.6
120	121	6.9	3.2	5.7
94	95	5.6	2.7	4.2
5	6	5.4	3.9	1.7
102	103	7.1	3.0	5.9
51	52	6.4	3.2	4.5
78	79	6.0	2.9	4.5
42	43	4.4	3.2	1.3
92	93	5.8	2.6	4.0
66	67	5.6	3.0	4.5
31	32	5.4	3.4	1.5
35	36	5.0	3.2	1.2
90	91	5.5	2.6	4.4
84	85	5.4	3.0	4.5
77	78	6.7	3.0	5.0
40	41	5.0	3.5	1.3
125	126	7.2	3.2	6.0
99	100	5.7	2.8	4.1
33	34	5.5	4.2	1.4
19	20	5.1	3.8	1.5
73	74	6.1	2.8	4.7
146	147	6.3	2.5	5.0

[6]:
```
from sklearn.preprocessing import LabelEncoder
la_object = LabelEncoder()
y = la_object.fit_transform(y)
y
```

[6]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
       2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```

[7]:
```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(X_train, y_train)
```

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

[8]:
```
y_predicted = model.predict(X_test)

y_predicted
```

[8]:
```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-virginica',
       'Iris-setosa', 'Iris-setosa', 'Iris-virginica', 'Iris-versicolor',
       'Iris-setosa', 'Iris-virginica', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa',
       'Iris-virginica', 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa',
       'Iris-versicolor', 'Iris-virginica'], dtype='<U15')
```

[9]:
```
model.score(X_test,y_test)
```

[9]:
```
1.0
```

[10]:
```
model.score(X_test,y_test)
```

[10]:
```
1.0
```

[11]:
```
from sklearn.metrics import confusion_matrix,classification_report
cm = confusion_matrix(y_test, y_predicted)
cm
```

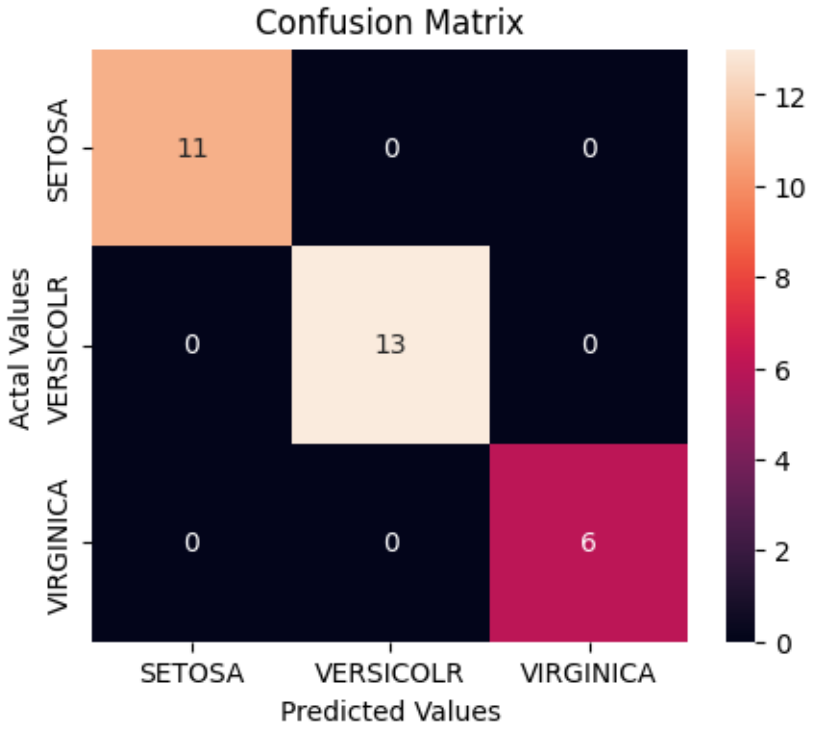
[11]:
```
array([[11,  0,  0],
       [ 0, 13,  0],
       [ 0,  0,  6]], dtype=int64)
```

[13]:
```
cl_report=classification_report(y_test,y_predicted)
cl_report
```

[13]:
```
'                  precision    recall  f1-score   support\n\n    Iris-setosa
1.00      1.00      1.00        11\nIris-versicolor       1.00      1.00
1.00        13\n Iris-virginica       1.00      1.00      1.00         6\n\n
accuracy                           1.00        30\n      macro avg       1.00
1.00      1.00        30\n   weighted avg       1.00      1.00      1.00
30\n'
```

[14]:
```
cm_df = pd.DataFrame(cm,
 index = ['SETOSA','VERSICOLR','VIRGINICA'],
 columns = ['SETOSA','VERSICOLR','VIRGINICA'])
```

[15]:
```
import seaborn as sns
plt.figure(figsize=(5,4))
sns.heatmap(cm_df, annot=True)
plt.title('Confusion Matrix')
plt.ylabel('Actal Values')
plt.xlabel('Predicted Values')
plt.show()
```



[17]:
```
def accuracy_cm(tp,fn,fp,tn):
 return (tp+tn)/(tp+fp+tn+fn)
def precision_cm(tp,fn,fp,tn):
 return tp/(tp+fp)
```

```
def recall_cm(tp,fn,fp,tn):
 return tp/(tp+fn)
def f1_score(tp,fn,fp,tn):
 return (2/((1/recall_cm(tp,fn,fp,tn))+precision_cm(tp,fn,fp,tn)))
def error_rate_cm(tp,fn,fp,tn):
 return 1-accuracy_cm(tp,fn,fp,tn)
```

[18]:
```
tp = cm[2][2]
fn = cm[2][0]+cm[2][1]
fp = cm[0][2]+cm[1][2]
tn = cm[0][0]+cm[0][1]+cm[1][0]+cm[1][1]
print("For Virginica \n")
print("Accuracy : ",accuracy_cm(tp,fn,fp,tn))
print("Precision : ",precision_cm(tp,fn,fp,tn))
print("Recall : ",recall_cm(tp,fn,fp,tn))
print("F1-Score : ",f1_score(tp,fn,fp,tn))
print("Error rate : ",error_rate_cm(tp,fn,fp,tn))
```

```
For Virginica

Accuracy :  1.0
Precision :  1.0
Recall :  1.0
F1-Score :  1.0
Error rate :  0.0
```

[ ]: