naive_bayes

May 11, 2022

0.0.1 Data Analytics III

- Implement Simple Naïve Bayes classification algorithm using Python/R on iris.csv dataset.
- Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

[2]:
```
#Importing the Libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
url="Iris.csv"
df = pd.read_csv(url)
df.head(10)
```

[2]:

	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

[6]:
```
#seperating input and output for Naive Bayes implementation
X = df.iloc[:,1:5].values
Y = df['Species'].values
```

[7]:
```
# Splitting the dataset into the Training set and Test set

from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.3)
```

[8]:
```
# Feature Scaling

from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

[10]:
```
#Training the Naive Bayes Classification model on the Training Set

from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
classifier.fit(X_train, Y_train)
```

[10]:
```
GaussianNB()
```

[11]:
```
#Predicting the Test set results
y_pred = classifier.predict(X_test)
print(y_pred)
```

```
['Iris-setosa' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-virginica' 'Iris-virginica' 'Iris-virginica' 'Iris-virginica'
 'Iris-virginica' 'Iris-setosa' 'Iris-virginica' 'Iris-versicolor'
 'Iris-setosa' 'Iris-virginica' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-virginica' 'Iris-versicolor' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-virginica' 'Iris-virginica' 'Iris-setosa' 'Iris-setosa'
 'Iris-setosa' 'Iris-virginica' 'Iris-versicolor' 'Iris-versicolor'
 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa' 'Iris-virginica'
 'Iris-setosa' 'Iris-versicolor' 'Iris-setosa' 'Iris-virginica'
 'Iris-setosa' 'Iris-versicolor' 'Iris-setosa' 'Iris-setosa'
 'Iris-virginica' 'Iris-setosa' 'Iris-versicolor' 'Iris-virginica'
 'Iris-setosa']
```

[13]:
```
y_pred = classifier.predict(X_test)
y_pred

#Comparing the Real Values with Predicted Values
cmp = pd.DataFrame({'Real Values':Y_test, 'Predicted Values':y_pred})
print(cmp)
```

```
        Real Values Predicted Values
0       Iris-setosa      Iris-setosa
1   Iris-versicolor  Iris-versicolor
2   Iris-versicolor  Iris-versicolor
3   Iris-versicolor  Iris-versicolor
4    Iris-virginica   Iris-virginica
5    Iris-virginica   Iris-virginica
6    Iris-virginica   Iris-virginica
7    Iris-virginica   Iris-virginica
```

```
8    Iris-virginica   Iris-virginica
9       Iris-setosa      Iris-setosa
10  Iris-versicolor   Iris-virginica
11  Iris-versicolor  Iris-versicolor
12      Iris-setosa      Iris-setosa
13   Iris-virginica   Iris-virginica
14  Iris-versicolor  Iris-versicolor
15   Iris-virginica  Iris-versicolor
16   Iris-virginica   Iris-virginica
17  Iris-versicolor  Iris-versicolor
18  Iris-versicolor  Iris-versicolor
19  Iris-versicolor  Iris-versicolor
20   Iris-virginica   Iris-virginica
21   Iris-virginica   Iris-virginica
22      Iris-setosa      Iris-setosa
23      Iris-setosa      Iris-setosa
24      Iris-setosa      Iris-setosa
25   Iris-virginica   Iris-virginica
26  Iris-versicolor  Iris-versicolor
27  Iris-versicolor  Iris-versicolor
28  Iris-versicolor  Iris-versicolor
29      Iris-setosa      Iris-setosa
30      Iris-setosa      Iris-setosa
31   Iris-virginica   Iris-virginica
32      Iris-setosa      Iris-setosa
33  Iris-versicolor  Iris-versicolor
34      Iris-setosa      Iris-setosa
35   Iris-virginica   Iris-virginica
36      Iris-setosa      Iris-setosa
37  Iris-versicolor  Iris-versicolor
38      Iris-setosa      Iris-setosa
39      Iris-setosa      Iris-setosa
40   Iris-virginica   Iris-virginica
41      Iris-setosa      Iris-setosa
42  Iris-versicolor  Iris-versicolor
43   Iris-virginica   Iris-virginica
44      Iris-setosa      Iris-setosa
```

[17]:
```
#Confusion Matrix and Accuracy

from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_test, y_pred)
cm
```

[17]:
```
array([[15,  0,  0],
       [ 0, 14,  1],
       [ 0,  1, 14]])
```

[22]:
```
from sklearn.metrics import accuracy_score
print ("Accuracy : ", accuracy_score(Y_test, y_pred))
```

```
Accuracy :  0.9555555555555556
```

[23]:
```
from sklearn.metrics import precision_score
print ("precision : ",precision_score(Y_test, y_pred,average='macro'))
```

```
precision :  0.9555555555555556
```

[25]:
```
from sklearn.metrics import recall_score
print ("recall : ",recall_score(Y_test, y_pred,average='macro'))
```

```
recall :  0.9555555555555556
```