yash-dsbdal-a6

February 22, 2024

Title of the Assignment: 1. Implement Simple Naïve Bayes classification algorithm using Python/R on iris.csv dataset. 2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.

Import libraries and create alias for Pandas, Numpy

[ ]:
```
import pandas as pd
import numpy as np
```

Import the Iris Dataset

[ ]:
```
from google.colab import files
files.upload()
```

```
<IPython.core.display.HTML object>
```

Basic Operations

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

[ ]:
```
df.describe()
```

[ ]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

Check for Null Values

[ ]:
```
df.isnull().sum()
```

[ ]:
```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
```

```
PetalWidthCm     0
Species          0
dtype: int64
```

[ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             150 non-null    int64
 1   SepalLengthCm  150 non-null    float64
 2   SepalWidthCm   150 non-null    float64
 3   PetalLengthCm  150 non-null    float64
 4   PetalWidthCm   150 non-null    float64
 5   Species        150 non-null    object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

Use Naive Bayes algorithm(Train the Machine) to Create Model

[ ]:
```
X = df.drop(['Species'], axis = 1)
Y = df['Species']
```

[ ]:
```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(X, Y, test_size =0.
  ↪2,random_state = 0)
```

[ ]:
```
from sklearn.naive_bayes import GaussianNB
gaussian = GaussianNB()
```

[ ]:
```
gaussian.fit(xtrain, ytrain)
```

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

Predict the y_pred for all values of train_x and test_x

[ ]:
```
y_pred = gaussian.predict(xtest)
```

[ ]:
```
print(xtrain)
print("------------\n")
print(xtest)
print("------------\n")
print(ytrain)
print("------------\n")
print(ytest)
print("------------\n")
```

```
print(y_pred)
```

```
      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
137  138            6.4           3.1            5.5           1.8
84    85            5.4           3.0            4.5           1.5
27    28            5.2           3.5            1.5           0.2
127  128            6.1           3.0            4.9           1.8
132  133            6.4           2.8            5.6           2.2
..   …              …             …              …             …
9     10            4.9           3.1            1.5           0.1
103  104            6.3           2.9            5.6           1.8
67    68            5.8           2.7            4.1           1.0
117  118            7.7           3.8            6.7           2.2
47    48            4.6           3.2            1.4           0.2

[120 rows x 5 columns]
------------

      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
114  115            5.8           2.8            5.1           2.4
62    63            6.0           2.2            4.0           1.0
33    34            5.5           4.2            1.4           0.2
107  108            7.3           2.9            6.3           1.8
7      8            5.0           3.4            1.5           0.2
100  101            6.3           3.3            6.0           2.5
40    41            5.0           3.5            1.3           0.3
86    87            6.7           3.1            4.7           1.5
76    77            6.8           2.8            4.8           1.4
71    72            6.1           2.8            4.0           1.3
134  135            6.1           2.6            5.6           1.4
51    52            6.4           3.2            4.5           1.5
73    74            6.1           2.8            4.7           1.2
54    55            6.5           2.8            4.6           1.5
63    64            6.1           2.9            4.7           1.4
37    38            4.9           3.1            1.5           0.1
78    79            6.0           2.9            4.5           1.5
90    91            5.5           2.6            4.4           1.2
45    46            4.8           3.0            1.4           0.3
16    17            5.4           3.9            1.3           0.4
121  122            5.6           2.8            4.9           2.0
66    67            5.6           3.0            4.5           1.5
24    25            4.8           3.4            1.9           0.2
8      9            4.4           2.9            1.4           0.2
126  127            6.2           2.8            4.8           1.8
22    23            4.6           3.6            1.0           0.2
44    45            5.1           3.8            1.9           0.4
97    98            6.2           2.9            4.3           1.3
```

```
93    94            5.0           2.3            3.3           1.0
26    27            5.0           3.4            1.6           0.4
------------

137     Iris-virginica
84     Iris-versicolor
27         Iris-setosa
127     Iris-virginica
132     Iris-virginica
            …
9          Iris-setosa
103     Iris-virginica
67     Iris-versicolor
117     Iris-virginica
47         Iris-setosa
Name: Species, Length: 120, dtype: object
------------

114     Iris-virginica
62     Iris-versicolor
33         Iris-setosa
107     Iris-virginica
7          Iris-setosa
100     Iris-virginica
40         Iris-setosa
86     Iris-versicolor
76     Iris-versicolor
71     Iris-versicolor
134     Iris-virginica
51     Iris-versicolor
73     Iris-versicolor
54     Iris-versicolor
63     Iris-versicolor
37         Iris-setosa
78     Iris-versicolor
90     Iris-versicolor
45         Iris-setosa
16         Iris-setosa
121     Iris-virginica
66     Iris-versicolor
24         Iris-setosa
8          Iris-setosa
126     Iris-virginica
22         Iris-setosa
44         Iris-setosa
97     Iris-versicolor
93     Iris-versicolor
26         Iris-setosa
```

```
Name: Species, dtype: object
------------

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

Evaluate the performance of Model for train_y and test_y

[ ]:
```
from sklearn.metrics import␣
 ↪precision_score,confusion_matrix,accuracy_score,recall_score,␣
 ↪classification_report
cm= confusion_matrix(ytest, y_pred)
```

Confusion Matrix

[ ]:
```
cm= confusion_matrix(ytest, y_pred)
cm
```

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

Accuracy Score

[ ]:
```
print ("Accuracy : ", accuracy_score(ytest, y_pred))
```

```
Accuracy :  1.0
```

Error Rate

[ ]:
```
error_rate = 1- accuracy_score(ytest, y_pred)
```

[ ]:
```
error_rate
```

[ ]:
```
0.0
```

Classification

[ ]:
```
print("classification report: ",classification_report(ytest, y_pred))
```

```
classification report:                    precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        11
Iris-versicolor       1.00      1.00      1.00        13
 Iris-virginica       1.00      1.00      1.00         6
```

```
    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```