Assignment No. 6

Aim :Data Analytics III

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.

Sol^n:

OUTPUT :

{'data': array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4],
       [4.6, 3.4, 1.4, 0.3],
       [5. , 3.4, 1.5, 0.2],
       [4.4, 2.9, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.1],
       [5.4, 3.7, 1.5, 0.2],
       [4.8, 3.4, 1.6, 0.2],
       [4.8, 3. , 1.4, 0.1],
       [4.3, 3. , 1.1, 0.1],
       [5.8, 4. , 1.2, 0.2],
       [5.7, 4.4, 1.5, 0.4],
       [5.4, 3.9, 1.3, 0.4],
       [5.1, 3.5, 1.4, 0.3],
       [5.7, 3.8, 1.7, 0.3],
       [5.1, 3.8, 1.5, 0.3],
       [5.4, 3.4, 1.7, 0.2],
       [5.1, 3.7, 1.5, 0.4],
       [4.6, 3.6, 1. , 0.2],
       [5.1, 3.3, 1.7, 0.5],
       [4.8, 3.4, 1.9, 0.2],
       [5. , 3. , 1.6, 0.2],
       [5. , 3.4, 1.6, 0.4],
       [5.2, 3.5, 1.5, 0.2],
       [5.2, 3.4, 1.4, 0.2],
       [4.7, 3.2, 1.6, 0.2],
       [4.8, 3.1, 1.6, 0.2],
       [5.4, 3.4, 1.5, 0.4],
       [5.2, 4.1, 1.5, 0.1],
       [5.5, 4.2, 1.4, 0.2],
       [4.9, 3.1, 1.5, 0.2],
       [5. , 3.2, 1.2, 0.2],
       [5.5, 3.5, 1.3, 0.2],
       [4.9, 3.6, 1.4, 0.1],
       [4.4, 3. , 1.3, 0.2],

DSBDAL6 Draft saved

File   Edit   View   Run   Add-ons   Help

+   Code   ▸ Run All   ● Draft Session (18m)

```
[6. , 2.2, 4. , 1. ],
[6.1, 2.9, 4.7, 1.4],
[5.6, 2.9, 3.6, 1.3],
[6.7, 3.1, 4.4, 1.4],
[5.6, 3. , 4.5, 1.5],
[5.8, 2.7, 4.1, 1. ],
[6.2, 2.2, 4.5, 1.5],
[5.6, 2.5, 3.9, 1.1],
[5.9, 3.2, 4.8, 1.8],
[6.1, 2.8, 4. , 1.3],
[6.3, 2.5, 4.9, 1.5],
[6.1, 2.8, 4.7, 1.2],
[6.4, 2.9, 4.3, 1.3],
[6.6, 3. , 4.4, 1.4],
[6.8, 2.8, 4.8, 1.4],
[6.7, 3. , 5. , 1.7],
[6. , 2.9, 4.5, 1.5],
[5.7, 2.6, 3.5, 1. ],
[5.5, 2.4, 3.8, 1.1],
[5.5, 2.4, 3.7, 1. ],
[5.8, 2.7, 3.9, 1.2],
[6. , 2.7, 5.1, 1.6],
[5.4, 3. , 4.5, 1.5],
[6. , 3.4, 4.5, 1.6],
[6.7, 3.1, 4.7, 1.5],
[6.3, 2.3, 4.4, 1.3],
[5.6, 3. , 4.1, 1.3],
[5.5, 2.5, 4. , 1.3],
[5.5, 2.6, 4.4, 1.2],
[6.1, 3. , 4.6, 1.4],
[5.8, 2.6, 4. , 1.2],
[5. , 2.3, 3.3, 1. ],
[5.6, 2.7, 4.2, 1.3],
[5.7, 3. , 4.2, 1.2],
[5.7, 2.9, 4.2, 1.3],
[6.2, 2.9, 4.3, 1.3],
[5.1, 2.5, 3. , 1.1],
[5.7, 2.8, 4.1, 1.3],
[6.3, 3.3, 6. , 2.5],
[5.8, 2.7, 5.1, 1.9],
```

Console

Data   + Add data

Input

irisdataset
   iris.data

Output (44.1MB / 19.6GB)

▸ /kaggle/working

Settings

Schedule a notebook run

Code Help

FIND CODE HELP

Find Code Help

Search for examples of how to do things

```
[5.6, 2.7, 4.2, 1.3],
[5.7, 3. , 4.2, 1.2],
[5.7, 2.9, 4.2, 1.3],
[6.2, 2.9, 4.3, 1.3],
[5.1, 2.5, 3. , 1.1],
[5.7, 2.8, 4.1, 1.3],
[6.3, 3.3, 6. , 2.5],
[5.8, 2.7, 5.1, 1.9],
[7.1, 3. , 5.9, 2.1],
[6.3, 2.9, 5.6, 1.8],
[6.5, 3. , 5.8, 2.2],
[7.6, 3. , 6.6, 2.1],
[4.9, 2.5, 4.5, 1.7],
[7.3, 2.9, 6.3, 1.8],
[6.7, 2.5, 5.8, 1.8],
[7.2, 3.6, 6.1, 2.5],
[6.5, 3.2, 5.1, 2. ],
[6.4, 2.7, 5.3, 1.9],
[6.8, 3. , 5.5, 2.1],
[5.7, 2.5, 5. , 2. ],
[5.8, 2.8, 5.1, 2.4],
[6.4, 3.2, 5.3, 2.3],
[6.5, 3. , 5.5, 1.8],
[7.7, 3.8, 6.7, 2.2],
[7.7, 2.6, 6.9, 2.3],
[6. , 2.2, 5. , 1.5],
[6.9, 3.2, 5.7, 2.3],
[5.6, 2.8, 4.9, 2. ],
[7.7, 2.8, 6.7, 2. ],
[6.3, 2.7, 4.9, 1.8],
[6.7, 3.3, 5.7, 2.1],
[7.2, 3.2, 6. , 1.8],
[6.2, 2.8, 4.8, 1.8],
[6.1, 3. , 4.9, 1.8],
[6.4, 2.8, 5.6, 2.1],
[7.2, 3. , 5.8, 1.6],
[7.4, 2.8, 6.1, 1.9],
[7.9, 3.8, 6.4, 2. ],
[6.4, 2.8, 5.6, 2.2],
[6.3, 2.8, 5.1, 1.5],
```

Console

Data   + Add data

Input

irisdataset
   iris.data

Output (44.1MB / 19.6GB)

▸ /kaggle/working

Settings

Schedule a notebook run

Code Help

FIND CODE HELP

Find Code Help

Search for examples of how to do things

[6.7, 3.3, 5.7, 2.1],
[7.2, 3.2, 6. , 1.8],
[6.2, 2.8, 4.8, 1.8],
[6.1, 3. , 4.9, 1.8],
[6.4, 2.8, 5.6, 2.1],
[7.2, 3. , 5.8, 1.6],
[7.4, 2.8, 6.1, 1.9],
[7.9, 3.8, 6.4, 2.],
[6.4, 2.8, 5.6, 2.2],
[6.3, 2.8, 5.1, 1.5],
[6.1, 2.6, 5.6, 1.4],
[7.7, 3. , 6.1, 2.3],
[6.3, 3.4, 5.6, 2.4],
[6.4, 3.1, 5.5, 1.8],
[6. , 3. , 4.8, 1.8],
[6.9, 3.1, 5.4, 2.1],
[6.7, 3.1, 5.6, 2.4],
[6.9, 3.1, 5.1, 2.3],
[5.8, 2.7, 5.1, 1.9],
[6.8, 3.2, 5.9, 2.3],
[6.7, 3.3, 5.7, 2.5],
[6.7, 3. , 5.2, 2.3],
[6.3, 2.5, 5. , 1.9],
[6.5, 3. , 5.2, 2.],
[6.2, 3.4, 5.4, 2.3],
[5.9, 3. , 5.1, 1.8]]), 'target': 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, 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]), 'frame': None, 'target_names': array(['setosa', 'versicolor', 'virginica'], dtype='<U
10'), 'DESCR': '.. _iris_dataset:\n\nIris plants dataset\n--------------------\n\n**Data Set Characteristics:**\n\n   :Number of Instances: 150 (50
in each of three classes)\n   :Number of Attributes: 4 numeric, predictive attributes and the class\n   :Attribute Information:\n   - sepal l
ength in cm\n   - sepal width in cm\n   - petal length in cm\n   - petal width in cm\n   - class:\n   - Iris-Setosa
\n   - Iris-Versicolour\n   - Iris-Virginica\n\n   :Summary Statistics:\n\n   ============== ==== ==== ======= ===== ====
========= ==== ====\n   Min Max   Mean   SD   Class Correlation\n   ============== ==== ==== ======= ===== ==== ===== ======
============\n   sepal length: 4.3 7.9   5.84   0.83   0.7826\n   sepal width: 2.0 4.4   3.05   0.43   -0.4194\n   petal length: 1.0
6.9 3.76   1.76   0.9490 (high!)\n   petal width: 0.1 2.5   1.20   0.76   0.9565 (high!)\n   ============== ==== ==== ======= ===== ====
==============\n\n   :Missing Attribute Values: None\n   :Class Distribution: 33.3% for each of 3 classes.\n   :Creator: R.A. Fisher\n   :Dono

DSBDAL6 Draft saved

Share   Save Version   0

File Edit View Run Add-ons Help

Code   ● Draft Session (19m)   Run All

```
[4]: df.describe
```

```
[4]: <bound method NDFrame.describe of        5.1  3.5  1.4  0.2    Iris-setosa
     0    4.9  3.0  1.4  0.2    Iris-setosa
     1    4.7  3.2  1.3  0.2    Iris-setosa
     2    4.6  3.1  1.5  0.2    Iris-setosa
     3    5.0  3.6  1.4  0.2    Iris-setosa
     4    5.4  3.9  1.7  0.4    Iris-setosa
     ..   ...  ...  ...  ...          ...
     144  6.7  3.0  5.2  2.3  Iris-virginica
     145  6.3  2.5  5.0  1.9  Iris-virginica
     146  6.5  3.0  5.2  2.0  Iris-virginica
     147  6.2  3.4  5.4  2.3  Iris-virginica
     148  5.9  3.0  5.1  1.8  Iris-virginica

     [149 rows x 5 columns]>
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 149 entries, 0 to 148
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   5.1           149 non-null    float64
 1   3.5           149 non-null    float64
 2   1.4           149 non-null    float64
 3   0.2           149 non-null    float64
 4   Iris-setosa   149 non-null    object
dtypes: float64(4), object(1)
memory usage: 5.9+ KB
```

Console

Data   + Add data

Input

irisdataset
    iris.data

Output (44.1MB / 19.6GB)

/kaggle/working

Settings

Schedule a notebook run

Code Help

FIND CODE HELP

Find Code Help

Search for examples of how to do things

DSBDAL6 Draft saved

Share   Save Version   0

File Edit View Run Add-ons Help

Code   ● Draft Session (19m)   Run All

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

```
[6]:    5.1  3.5  1.4  0.2  Iris-setosa
     0  4.9  3.0  1.4  0.2  Iris-setosa
     1  4.7  3.2  1.3  0.2  Iris-setosa
     2  4.6  3.1  1.5  0.2  Iris-setosa
     3  5.0  3.6  1.4  0.2  Iris-setosa
     4  5.4  3.9  1.7  0.4  Iris-setosa
```

```
[9]: print(iris.feature_names)
```

```
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
```

```
[10]: print(iris.target_names)
```

```
['setosa' 'versicolor' 'virginica']
```

```
[12]: X=pd.DataFrame(iris['data'])
      print(X.head())
```

Console

Data   + Add data

Input

irisdataset
    iris.data

Output (44.1MB / 19.6GB)

/kaggle/working

Settings

Schedule a notebook run

Code Help

FIND CODE HELP

Find Code Help

Search for examples of how to do things

```
[12]: X=pd.DataFrame(iris['data'])
      print(X.head())

            0    1    2    3
      0   5.1  3.5  1.4  0.2
      1   4.9  3.0  1.4  0.2
      2   4.7  3.2  1.3  0.2
      3   4.6  3.1  1.5  0.2
      4   5.0  3.6  1.4  0.2
```

```
[13]: y=print(iris.target)

      [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 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 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 2 2 2 2
       2 2]
```

```
[16]: from sklearn.model_selection import train_test_split
      x_train, x_test, y_train, y_test=train_test_split(iris.data, iris.target, test_size=0.30, random_state=100)
```

```
[17]: from sklearn.naive_bayes import GaussianNB
```

```
[17]: from sklearn.naive_bayes import GaussianNB
```

```
[18]: gnb=GaussianNB()
      gnb.fit(x_train, y_train)
      y_pred=gnb.predict(x_test)
      print(y_pred)

      [2 0 2 0 2 2 0 0 2 0 0 2 0 0 2 1 1 2 2 2 2 0 2 0 1 2 1 0 1 2 1 1 1 0 0 1 0
       1 2 2 0 1 2 2 0]
```

```
[20]: from sklearn import metrics
      print(metrics.accuracy_score(y_test, y_pred))

      0.9555555555555556
```

```
[21]: from sklearn.metrics import confusion_matrix
      cm=np.array(confusion_matrix(y_test, y_pred))
      cm

[21]: array([[16,  0,  0],
             [ 0, 10,  1],
             [ 0,  1, 17]])
```

```python
df.columns
```

```
Index(['5.1', '3.5', '1.4', '0.2', 'Iris-setosa'], dtype='object')
```

```python
df.shape
```

```
(149, 5)
```

```python
df.dtypes
```

```
5.1            float64
3.5            float64
1.4            float64
0.2            float64
Iris-setosa     object
dtype: object
```

```python
df.isnull().sum()
```

```
5.1          0
3.5          0
1.4          0
0.2          0
```

```python
df.isnull().sum()
```

```
5.1          0
3.5          0
1.4          0
0.2          0
Iris-setosa  0
dtype: int64
```

```python
df.corr()
```

	5.1	3.5	1.4	0.2
5.1	1.000000	-0.103784	0.871283	0.816971
3.5	-0.103784	1.000000	-0.415218	-0.350733
1.4	0.871283	-0.415218	1.000000	0.962314
0.2	0.816971	-0.350733	0.962314	1.000000

```python
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

DSBDAL6 Draft saved

File Edit View Run Add-ons Help

Share   Save Version   0

Code

Draft Session (27m)

Data   + Add data

Input

- irisdataset
  - iris.data

Output (44.1MB / 19.6GB)

- /kaggle/working

Settings

Schedule a notebook run

Code Help

FIND CODE HELP

Find Code Help

Search for examples of how to do things

```python
[30]: import seaborn as sns
      import matplotlib.pyplot as plt
      %matplotlib inline
```

```python
[31]: sns.heatmap(df.corr())
```

[31]: <AxesSubplot:>


```python
[32]: from sklearn.metrics import classification_report
      print(classification_report(y_test, y_pred))
```

Console

```python
[32]: from sklearn.metrics import classification_report
      print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        16
           1       0.91      0.91      0.91        11
           2       0.94      0.94      0.94        18

    accuracy                           0.96        45
   macro avg       0.95      0.95      0.95        45
weighted avg       0.96      0.96      0.96        45
```

+ Code   + Markdown

Console