practical-6-piyusha

April 3, 2025

0.1 23CO315 Piyusha Supe

Practical- 6 - 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.

[2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score,␣
↪recall_score, f1_score, classification_report

from google.colab import files
files.upload()
```

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

```
Saving IRIS.csv to IRIS.csv
```

[2]:

```
{'IRIS.csv': b'sepal_length,sepal_width,petal_length,petal_width,species\r\n5.1,
3.5,1.4,0.2,Iris-setosa\r\n4.9,3,1.4,0.2,Iris-setosa\r\n4.7,3.2,1.3,0.2,Iris-
setosa\r\n4.6,3.1,1.5,0.2,Iris-setosa\r\n5,3.6,1.4,0.2,Iris-
setosa\r\n5.4,3.9,1.7,0.4,Iris-setosa\r\n4.6,3.4,1.4,0.3,Iris-
setosa\r\n5,3.4,1.5,0.2,Iris-setosa\r\n4.4,2.9,1.4,0.2,Iris-
setosa\r\n4.9,3.1,1.5,0.1,Iris-setosa\r\n5.4,3.7,1.5,0.2,Iris-
setosa\r\n4.8,3.4,1.6,0.2,Iris-setosa\r\n4.8,3,1.4,0.1,Iris-
setosa\r\n4.3,3,1.1,0.1,Iris-setosa\r\n5.8,4,1.2,0.2,Iris-
setosa\r\n5.7,4.4,1.5,0.4,Iris-setosa\r\n5.4,3.9,1.3,0.4,Iris-
setosa\r\n5.1,3.5,1.4,0.3,Iris-setosa\r\n5.7,3.8,1.7,0.3,Iris-
setosa\r\n5.1,3.8,1.5,0.3,Iris-setosa\r\n5.4,3.4,1.7,0.2,Iris-
setosa\r\n5.1,3.7,1.5,0.4,Iris-setosa\r\n4.6,3.6,1,0.2,Iris-
setosa\r\n5.1,3.3,1.7,0.5,Iris-setosa\r\n4.8,3.4,1.9,0.2,Iris-
setosa\r\n5,3,1.6,0.2,Iris-setosa\r\n5,3.4,1.6,0.4,Iris-
setosa\r\n5.2,3.5,1.5,0.2,Iris-setosa\r\n5.2,3.4,1.4,0.2,Iris-
setosa\r\n4.7,3.2,1.6,0.2,Iris-setosa\r\n4.8,3.1,1.6,0.2,Iris-
```

setosa\r\n5.4,3.4,1.5,0.4,Iris-setosa\r\n5.2,4.1,1.5,0.1,Iris-
setosa\r\n5.5,4.2,1.4,0.2,Iris-setosa\r\n4.9,3.1,1.5,0.1,Iris-
setosa\r\n5,3.2,1.2,0.2,Iris-setosa\r\n5.5,3.5,1.3,0.2,Iris-
setosa\r\n4.9,3.1,1.5,0.1,Iris-setosa\r\n4.4,3,1.3,0.2,Iris-
setosa\r\n5.1,3.4,1.5,0.2,Iris-setosa\r\n5,3.5,1.3,0.3,Iris-
setosa\r\n4.5,2.3,1.3,0.3,Iris-setosa\r\n4.4,3.2,1.3,0.2,Iris-
setosa\r\n5,3.5,1.6,0.6,Iris-setosa\r\n5.1,3.8,1.9,0.4,Iris-
setosa\r\n4.8,3,1.4,0.3,Iris-setosa\r\n5.1,3.8,1.6,0.2,Iris-
setosa\r\n4.6,3.2,1.4,0.2,Iris-setosa\r\n5.3,3.7,1.5,0.2,Iris-
setosa\r\n5,3.3,1.4,0.2,Iris-setosa\r\n7,3.2,4.7,1.4,Iris-
versicolor\r\n6.4,3.2,4.5,1.5,Iris-versicolor\r\n6.9,3.1,4.9,1.5,Iris-
versicolor\r\n5.5,2.3,4,1.3,Iris-versicolor\r\n6.5,2.8,4.6,1.5,Iris-
versicolor\r\n5.7,2.8,4.5,1.3,Iris-versicolor\r\n6.3,3.3,4.7,1.6,Iris-
versicolor\r\n4.9,2.4,3.3,1,Iris-versicolor\r\n6.6,2.9,4.6,1.3,Iris-
versicolor\r\n5.2,2.7,3.9,1.4,Iris-versicolor\r\n5,2,3.5,1,Iris-
versicolor\r\n5.9,3,4.2,1.5,Iris-versicolor\r\n6,2.2,4,1,Iris-
versicolor\r\n6.1,2.9,4.7,1.4,Iris-versicolor\r\n5.6,2.9,3.6,1.3,Iris-
versicolor\r\n6.7,3.1,4.4,1.4,Iris-versicolor\r\n5.6,3,4.5,1.5,Iris-
versicolor\r\n5.8,2.7,4.1,1,Iris-versicolor\r\n6.2,2.2,4.5,1.5,Iris-
versicolor\r\n5.6,2.5,3.9,1.1,Iris-versicolor\r\n5.9,3.2,4.8,1.8,Iris-
versicolor\r\n6.1,2.8,4,1.3,Iris-versicolor\r\n6.3,2.5,4.9,1.5,Iris-
versicolor\r\n6.1,2.8,4.7,1.2,Iris-versicolor\r\n6.4,2.9,4.3,1.3,Iris-
versicolor\r\n6.6,3,4.4,1.4,Iris-versicolor\r\n6.8,2.8,4.8,1.4,Iris-
versicolor\r\n6.7,3,5,1.7,Iris-versicolor\r\n6,2.9,4.5,1.5,Iris-
versicolor\r\n5.7,2.6,3.5,1,Iris-versicolor\r\n5.5,2.4,3.8,1.1,Iris-
versicolor\r\n5.5,2.4,3.7,1,Iris-versicolor\r\n5.8,2.7,3.9,1.2,Iris-
versicolor\r\n6,2.7,5.1,1.6,Iris-versicolor\r\n5.4,3,4.5,1.5,Iris-
versicolor\r\n6,3.4,4.5,1.6,Iris-versicolor\r\n6.7,3.1,4.7,1.5,Iris-
versicolor\r\n6.3,2.3,4.4,1.3,Iris-versicolor\r\n5.6,3,4.1,1.3,Iris-
versicolor\r\n5.5,2.5,4,1.3,Iris-versicolor\r\n5.5,2.6,4.4,1.2,Iris-
versicolor\r\n6.1,3,4.6,1.4,Iris-versicolor\r\n5.8,2.6,4,1.2,Iris-
versicolor\r\n5,2.3,3.3,1,Iris-versicolor\r\n5.6,2.7,4.2,1.3,Iris-
versicolor\r\n5.7,3,4.2,1.2,Iris-versicolor\r\n5.7,2.9,4.2,1.3,Iris-
versicolor\r\n6.2,2.9,4.3,1.3,Iris-versicolor\r\n5.1,2.5,3,1.1,Iris-
versicolor\r\n5.7,2.8,4.1,1.3,Iris-versicolor\r\n6.3,3.3,6,2.5,Iris-
virginica\r\n5.8,2.7,5.1,1.9,Iris-virginica\r\n7.1,3,5.9,2.1,Iris-
virginica\r\n6.3,2.9,5.6,1.8,Iris-virginica\r\n6.5,3,5.8,2.2,Iris-
virginica\r\n7.6,3,6.6,2.1,Iris-virginica\r\n4.9,2.5,4.5,1.7,Iris-
virginica\r\n7.3,2.9,6.3,1.8,Iris-virginica\r\n6.7,2.5,5.8,1.8,Iris-
virginica\r\n7.2,3.6,6.1,2.5,Iris-virginica\r\n6.5,3.2,5.1,2,Iris-
virginica\r\n6.4,2.7,5.3,1.9,Iris-virginica\r\n6.8,3,5.5,2.1,Iris-
virginica\r\n5.7,2.5,5,2,Iris-virginica\r\n5.8,2.8,5.1,2.4,Iris-
virginica\r\n6.4,3.2,5.3,2.3,Iris-virginica\r\n6.5,3,5.5,1.8,Iris-
virginica\r\n7.7,3.8,6.7,2.2,Iris-virginica\r\n7.7,2.6,6.9,2.3,Iris-
virginica\r\n6,2.2,5,1.5,Iris-virginica\r\n6.9,3.2,5.7,2.3,Iris-
virginica\r\n5.6,2.8,4.9,2,Iris-virginica\r\n7.7,2.8,6.7,2,Iris-
virginica\r\n6.3,2.7,4.9,1.8,Iris-virginica\r\n6.7,3.3,5.7,2.1,Iris-

```
virginica\r\n7.2,3.2,6,1.8,Iris-virginica\r\n6.2,2.8,4.8,1.8,Iris-
virginica\r\n6.1,3,4.9,1.8,Iris-virginica\r\n6.4,2.8,5.6,2.1,Iris-
virginica\r\n7.2,3,5.8,1.6,Iris-virginica\r\n7.4,2.8,6.1,1.9,Iris-
virginica\r\n7.9,3.8,6.4,2,Iris-virginica\r\n6.4,2.8,5.6,2.2,Iris-
virginica\r\n6.3,2.8,5.1,1.5,Iris-virginica\r\n6.1,2.6,5.6,1.4,Iris-
virginica\r\n7.7,3,6.1,2.3,Iris-virginica\r\n6.3,3.4,5.6,2.4,Iris-
virginica\r\n6.4,3.1,5.5,1.8,Iris-virginica\r\n6,3,4.8,1.8,Iris-
virginica\r\n6.9,3.1,5.4,2.1,Iris-virginica\r\n6.7,3.1,5.6,2.4,Iris-
virginica\r\n6.9,3.1,5.1,2.3,Iris-virginica\r\n5.8,2.7,5.1,1.9,Iris-
virginica\r\n6.8,3.2,5.9,2.3,Iris-virginica\r\n6.7,3.3,5.7,2.5,Iris-
virginica\r\n6.7,3,5.2,2.3,Iris-virginica\r\n6.3,2.5,5,1.9,Iris-
virginica\r\n6.5,3,5.2,2,Iris-virginica\r\n6.2,3.4,5.4,2.3,Iris-
virginica\r\n5.9,3,5.1,1.8,Iris-virginica\r\n'}
```

[3]:
```python
df = pd.read_csv("IRIS.csv")
```

[5]:
```python
#data preprocessing
print(df.head())
print(df.tail())
print(df.info())
print(df.describe())
print(df.shape)
print(df.size)
print(df.ndim)
print(df.columns)

#check for null values
print(df.isnull())
print(df.isna())
print(df.isna().sum())
print(df.isnull().sum())
```

```
   sepal_length  sepal_width  petal_length  petal_width      species
0           5.1          3.5           1.4          0.2  Iris-setosa
1           4.9          3.0           1.4          0.2  Iris-setosa
2           4.7          3.2           1.3          0.2  Iris-setosa
3           4.6          3.1           1.5          0.2  Iris-setosa
4           5.0          3.6           1.4          0.2  Iris-setosa
     sepal_length  sepal_width  petal_length  petal_width         species
145           6.7          3.0           5.2          2.3  Iris-virginica
146           6.3          2.5           5.0          1.9  Iris-virginica
147           6.5          3.0           5.2          2.0  Iris-virginica
148           6.2          3.4           5.4          2.3  Iris-virginica
149           5.9          3.0           5.1          1.8  Iris-virginica
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
```

```
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
None
       sepal_length  sepal_width  petal_length  petal_width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.054000      3.758667     1.198667
std        0.828066     0.433594      1.764420     0.763161
min        4.300000     2.000000      1.000000     0.100000
25%        5.100000     2.800000      1.600000     0.300000
50%        5.800000     3.000000      4.350000     1.300000
75%        6.400000     3.300000      5.100000     1.800000
max        7.900000     4.400000      6.900000     2.500000
(150, 5)
750
2
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
     sepal_length  sepal_width  petal_length  petal_width  species
0           False        False         False        False    False
1           False        False         False        False    False
2           False        False         False        False    False
3           False        False         False        False    False
4           False        False         False        False    False
..            …          …           …          …       …
145         False        False         False        False    False
146         False        False         False        False    False
147         False        False         False        False    False
148         False        False         False        False    False
149         False        False         False        False    False

[150 rows x 5 columns]
     sepal_length  sepal_width  petal_length  petal_width  species
0           False        False         False        False    False
1           False        False         False        False    False
2           False        False         False        False    False
3           False        False         False        False    False
4           False        False         False        False    False
..            …          …           …          …       …
145         False        False         False        False    False
146         False        False         False        False    False
```

```
147         False        False         False        False     False
148         False        False         False        False     False
149         False        False         False        False     False

[150 rows x 5 columns]
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

[7]:
```python
# Selecting features and target
X = df.iloc[:, :-1].values  # All columns except the last one
y = df.iloc[:, -1].values   # Last column (encoded species)

# Splitting dataset (80% training, 20% testing)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
```

[8]:
```python
# Train Gaussian Naïve Bayes classifier
model = GaussianNB()
model.fit(X_train, y_train)

# Predict on test data
y_pred = model.predict(X_test)
```

[9]:
```python
# Compute accuracy
accuracy = accuracy_score(y_test, y_pred)

# Compute confusion matrix
cm = confusion_matrix(y_test, y_pred)

# Compute precision, recall, F1-score
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')

# Compute error rate
error_rate = 1 - accuracy
```

[10]:
```
print(f"\nAccuracy: {accuracy:.4f}")

print("\nConfusion Matrix:")
print(cm)

print("\nClassification Report:")
print(classification_report(y_test, y_pred))

print(f"\nPrecision: {precision:.4f}")
print(f"\nRecall: {recall:.4f}")
print(f"\nError Rate: {error_rate:.4f}")
print(f"\nF1 Score: {f1:.4f}")
```

```
Accuracy: 1.0000

Confusion Matrix:
[[10  0  0]
 [ 0  9  0]
 [ 0  0 11]]

Classification Report:
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        10
Iris-versicolor       1.00      1.00      1.00         9
 Iris-virginica       1.00      1.00      1.00        11

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


Precision: 1.0000

Recall: 1.0000

Error Rate: 0.0000

F1 Score: 1.0000
```