

Name : Shivam Indrabhan Borse

Roll No : 21119

Subject: Software Laboratory III (DATA SCIENCE)

Assignment No : 06

Problem statement:

Data Analytics III

1. 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.

CODE :

----- Assignment No : 06 -----

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.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

✓ 2.0s

DE ▶ ▢ ... 🗑

```
iris_df = pd.read_csv('iris.csv')
iris_df
```

✓ 0.0s

	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

```
iris_df.shape
```

[3] ✓ 0.0s

... (150, 6)

```
X = iris_df.iloc[:, :4].values
Y = iris_df['Species'].values
```

[4] ✓ 0.0s

```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=8)
```

[5] ✓ 0.0s

```
classifier = GaussianNB()
classifier.fit(X_train, y_train)
```

[6] ✓ 0.0s

... GaussianNB()

```
y_pred = classifier.predict(X_test)
y_pred
```

[7] ✓ 0.0s

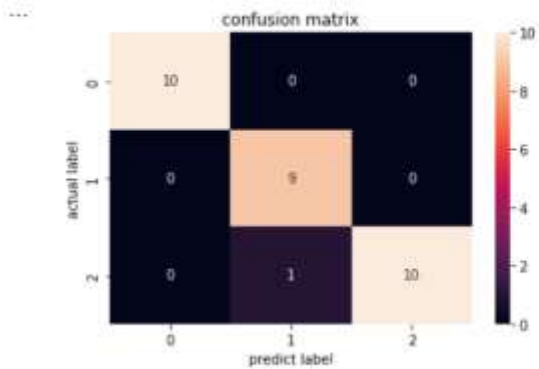
... array(['Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-virginica',
 'Iris-versicolor', 'Iris-setosa', 'Iris-setosa', 'Iris-virginica',
 'Iris-virginica', 'Iris-versicolor', 'Iris-versicolor',
 'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor',
 'Iris-virginica', 'Iris-virginica', 'Iris-virginica',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

+ Code

+ Markdown

```
cm = confusion_matrix(y_test,y_pred)
sns.heatmap(cm,annot=True)
plt.xlabel('predict label')
plt.ylabel('actual label')
plt.title('confusion matrix')
plt.show()
print(cm)
```

[8] ✓ 0.2s



```
[[10  0  0]
 [ 0  9  0]
 [ 0  1 10]]
```

```
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
```

[9] ✓ 0.0s

```
result = confusion_matrix(y_test , y_pred)
print('confusion_matrix : ',result)
```

[10] ✓ 0.0s

... confusion_matrix : [[10 0 0]
[0 9 0]
[0 1 10]]

```
print('accuracy score : ',accuracy_score(y_test,y_pred))
```

[11] ✓ 0.0s

... accuracy score : 0.9666666666666667

```
print('accuracy score : ',accuracy_score(y_test,y_pred))
```

```
[11] ✓ 0.0s
```

```
... accuracy score : 0.9666666666666667
```

```
print('classification_report : ',classification_report(y_test,y_pred))
```

```
[12] ✓ 0.0s
```

```
... classification_report :
```

			precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	1.00	1.00	10
Iris-versicolor	0.90	1.00	0.95	0.95	0.95	9
Iris-virginica	1.00	0.91	0.95	0.95	0.95	11
accuracy			0.97	0.97	0.97	30
macro avg	0.97	0.97	0.97	0.97	0.97	30
weighted avg	0.97	0.97	0.97	0.97	0.97	30

--> Shivam Borse