

```
In [1]: import numpy as np
import pandas as pd
import matplotlib as plt
```

```
In [2]: df=pd.read_csv("/home/ubuntu/DSBDAL/Iris.csv")
```

```
In [3]: df.head()
```

```
In [4]: import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import seaborn as sn
```

```
In [5]: df['Species'].unique()
```

```
In [6]:

df.describe(include='all')
```

```
In [7]: df.info()
```

```
In [8]: df.drop(columns="Id",inplace=True)
```

```
In [9]: df.isnull().sum()
```

```
In [10]: from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score ,precision_score,recall_score
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
```

```
In [11]:

X=df.iloc[:,0:4].values
y=df.iloc[:,4].values
```

```
In [12]: X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,rand
```

```
In [13]: from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
```

```
In [14]: gaussian = GaussianNB()
gaussian.fit(X_train, y_train)
Y_pred = gaussian.predict(X_test)
accuracy_nb=round(accuracy_score(y_test,Y_pred)* 100, 2)
acc_gaussian = round(gaussian.score(X_train, y_train) * 100, 2)
```

```
In [15]: cm = confusion_matrix(y_test, Y_pred)
```

```
In [16]: print(cm
)
```

```
In [17]: accuracy = accuracy_score(y_test,Y_pred)
precision =precision_score(y_test, Y_pred,average='micro')
recall = recall_score(y_test, Y_pred,average='micro')
f1 = f1_score(y_test,Y_pred,average='micro')
print('accuracy Naive Bayes: %.3f' %accuracy)
print('precision Naive Bayes: %.3f' %precision)
print('recall_Naive Bayes: %.3f' %recall)
print('f1-score_Naive Bayes : %.3f' %f1)
```

```
In [18]: confusion_matrix = pd.crosstab(y_test, Y_pred, rownames=['Actual'], c
sn.heatmap(confusion_matrix, annot=True)
```

```
In [19]: from sklearn.metrics import accuracy_score
accuracy =accuracy_score(y_test, Y_pred)
accuracy_percentage = 100 * accuracy
```

```
In [20]: print(accuracy)
```

```
In [21]: print(accuracy_percentage)
```

```
In [22]: from sklearn.metrics import classification_report
print(classification_report(y_test,Y_pred))
```

```
In [ ]:
```