

2_NB_Classifier_Iris_2Classes

```
[49]: #Import scikit-learn dataset library
import numpy as np
from sklearn import datasets
from sklearn.naive_bayes import GaussianNB

#Load dataset
iris = datasets.load_iris()
```

```
[50]: # print the names of the 13 features
print("Features: ", iris.feature_names)

# print the label type of wine(class_0, class_1, class_2)
print("Labels: ", iris.target_names)

# print data(feature)shape
print("\nData shape: ",iris.data.shape)
#print data(target)shape
print("\nTraget shape: ",iris.target.shape)

#print("\nData: ",iris.data)
#print("\nTarget: ",iris.target)

print("\nData type: ",type(iris.data))

newdata = iris.data[50:,:]
newtarget = iris.target[50:]

# print data(feature)shape
print("\nNew Data shape: ",newdata.shape)
#print data(target)shape
print("\nNew Traget shape: ",newtarget.shape)
```

Features: ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']

Labels: ['setosa' 'versicolor' 'virginica']

Data shape: (150, 4)

Traget shape: (150,)

Data type: <class 'numpy.ndarray'>

New Data shape: (100, 4)

New Traget shape: (100,)

```
[51]: #import the necessary module
      from sklearn.model_selection import train_test_split

      #split data set into train and test sets
      data_train, data_test, target_train, target_test = train_test_split(newdata,
                                                                           newtarget, test_size = 0.30, random_state = 5)
```

```
[52]: import numpy as np
      gnb = GaussianNB()

      #Train the model using the training sets
      gnb.fit(data_train, target_train)

      #Predict the response for test dataset
      target_pred = gnb.predict(data_test)
```

```
[53]: #Import scikit-learn metrics module for accuracy calculation
      from sklearn import metrics

      # Model Accuracy, how often is the classifier correct?
      print("Accuracy:",metrics.accuracy_score(target_test, target_pred))
```

Accuracy: 0.9

```
[54]: #Import confusion_matrix from scikit-learn metrics module for confusion_matrix
      from sklearn.metrics import confusion_matrix
      confusion_matrix(target_test, target_pred)
```

```
[54]: array([[16,  1],
             [ 2, 11]])
```

```
[55]: from sklearn.metrics import precision_score
      from sklearn.metrics import recall_score

      precision = precision_score(target_test, target_pred)
      recall = recall_score(target_test, target_pred)

      print('precision: {}'.format(precision))
      print('recall: {}'.format(recall))
```

precision: 0.8888888888888888

recall: 0.9411764705882353

[55]: