3_NB_Classifier_Iris_3Classes

December 19, 2021

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

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

[ ]:
```
# 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
iris.data.shape
```

```
Features:  ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal
width (cm)']
Labels:  ['setosa' 'versicolor' 'virginica']
```

[ ]:
```
(150, 4)
```

[ ]:
```
#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(iris.data,
                        iris.target, test_size = 0.30, random_state = 10)
```

[ ]:
```
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)
```

[ ]:
```
#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: 1.0
```

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

[ ]:
```
array([[14,  0,  0],
       [ 0, 17,  0],
       [ 0,  0, 14]])
```

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

precision = precision_score(target_test, target_pred, average=None)
recall = recall_score(target_test, target_pred, average=None)

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

```
precision: [1. 1. 1.]
recall: [1. 1. 1.]
```

Exercise:

1) 1<=Rollnumber<=25: #Task 1: Try the algo on Dataset1 - OneHotEncoding of features: and Train test Division 70%-30%

#Task 2: Apply algorithm on digits dataset - LabelEncoding of features: and Train test Division 80%-20%

2) 26<=Rollnumber<=50: #Task 1: Try the algo on Dataset1 - LabelEncoding of features: and Train test Division 80%-20%

#Task 2: Apply algorithm on breast cancer wisconsin dataset - One Hot Encoding of features: and Train test Division 60%-40%

3) 51<=Rollnumber<=75: #Task 1: Try the algo on Dataset2 - LabelEncoding of features: and Train test Division 90%-10%

#Task 2: Apply algorithm on digits dataset - One Hot Encoding of features: and Train test Division 65%-35%

4) 76<=Rollnumber<=100: #Task 1: Try the algo on Dataset2 - OneHotEncoding of features: and Train test Division 75%-25% #Task 2: Apply algorithm on wine dataset - LabelEncoding of features: and Train test Division 80%-20%

5) 101<=Rollnumber<=125: #Task 1: Try the algo on Dataset3 - OneHotEncoding of features:and Train test Division 85%-15% #Task 2: Apply algorithm on wine dataset - LabelEncoding of features: and Train test Division 66%-34%

6) 126<=Rollnumber + All with No RollNumbers: #Task 1: Try the algo on Dataset3 - LabelEncoding of features:and Train test Division 95%-5%

#Task 2: Apply algorithm on breast cancer wisconsin dataset - One Hot Encoding of features: and Train test Division 50%-50%

Instruction for Task-1 & 2:

i) Set Random state of model equals to your roll number (or last 2 digit of your id -if you don't have roll number)

Questions: For Task - 1

(1) What will be the value of Play, if Outlook is 'Rainy', Temperature is 'Mild', Humidity ='Normal', and Wind = 'False'?

(2) What will be the value of Play, if Outlook is 'Sunny', Temeprature is 'Cool', Humidity ='High', and Wind = 'True'?

(3) Accuracy , precision and recall of both Models?

[ ]: