assignment7

May 2, 2024

0.1 NAIVE BAYES TEST

- Importing the libraries

[22]:
```
import pandas as pd
import numpy as np
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
from sklearn.metrics import accuracy_score, confusion_matrix, recall_score,␣
↪precision_score
```

- **Reading the dataset**

[10]:
```
df = pd.read_csv("Iris.csv")
df
```

[10]:
```
      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  \
0      1            5.1           3.5            1.4           0.2
1      2            4.9           3.0            1.4           0.2
2      3            4.7           3.2            1.3           0.2
3      4            4.6           3.1            1.5           0.2
4      5            5.0           3.6            1.4           0.2
..   …              …             …              …             …
145  146            6.7           3.0            5.2           2.3
146  147            6.3           2.5            5.0           1.9
147  148            6.5           3.0            5.2           2.0
148  149            6.2           3.4            5.4           2.3
149  150            5.9           3.0            5.1           1.8

            Species
0       Iris-setosa
1       Iris-setosa
2       Iris-setosa
3       Iris-setosa
4       Iris-setosa
..              …
145  Iris-virginica
146  Iris-virginica
```

```
147  Iris-virginica
148  Iris-virginica
149  Iris-virginica

[150 rows x 6 columns]
```

- **Splitting the dataset int Dependent and Independent Varaibles**

[23]:
```
df_x = df.iloc[:,1:5]
df_x
```

[23]:
```
     SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
0              5.1           3.5            1.4           0.2
1              4.9           3.0            1.4           0.2
2              4.7           3.2            1.3           0.2
3              4.6           3.1            1.5           0.2
4              5.0           3.6            1.4           0.2
..             …             …              …             …
145            6.7           3.0            5.2           2.3
146            6.3           2.5            5.0           1.9
147            6.5           3.0            5.2           2.0
148            6.2           3.4            5.4           2.3
149            5.9           3.0            5.1           1.8

[150 rows x 4 columns]
```

[24]:
```
df_y = df.iloc[:, 5]
df_y
```

[24]:
```
0         Iris-setosa
1         Iris-setosa
2         Iris-setosa
3         Iris-setosa
4         Iris-setosa
             …
145    Iris-virginica
146    Iris-virginica
147    Iris-virginica
148    Iris-virginica
149    Iris-virginica
Name: Species, Length: 150, dtype: object
```

- **Splitting the dataset into the Training set and Test set**

[12]:
```
X_train, X_test, y_train, y_test = train_test_split(df_x, df_y, test_size=0.2,␣
↪random_state=2)
```

- Feature Scaling

[13]:
```
sc = preprocessing.StandardScaler()
x_train = sc.fit_transform(X_train)
```

- **Applying Gaussian Naive Bayes**

[14]:
```
GNB = GaussianNB()
```

[15]:
```
GNB.fit(x_train, y_train)
```

[15]:
```
GaussianNB()
```

[16]:
```
y_pred = GNB.predict(X_test)
```

```
c:\Users\Mrudul Patel\anaconda3\lib\site-packages\sklearn\base.py:443:
UserWarning: X has feature names, but GaussianNB was fitted without feature
names
  warnings.warn(
```

[17]:
```
y_pred
```

[17]:
```
array(['Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica'], dtype='<U15')
```

- **Confusion Matrix, Accuracy Score, Error, Precision Score, Recall Score**

[18]:
```
cm = confusion_matrix(y_test, y_pred)
```

[19]:
```
ac_score = accuracy_score(y_test, y_pred)
```

[20]:
```
error = 1 - ac_score
```

[44]:
```
prec_score = precision_score(y_test, y_pred, average=None)
```

```
c:\Users\Mrudul Patel\anaconda3\lib\site-
packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning:
Precision is ill-defined and being set to 0.0 in labels with no predicted
samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

[45]:
```
rec_score = recall_score(y_test, y_pred, average=None)
```

[42]:
```
print("Confusion Matrix: ", cm)
print("Accuracy Score: ", ac_score)
print("Error: ", error)
print("Precision Score: ", prec_score)
print("Recall Score: ", rec_score)
```

```
Confusion Matrix:  [[ 0  0 14]
 [ 0  0  8]
 [ 0  0  8]]
Accuracy Score:  0.26666666666666666
Error:  0.7333333333333334
Precision Score:  [0.         0.         0.26666667]
Recall Score:  [0. 0. 1.]
```

[34]:
```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
                 precision    recall  f1-score   support

    Iris-setosa       0.00      0.00      0.00        14
Iris-versicolor       0.00      0.00      0.00         8
 Iris-virginica       0.27      1.00      0.42         8

       accuracy                           0.27        30
      macro avg       0.09      0.33      0.14        30
   weighted avg       0.07      0.27      0.11        30
```

```
c:\Users\Mrudul Patel\anaconda3\lib\site-
packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
c:\Users\Mrudul Patel\anaconda3\lib\site-
packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
c:\Users\Mrudul Patel\anaconda3\lib\site-
packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning:
Precision and F-score are ill-defined and being set to 0.0 in labels with no
predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

[ ]: