exp-6-diabetes

March 27, 2025

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv("C:\\Users\\Welcome\\Downloads\\diabetes.csv")
data.head(4)
Pregnancies        Glucose        BloodPressure        SkinThickness        Insulin        BMI
0        6        148        72        35        0        33.6        0.
 ↪627        50        1
1        1        85        66        29        0        26.6        0.
 ↪351        31        0
2        8        183        64        0        0        23.3        0.
 ↪672        32        1
3        1        89        66        23        94        28.1        0.
 ↪167        21        0
#dividing the data in
y=data['Outcome']
x=data[['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','BMI','DiabetesPedig
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size =0.
 ↪2,random_state = 0)
print("data splited")
data splited
from sklearn.naive_bayes import GaussianNB
gaussian = GaussianNB()
gaussian.fit(xtrain, ytrain)
GaussianNB()
In a Jupyter environment, please rerun this cell to show the HTML␣
 ↪representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this␣
 ↪page with nbviewer.org.
Y_pred = gaussian.predict(xtest)
xtest
Pregnancies        Glucose        BloodPressure        SkinThickness        Insulin        BMI
661        1        199        76        43        0        42.9        1.
 ↪394        22
122        2        107        74        30        100        33.6        0.
 ↪404        23
```

```
113         4         76         62         0         0         34.0         0.
 ↪391         25
14         5         166         72         19         175         25.8         0.
 ↪587         51
529         0         111         65         0         0         24.6         0.
 ↪660         31
...         ...         ...         ...         ...         ...         ...         ...
 ↪         ...
476         2         105         80         45         191         33.7         0.
 ↪711         29
482         4         85         58         22         49         27.8         0.
 ↪306         28
230         4         142         86         0         0         44.0         0.
 ↪645         22
527         3         116         74         15         105         26.3         0.
 ↪107         24
380         1         107         72         30         82         30.8         0.
 ↪821         24
154 rows × 8 columns

Y_pred
array([1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1,
       1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1,
       1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      dtype=int64)
from sklearn.metrics import␣
 ↪precision_score,confusion_matrix,accuracy_score,recall_score
cm = confusion_matrix(ytest, Y_pred)
cm
array([[93, 14],
       [18, 29]], dtype=int64)
accuracy = accuracy_score(ytest,Y_pred)
precision = precision_score(ytest, Y_pred,average='micro')
recall = recall_score(ytest, Y_pred,average='micro')
print("accuracy is",accuracy)
print("precision is ",precision)
print("recall is ",recall)
accuracy is 0.7922077922077922
precision is  0.7922077922077922
recall is  0.7922077922077922
```

```
  Cell In[1], line 38
```

```
    154 rows × 8 columns
            ^
SyntaxError: invalid character '×' (U+00D7)
```

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv("C:\\Users\\Welcome\\Downloads\\diabetes.csv")
data.head(4)
Pregnancies        Glucose        BloodPressure        SkinThickness        Insulin        BMI
0        6        148        72        35        0        33.6        0.
 ↪627        50        1
1        1        85        66        29        0        26.6        0.
 ↪351        31        0
2        8        183        64        0        0        23.3        0.
 ↪672        32        1
3        1        89        66        23        94        28.1        0.
 ↪167        21        0
#dividing the data in
y=data['Outcome']
x=data[['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','BMI','DiabetesPedig
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size =0.
 ↪2,random_state = 0)
print("data splited")
data splited
from sklearn.naive_bayes import GaussianNB
gaussian = GaussianNB()
gaussian.fit(xtrain, ytrain)
GaussianNB()
In a Jupyter environment, please rerun this cell to show the HTML␣
 ↪representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this␣
 ↪page with nbviewer.org.
Y_pred = gaussian.predict(xtest)
xtest
Pregnancies        Glucose        BloodPressure        SkinThickness        Insulin        BMI
661        1        199        76        43        0        42.9        1.
 ↪394        22
122        2        107        74        30        100        33.6        0.
 ↪404        23
113        4        76        62        0        0        34.0        0.
 ↪391        25
14        5        166        72        19        175        25.8        0.
 ↪587        51
```

```
529        0        111        65        0        0        24.6        0.
 ↪660        31
...        ...        ...        ...        ...        ...        ...        ...
 ↪        ...
476        2        105        80        45        191        33.7        0.
 ↪711        29
482        4        85        58        22        49        27.8        0.
 ↪306        28
230        4        142        86        0        0        44.0        0.
 ↪645        22
527        3        116        74        15        105        26.3        0.
 ↪107        24
380        1        107        72        30        82        30.8        0.
 ↪821        24
154 rows × 8 columns

Y_pred
array([1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1,
       1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1,
       1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
      dtype=int64)
from sklearn.metrics import␣
 ↪precision_score,confusion_matrix,accuracy_score,recall_score
cm = confusion_matrix(ytest, Y_pred)
cm
array([[93, 14],
       [18, 29]], dtype=int64)
accuracy = accuracy_score(ytest,Y_pred)
precision = precision_score(ytest, Y_pred,average='micro')
recall = recall_score(ytest, Y_pred,average='micro')
print("accuracy is",accuracy)
print("precision is ",precision)
print("recall is ",recall)
accuracy is 0.7922077922077922
precision is  0.7922077922077922
recall is  0.7922077922077922
```

```
  Cell In[1], line 38
    154 rows × 8 columns
              ^
SyntaxError: invalid character '×' (U+00D7)
```

[ ]: