bus0gt4m5

April 29, 2024

[ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
import os
import scipy as sp
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
data = pd.read_csv("/content/diabetes.csv")
data.head()
```

[ ]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
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
4	0	137	40	35	168	43.1	2.288	33	1

[ ]:
```
data.describe()
```

[ ]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin
count	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479
std	3.369578	31.972618	19.355807	15.952218	115.244002
min	0.000000	0.000000	0.000000	0.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000
75%	6.000000	140.250000	80.000000	32.000000	127.250000
max	17.000000	199.000000	122.000000	99.000000	846.000000

```
              BMI  DiabetesPedigreeFunction         Age     Outcome
count  768.000000                768.000000  768.000000  768.000000
mean    31.992578                  0.471876   33.240885    0.348958
std      7.884160                  0.331329   11.760232    0.476951
min      0.000000                  0.078000   21.000000    0.000000
25%     27.300000                  0.243750   24.000000    0.000000
50%     32.000000                  0.372500   29.000000    0.000000
75%     36.600000                  0.626250   41.000000    1.000000
max     67.100000                  2.420000   81.000000    1.000000
```

[ ]:
```
data.shape
data.value_counts()
```

[ ]:
```
Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin  BMI
DiabetesPedigreeFunction  Age  Outcome
0            57       60             0              0        21.7  0.735
67   0          1
             67       76             0              0        45.3  0.194
46   0          1
5            103      108            37             0        39.2  0.305
65   0          1
             104      74             0              0        28.8  0.153
48   0          1
             105      72             29             325      36.9  0.159
28   0          1
                              ..
2            84       50             23             76       30.4  0.968
21   0          1
             85       65             0              0        39.6  0.930
27   0          1
             87       0              23             0        28.9  0.773
25   0          1
                      58             16             52       32.7  0.166
25   0          1
17           163      72             41             114      40.9  0.817
47   1          1
Name: count, Length: 768, dtype: int64
```

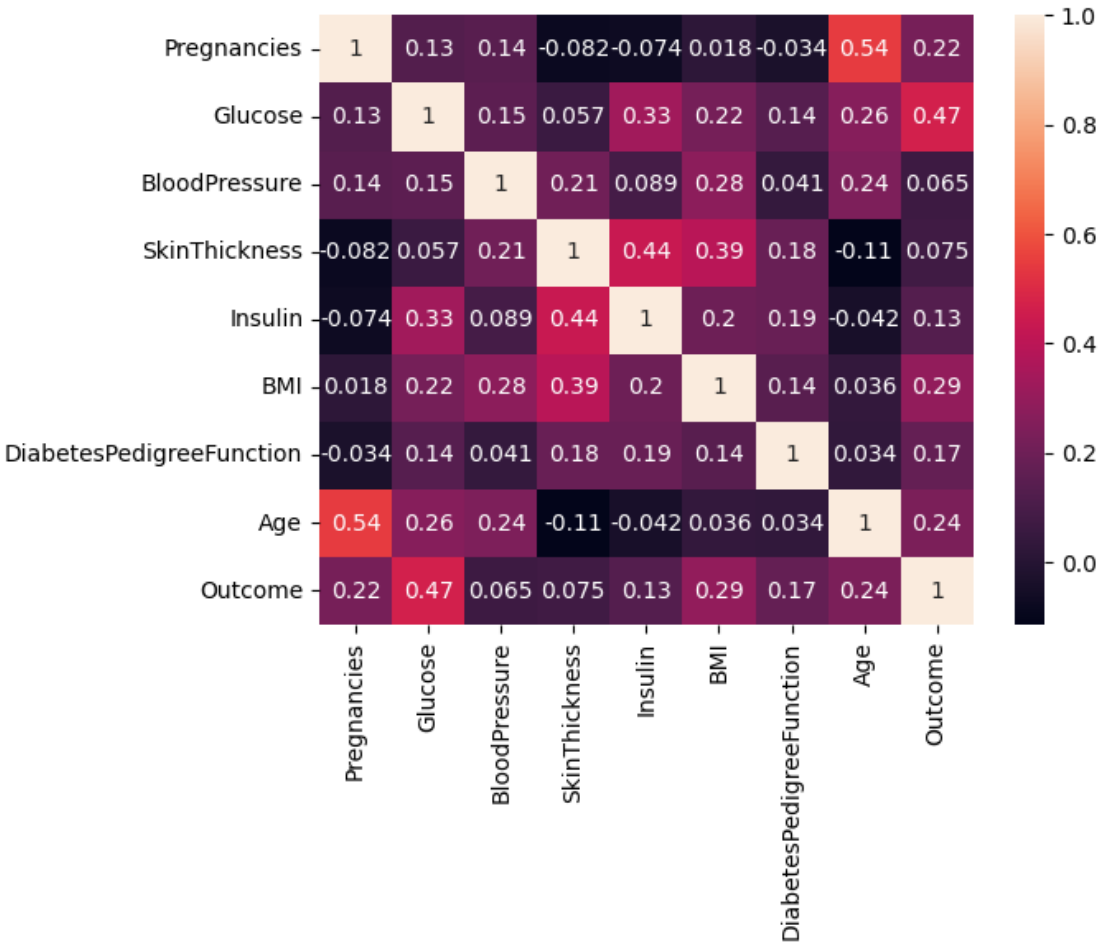
[ ]:
```
data.corr()
```

[ ]:
```
                          Pregnancies   Glucose  BloodPressure  SkinThickness  \
Pregnancies                  1.000000  0.129459       0.141282      -0.081672
Glucose                      0.129459  1.000000       0.152590       0.057328
BloodPressure                0.141282  0.152590       1.000000       0.207371
SkinThickness               -0.081672  0.057328       0.207371       1.000000
Insulin                     -0.073535  0.331357       0.088933       0.436783
```

```
BMI                         0.017683  0.221071       0.281805       0.392573
DiabetesPedigreeFunction   -0.033523  0.137337       0.041265       0.183928
Age                         0.544341  0.263514       0.239528      -0.113970
Outcome                     0.221898  0.466581       0.065068       0.074752

                           Insulin       BMI  DiabetesPedigreeFunction  \
Pregnancies              -0.073535  0.017683                 -0.033523
Glucose                   0.331357  0.221071                  0.137337
BloodPressure             0.088933  0.281805                  0.041265
SkinThickness             0.436783  0.392573                  0.183928
Insulin                   1.000000  0.197859                  0.185071
BMI                       0.197859  1.000000                  0.140647
DiabetesPedigreeFunction  0.185071  0.140647                  1.000000
Age                      -0.042163  0.036242                  0.033561
Outcome                   0.130548  0.292695                  0.173844

                               Age   Outcome
Pregnancies               0.544341  0.221898
Glucose                   0.263514  0.466581
BloodPressure             0.239528  0.065068
SkinThickness            -0.113970  0.074752
Insulin                  -0.042163  0.130548
BMI                       0.036242  0.292695
DiabetesPedigreeFunction  0.033561  0.173844
Age                       1.000000  0.238356
Outcome                   0.238356  1.000000
```

[ ]:
```
sns.heatmap(data.corr(), annot=True)
```

[ ]:
```
<Axes: >
```



[ ]:
```
x = data.drop(columns = 'Outcome')
y = data['Outcome']

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.
  ↪2,random_state=0)
```

#Gaussian Naive Bayes

[ ]:
```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(x_train,y_train)
```

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

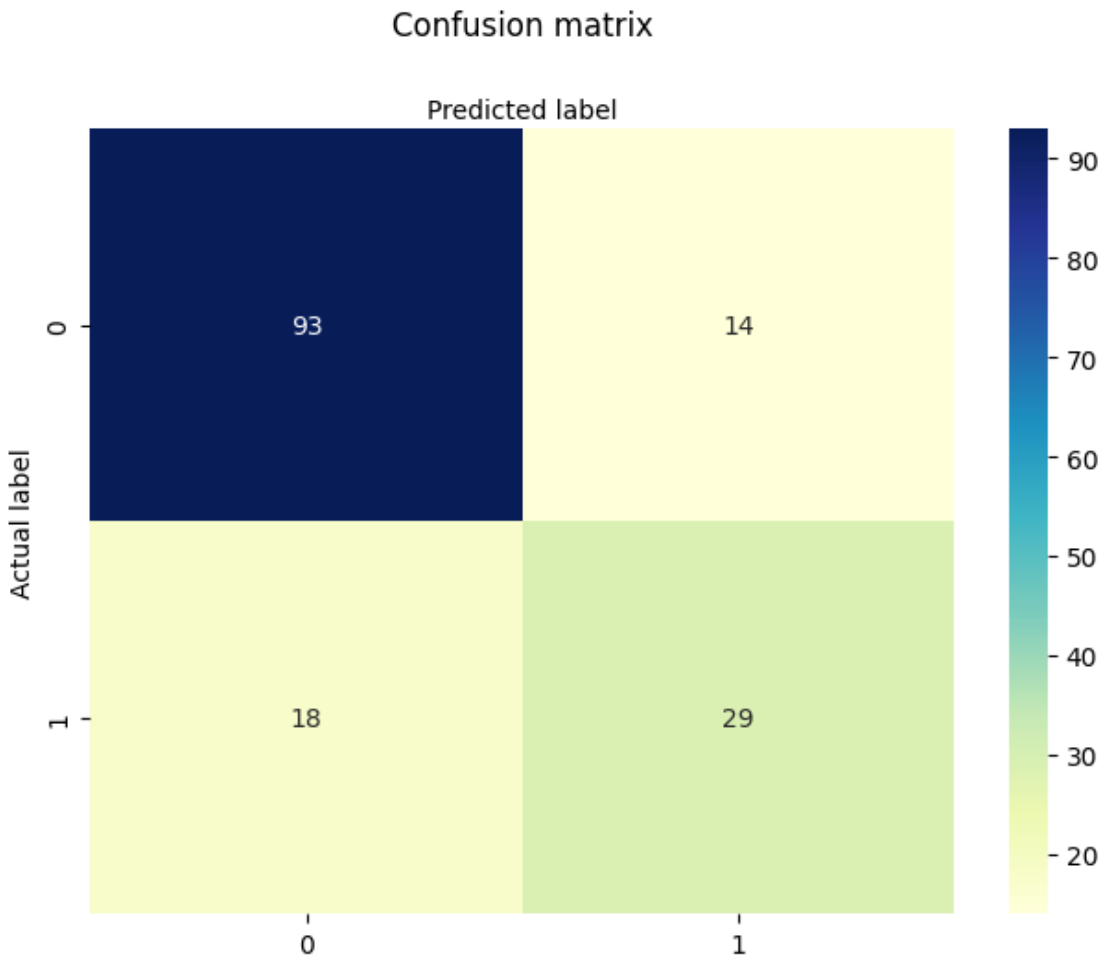
[ ]:
```
y_pred=gnb.predict(x_test)
from sklearn.metrics import␣
  ↪accuracy_score,classification_report,confusion_matrix
```

```
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
print("Classification Report is:\n",classification_report(y_test,y_pred))
print("Confusion Matrix:\n",confusion_matrix(y_test,y_pred))
print("Training Score:\n",gnb.score(x_train,y_train)*100)
print("Mean Squared Error:\n",mean_squared_error(y_test,y_pred))
print("R2 score is:\n",r2_score(y_test,y_pred))
```

```
Classification Report is:
              precision    recall  f1-score   support

           0       0.84      0.87      0.85       107
           1       0.67      0.62      0.64        47

    accuracy                           0.79       154
   macro avg       0.76      0.74      0.75       154
weighted avg       0.79      0.79      0.79       154

Confusion Matrix:
 [[93 14]
 [18 29]]
Training Score:
 75.7328990228013
Mean Squared Error:
 0.2077922077922078
R2 score is:
 0.020083515609465197
```

[ ]:
```
class_names=[0,1]
cnf_matrix=confusion_matrix(y_test,y_pred)
fig, ax = plt.subplots()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)
sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu" ,fmt='g')
ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```

[ ]: Text(0.5, 427.9555555555555, 'Predicted label')



[ ]:
```
print("Accuracy Score:\n",gnb.score(x_train,y_train)*100)
```

```
Accuracy Score:
 75.7328990228013
```

#Decision Tree Classifier

[ ]:
```
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier(max_depth=6,␣
 ↪random_state=123,criterion='entropy')

dtree.fit(x_train,y_train)
```

[ ]:
```
DecisionTreeClassifier(criterion='entropy', max_depth=6, random_state=123)
```

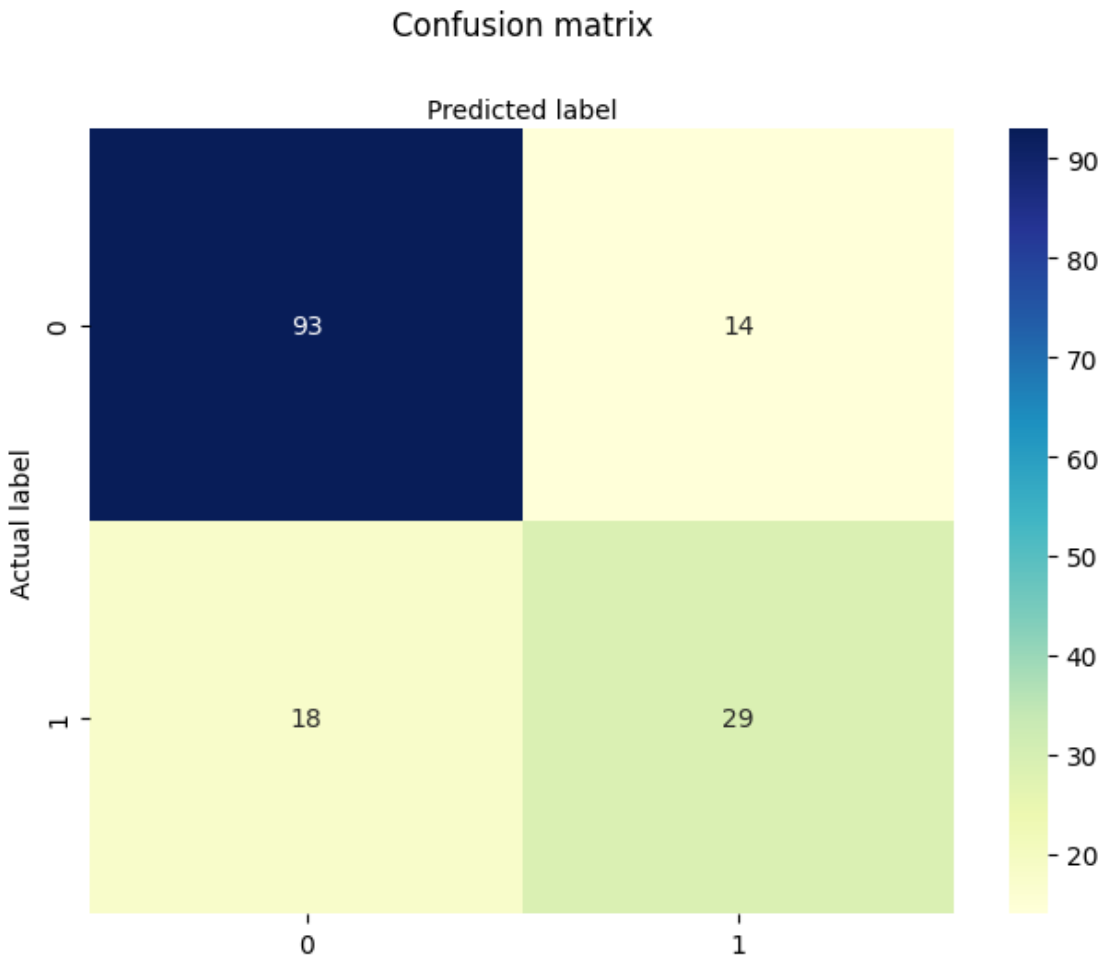
[ ]:
```
y2_pred=dtree.predict(x_test)
```

```
from sklearn.metrics import␣
 ↪accuracy_score,classification_report,confusion_matrix
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
print("Classification Report is:\n",classification_report(y_test,y_pred))
print("Confusion Matrix:\n",confusion_matrix(y_test,y_pred))
print("Training Score:\n",dtree.score(x_train,y_train)*100)
print("Mean Squared Error:\n",mean_squared_error(y_test,y_pred))
print("R2 score is:\n",r2_score(y_test,y_pred))
```

```
Classification Report is:
               precision    recall  f1-score   support

           0       0.84      0.87      0.85       107
           1       0.67      0.62      0.64        47

    accuracy                           0.79       154
   macro avg       0.76      0.74      0.75       154
weighted avg       0.79      0.79      0.79       154

Confusion Matrix:
 [[93 14]
 [18 29]]
Training Score:
 82.08469055374593
Mean Squared Error:
 0.2077922077922078
R2 score is:
 0.020083515609465197
```

[ ]:
```
class_names=[0,1]
cnf_matrix=confusion_matrix(y_test,y_pred)
fig, ax = plt.subplots()
tick_marks = np.arange(len(class_names))
plt.xticks(tick_marks, class_names)
plt.yticks(tick_marks, class_names)
sns.heatmap(pd.DataFrame(cnf_matrix), annot=True, cmap="YlGnBu" ,fmt='g')
ax.xaxis.set_label_position("top")
plt.tight_layout()
plt.title('Confusion matrix', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
```

[ ]:
```
Text(0.5, 427.9555555555555, 'Predicted label')
```



[ ]:

[ ]:
```
print(accuracy_score(y_test,y2_pred)*100)
```

```
73.37662337662337
```

#Regression Tree on Walmart Sales

[ ]:
```
from sklearn import metrics
```

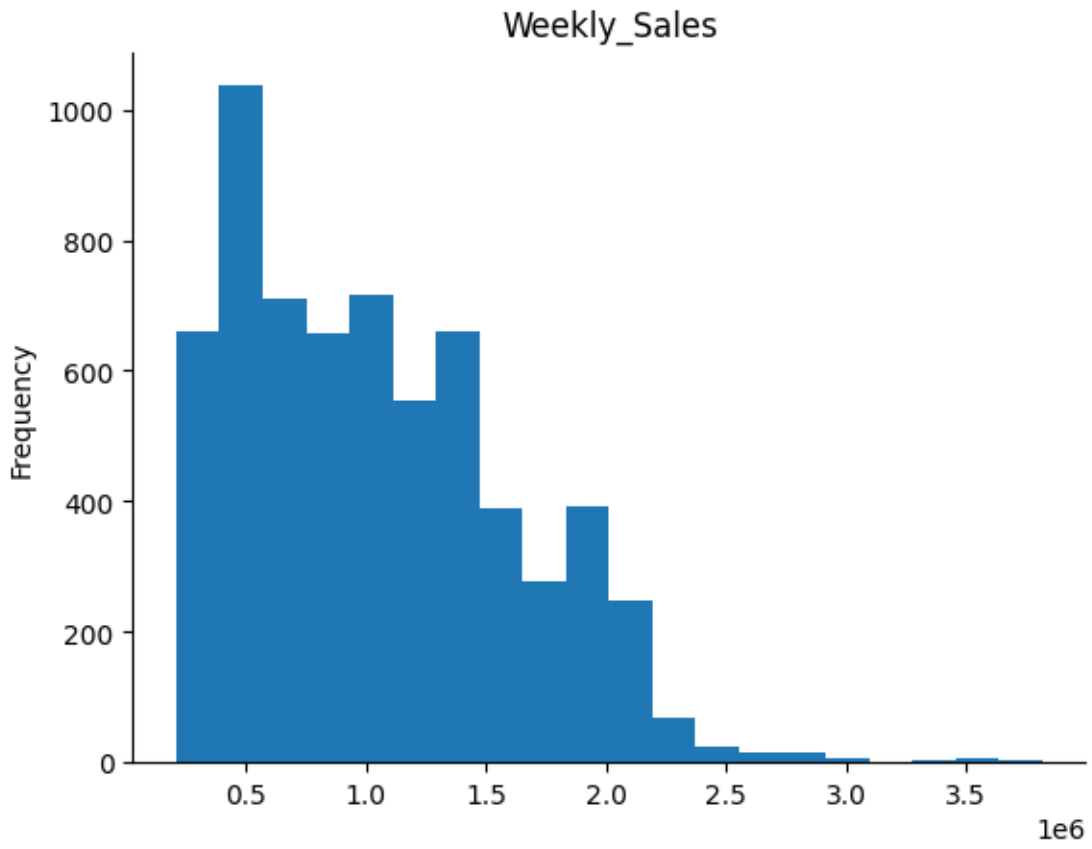
[ ]:
```
walmart = pd.read_csv('Walmart_sales.csv')

walmart.head()
```

[ ]:

	Store	Date	Weekly_Sales	Holiday_Flag	Temperature	Fuel_Price	\
0	1	05-02-2010	1643690.90	0	42.31	2.572	
1	1	12-02-2010	1641957.44	1	38.51	2.548	

```
2       1  19-02-2010    1611968.17             0        39.93       2.514
3       1  26-02-2010    1409727.59             0        46.63       2.561
4       1  05-03-2010    1554806.68             0        46.50       2.625

           CPI  Unemployment
0   211.096358         8.106
1   211.242170         8.106
2   211.289143         8.106
3   211.319643         8.106
4   211.350143         8.106
```

[ ]:
```
from matplotlib import pyplot as plt

walmart['Weekly_Sales'].plot(kind='hist', bins=20, title='Weekly_Sales')

plt.gca().spines[['top', 'right',]].set_visible(False)
```



[ ]:
```
walmart.shape
```

[ ]:
```
(6435, 8)
```

[ ]:
```
walmart.isna().sum()

walmart.drop('Date', axis=1, inplace=True)

X = walmart.drop('Weekly_Sales',axis=1)

y = walmart['Weekly_Sales']
```

[ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.
↪25,random_state=42)

from sklearn.tree import DecisionTreeRegressor


dtr = DecisionTreeRegressor()

dtr.fit(X_train,y_train)

y_pred = dtr.predict(X_test)
```

[ ]:
```
rmse = metrics.mean_squared_error(y_test, y_pred, squared=False)

print('RMSE: ', rmse)
```

```
RMSE:  203115.80854825137
```