lab3

January 20, 2025

[2]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

[1]:
```
import pandas as pd

df = pd.read_csv("./data.csv")
```

[33]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
scaler = StandardScaler()
```

[35]:
```
train, test = train_test_split(df, test_size=0.2)
```

[37]:
```
X_train = train[['Socioeconomic Score',
 'Study Hours',
 'Sleep Hours',
 'Attendance (%)']]

X_test = test[['Socioeconomic Score',
 'Study Hours',
 'Sleep Hours',
 'Attendance (%)']]
```

[38]:
```
y_train = train[['Grades']]
y_test = test[['Grades']]
```

[39]:
```
from sklearn.linear_model import LinearRegression
```

[40]:
```
regressor = LinearRegression(n_jobs=5)
```

[41]:
```
regressor.fit(X_train, y_train)
```

[41]:
```
LinearRegression(n_jobs=5)
```

[42]:
```
y_pred = regressor.predict(X_test)
```

[43]:

```
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
```

[44]:

```
mae = mean_absolute_error(y_pred=y_pred, y_true=y_test)
mse = mean_squared_error(y_pred=y_pred, y_true=y_test)
r2 = r2_score(y_pred=y_pred, y_true=y_test)
```

[45]:

```
print(f"R2 Score: {r2}")
print(f"MSE: {mse}")
print(f"MAE: {mae}")
```

```
R2 Score: 0.7928316268914254
MSE: 17.24643198486114
MAE: 3.2081125696592454
```

[46]:

```
y_test[:5], y_pred[:5]
```

[46]:

```
(      Grades
 602     43.0
 1196    34.0
 1226    35.0
 474     36.0
 1252    34.0,
 array([[40.57640149],
        [24.00012658],
        [34.21593047],
        [29.70645367],
        [31.41475296]]))
```

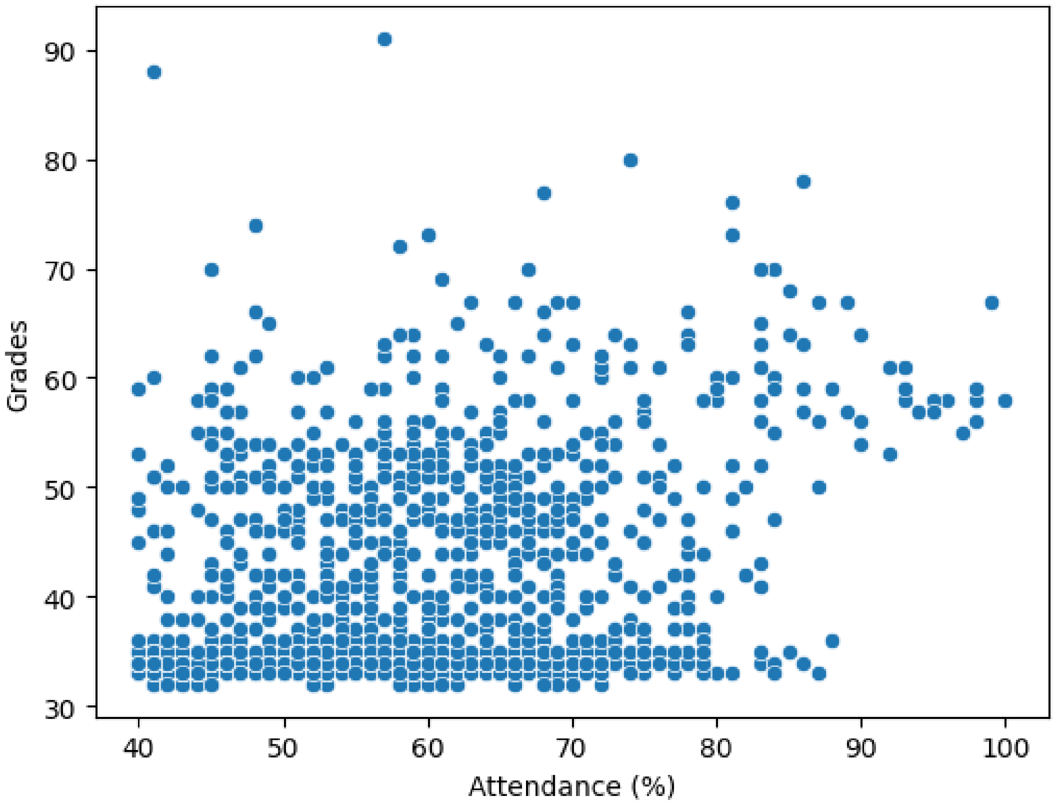
[47]:

```
print (df.columns)
```

```
Index(['Socioeconomic Score', 'Study Hours', 'Sleep Hours', 'Attendance (%)',
       'Grades'],
      dtype='object')
```

[48]:

```
sns.scatterplot(x=df['Attendance (%)'], y=df['Grades'])
plt.show()
```



Logistic Regression

[3]:

```python
df = pd.read_csv("./diabetes.csv")
```

[51]:

```python
train1, test1 = train_test_split(df, test_size=0.2)
```

[53]:

```python
X_train1 = train1[['Pregnancies',
 'Glucose',
 'BloodPressure',
 'SkinThickness',
 'Insulin',
 'BMI',
 'DiabetesPedigreeFunction',
 'Age']]

X_test1 = test1[['Pregnancies',
 'Glucose',
 'BloodPressure',
 'SkinThickness',
 'Insulin',
 'BMI',
 'DiabetesPedigreeFunction',
 'Age']]
```

[54]:

```
print(train1.columns)
```

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

[55]:

```
y_train1 = train1[['Outcome']]
y_test1 = test1[['Outcome']]
```

[56]:

```
y_train1 = y_train1.values.ravel()
y_test1 = y_test1.values.ravel()
```

[57]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,␣
 ↪confusion_matrix,classification_report
```

[58]:

```
model = LogisticRegression(n_jobs=5, max_iter=500)
model.fit(X_train1, y_train1)
```

[58]:

```
LogisticRegression(max_iter=500, n_jobs=5)
```

[59]:

```
y_pred1 = model.predict(X_test1)
accuracy = accuracy_score(y_test1, y_pred1)
print(f"Accuracy: {accuracy:.2f}")
```

```
Accuracy: 0.75
```

[60]:

```
cm = confusion_matrix(y_test1, y_pred1)
print("Confusion Matrix:")
print(cm)
```

```
Confusion Matrix:
[[87 12]
 [26 29]]
```

[61]:

```
report = classification_report(y_test1, y_pred1)
print("Classification Report:")
print(report)
```

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

           0       0.77      0.88      0.82        99
           1       0.71      0.53      0.60        55
```

```
    accuracy                           0.75       154
   macro avg       0.74      0.70      0.71       154
weighted avg       0.75      0.75      0.74       154
```

[62]:

```
y_test1[:10], y_pred1[:10]
```

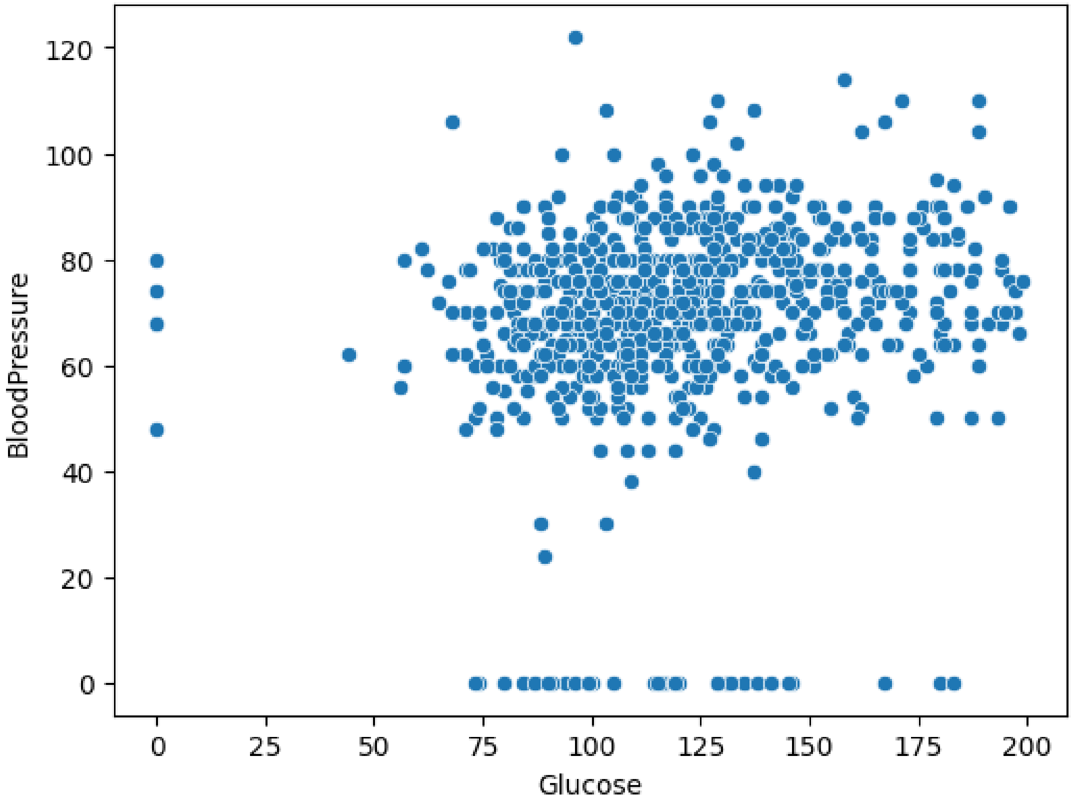
[62]:

```
(array([0, 0, 1, 0, 1, 0, 1, 0, 1, 0]), array([0, 0, 1, 0, 0, 0, 0, 0, 0, 0]))
```

[63]:

```
sns.scatterplot(x=df['Glucose'], y=df['BloodPressure'])

plt.show()
```


BloodPressure
Glucose