# Assignment

March 24, 2024

[1]:
```
import numpy as np
import pandas as pd
data=pd.read_csv('C:\\Users\\DELL PC\\Desktop\\LINEAR PROG\\Assighnment␣
 ↪_25-3-2024\\student_scores_dataset.csv')
data
```

[1]:
```
    Study Hours  Exam Scores
0           3.7         87.9
1           9.5        143.6
2           7.3        123.7
3           6.0         99.9
4           1.6         64.5
..          …           …
95          4.9         95.3
96          5.2        101.9
97          4.3         94.5
98          0.3         53.9
99          1.1         64.9

[100 rows x 2 columns]
```

[2]:
```
x = np.array(data['Study Hours']).reshape(-1,1)
y = np.array(data['Exam Scores'])
x
```

[2]:
```
array([[3.7],
       [9.5],
       [7.3],
       [6. ],
       [1.6],
       [1.6],
       [0.6],
       [8.7],
       [6. ],
       [7.1],
       [0.2],
       [9.7],
       [8.3],
```

```
[2.1],
[1.8],
[1.8],
[3. ],
[5.2],
[4.3],
[2.9],
[6.1],
[1.4],
[2.9],
[3.7],
[4.6],
[7.9],
[2. ],
[5.1],
[5.9],
[0.5],
[6.1],
[1.7],
[0.7],
[9.5],
[9.7],
[8.1],
[3. ],
[1. ],
[6.8],
[4.4],
[1.2],
[5. ],
[0.3],
[9.1],
[2.6],
[6.6],
[3.1],
[5.2],
[5.5],
[1.8],
[9.7],
[7.8],
[9.4],
[8.9],
[6. ],
[9.2],
[0.9],
[2. ],
[0.5],
[3.3],
```

```
       [3.9],
       [2.7],
       [8.3],
       [3.6],
       [2.8],
       [5.4],
       [1.4],
       [8. ],
       [0.7],
       [9.9],
       [7.7],
       [2. ],
       [0.1],
       [8.2],
       [7.1],
       [7.3],
       [7.7],
       [0.7],
       [3.6],
       [1.2],
       [8.6],
       [6.2],
       [3.3],
       [0.6],
       [3.1],
       [3.3],
       [7.3],
       [6.4],
       [8.9],
       [4.7],
       [1.2],
       [7.1],
       [7.6],
       [5.6],
       [7.7],
       [4.9],
       [5.2],
       [4.3],
       [0.3],
       [1.1]])
```

[3]:
```
y
```

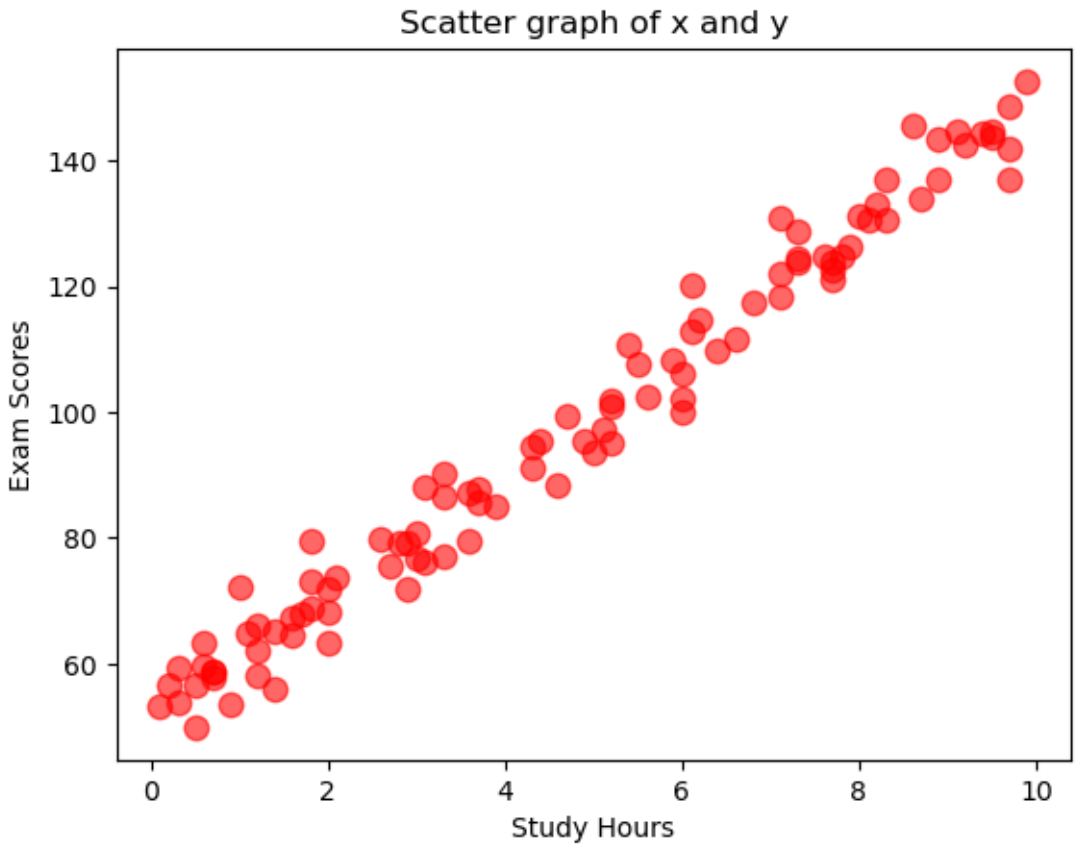
[3]:
```
array([ 87.9, 143.6, 123.7,  99.9,  64.5,  67.4,  63.2, 134. , 106.1,
       118.3,  56.6, 148.6, 130.6,  73.8,  68.7,  73.2,  76.9, 100.8,
        91.2,  71.8, 112.7,  65.3,  79.2,  85.5,  88.5, 126.4,  68.3,
        97.4, 108.4,  56.7, 120.2,  67.9,  57.8, 144.5, 137. , 130.7,
```

```
 80.8,  72.1, 117.5,  95.5,  62. ,  93.7,  59.2, 144.7,  79.8,
111.7,  88.2,  95. , 107.6,  79.4, 142. , 124.7, 144.4, 137. ,
102. , 142.5,  53.5,  72. ,  49.9,  90.3,  85. ,  75.5, 136.9,
 79.5,  79.2, 110.8,  56.1, 131.1,  58.8, 152.6, 121. ,  63.3,
 53.2, 133. , 121.9, 124.6, 123.7,  58.6,  87.3,  58. , 145.6,
114.7,  77.1,  59.6,  76.2,  86.5, 128.8, 109.7, 143.5,  99.3,
 66.1, 130.8, 124.9, 102.4, 122.6,  95.3, 101.9,  94.5,  53.9,
 64.9])
```

[4]:
```
#checking for missing data
data.isna().sum()
```

[4]:
```
Study Hours    0
Exam Scores    0
dtype: int64
```

[5]:
```
#Visualising the relationship between x and y
import matplotlib.pyplot as plt
plt.scatter(x,y, color='red', s=80, alpha=0.6)
plt.xlabel('Study Hours')
plt.ylabel('Exam Scores')
plt.title('Scatter graph of x and y')
plt.show()
```




[6]:
```
#Data preprocessing
from sklearn.model_selection import train_test_split
#1)splitting of data
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.8,␣
↪random_state=40)
```

[7]:
```
#2)standardising data
from sklearn.preprocessing import StandardScaler
scl = StandardScaler()
x_train_scaled = scl.fit_transform(x_train)
x_test_scaled = scl.transform(x_test)
```

[8]:
```
#Linear Regression Model
from sklearn.linear_model import LinearRegression
model = LinearRegression()
```

[9]:
```
#a)        Train a linear regression model on the training data.
model.fit(x_train_scaled, y_train)
```

[9]: LinearRegression()

[10]:
```
#predict y values
y_pred = model.predict(x_test_scaled)
y_pred
```

[10]:
```
array([ 64.19550884, 122.42827167,  87.10675979,  69.92332158,
       117.65509439, 145.33952263,  64.19550884, 115.74582348,
        94.74384344, 131.97462624, 122.42827167,  68.01405066,
       102.38092709, 101.42629163, 111.92728165, 102.38092709,
        84.24285342, 143.43025171, 120.51900076, 125.29217804,
        53.69451882,  66.10477975, 147.24879354, 129.11071987,
        71.83259249,  59.42233156,  59.42233156, 122.42827167,
       126.2468135 , 134.83853261, 110.9726462 ,  59.42233156,
       127.20144896,  88.06139525, 110.01801074,  54.64915427,
        81.37894705,  66.10477975, 105.24483346,  58.4676961 ,
        78.51504068,  64.19550884, 128.15608441,  57.51306064,
        96.65311435, 106.19946892,  99.51702072,  84.24285342,
       100.47165618,  63.24087338,  93.78920798, 143.43025171,
        81.37894705,  68.01405066,  79.46967614,  93.78920798,
        97.60774981, 145.33952263,  55.60378973, 130.06535532,
       139.61170989,  71.83259249,  88.06139525,  82.33358251,
        77.56040523,  89.97066616,  87.10675979, 126.2468135 ,
       102.38092709,  72.78722795, 109.06337529, 113.83655257,
        55.60378973, 110.01801074,  80.4243116 , 137.70243898,
        57.51306064,  69.92332158, 110.9726462 , 142.47561626])
```

[11]:
```
#b)        Evaluate the performance of the model on the testing data using␣
 ↪appropriate metrics
from sklearn.metrics import mean_absolute_error, mean_squared_error,r2_score
MAE = mean_absolute_error(y_test, y_pred)
MSE = mean_squared_error(y_test, y_pred)
R2  = r2_score(y_test, y_pred)

print('mean_absolute_error:',MAE)
print('mean_squared_error :',MSE)
print('r2_score           :',R2)
```

```
mean_absolute_error: 3.252620896629126
mean_squared_error : 17.981914358743296
r2_score           : 0.9782270818940543
```

[12]:
```
#c)Interpret the coefficients of the linear regression model and discuss their␣
 ↪significance.
coefficient = model.coef_
intercept   = model.intercept_
score       = model.score(x_test_scaled, y_test)

print('coefficient:',coefficient)
print('intercept  :',intercept)
```

```
print('score      :',score)
```

```
coefficient: [30.15741606]
intercept  : 102.19
score      : 0.9782270818940543
```

[14]:
```
#4)        Model Improvement
#a)        Implement any necessary feature engineering techniques to improve␣
 ↪the model's performance.
from sklearn.preprocessing import PolynomialFeatures
ply = PolynomialFeatures(degree=1)
x_train_ply = ply.fit_transform(x_train_scaled)
x_test_ply = ply.transform(x_test_scaled)
```

[15]:
```
#b)        Re-train the linear regression model on the updated dataset.
model_ply = LinearRegression()
model_ply.fit(x_train_ply, y_train)
```

[15]:
```
LinearRegression()
```

[16]:
```
#predictions for the updated dataset
new_y_pred = model_ply.predict(x_test_ply)
new_y_pred
```

[16]:
```
array([ 64.19550884, 122.42827167,  87.10675979,  69.92332158,
       117.65509439, 145.33952263,  64.19550884, 115.74582348,
        94.74384344, 131.97462624, 122.42827167,  68.01405066,
       102.38092709, 101.42629163, 111.92728165, 102.38092709,
        84.24285342, 143.43025171, 120.51900076, 125.29217804,
        53.69451882,  66.10477975, 147.24879354, 129.11071987,
        71.83259249,  59.42233156,  59.42233156, 122.42827167,
       126.2468135 , 134.83853261, 110.9726462 ,  59.42233156,
       127.20144896,  88.06139525, 110.01801074,  54.64915427,
        81.37894705,  66.10477975, 105.24483346,  58.4676961 ,
        78.51504068,  64.19550884, 128.15608441,  57.51306064,
        96.65311435, 106.19946892,  99.51702072,  84.24285342,
       100.47165618,  63.24087338,  93.78920798, 143.43025171,
        81.37894705,  68.01405066,  79.46967614,  93.78920798,
        97.60774981, 145.33952263,  55.60378973, 130.06535532,
       139.61170989,  71.83259249,  88.06139525,  82.33358251,
        77.56040523,  89.97066616,  87.10675979, 126.2468135 ,
       102.38092709,  72.78722795, 109.06337529, 113.83655257,
        55.60378973, 110.01801074,  80.4243116 , 137.70243898,
        57.51306064,  69.92332158, 110.9726462 , 142.47561626])
```

[17]:
```
#c)        Evaluate the performance of the improved model and compare it with␣
 ↪the initial model.
new_MAE = mean_absolute_error(y_test, new_y_pred)
```

```
new_MSE = mean_squared_error(y_test, new_y_pred)
new_R2  = r2_score(y_test, new_y_pred)

print('new_mean_absolute_error:',new_MAE)
print('new_mean_squared_error :',new_MSE)
print('new_r2_score           :',new_R2)
```

```
new_mean_absolute_error: 3.2526208966291223
new_mean_squared_error : 17.981914358743268
new_r2_score           : 0.9782270818940543
```

[18]:
```
#score of the improved model
new_score= model_ply.score(x_test_ply, y_test)
print('new_score           :',new_score)
```

```
new_score           : 0.9782270818940543
```

[ ]:
```

```