5-13-24 Python Assignment

May 20, 2024

1 5-13-24 Python Assignment

Using the SKLearn LinearRegression class to produce a model for weight in terms of height. Find the intercept and coefficient for your model and display it in your Jupyter notebook

[7]:

```python
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
```

[8]:

```python
mydata = pd.read_csv("https://raw.githubusercontent.com/aleahy-work/STAT223-S24/
  ↪main/Data/bodydata.csv")
mydata
```

[8]:

```
     AGE  GENDER  PULSE  SYSTOLIC  DIASTOLIC  HDL  LDL  WHITE   RED  PLATE  \
0     43       0     80       100         70   73   68    8.7  4.80    319
1     57       1     84       112         70   35  116    4.9  4.73    187
2     38       0     94       134         94   36  223    6.9  4.47    297
3     80       1     74       126         64   37   83    7.5  4.32    170
4     34       1     50       114         68   50  104    6.1  4.95    140
..   …      …      …        …          …    …   …     …     …      …
295   24       1     94        96         62   43  102    7.0  5.29    260
296   50       0     94       132         84   42   69    7.9  4.35    244
297   53       1     86       132         74   42  112    8.4  4.07     75
298   34       0     74       104         54   44  103    7.6  4.36    292
299   31       1     90       104         70   64  112    6.0  5.07    197

     WEIGHT  HEIGHT  WAIST  ARM CIRC   BMI
0      98.6   172.0  120.4      40.7  33.3
1      96.9   186.0  107.8      37.0  28.0
2     108.2   154.4  120.3      44.3  45.4
3      73.1   160.5   97.2      30.3  28.4
4      83.1   179.0   95.1      34.0  25.9
..      …      …      …       …     …
295    56.3   162.7   78.4      27.9  21.3
296   103.2   146.7  142.6      39.5  48.0
297   102.6   181.0  117.7      36.5  31.3
298    96.1   162.2  109.0      37.0  36.5
299    56.4   165.4   74.0      26.5  20.6
```

```
[300 rows x 15 columns]
```

```
[6]: mydata['HEIGHT'].shape
```

```
[6]: (300,)
```

```
[10]: myx = mydata['HEIGHT'].values.reshape(300,1)
```

```
[14]: myy = mydata['WEIGHT'].values
```

```
[11]: mylm = LinearRegression()
```

```
[13]: mymodel = mylm.fit(myx, myy)
```

```
[14]: mymodel.coef_
```

```
[14]: array([0.74691705])
```

```
[15]: mymodel.intercept_
```

```
[15]: -43.848386828346435
```

```
[16]: mymodel.score(myx, myy)
```

```
[16]: 0.13224470200036498
```

Produce a scatter plot of the height (=x) and weight (=y) and use matplotlib to overlay the regression line on top of the scatter plot. (Remember: Two points determine a line. How do you plot a line with matplotlib?)

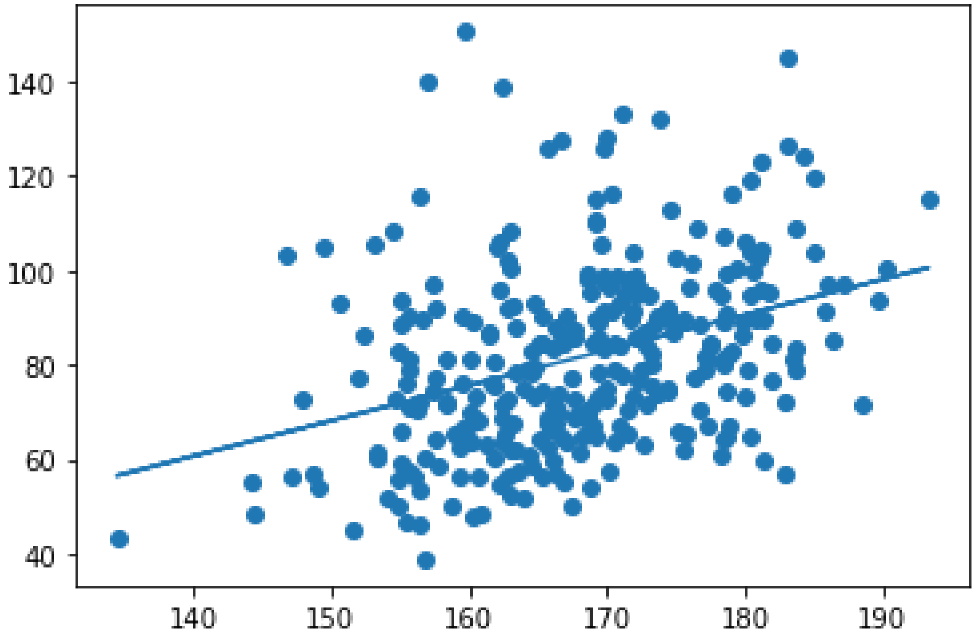
```
[20]: import matplotlib.pyplot as plt
```

```
[24]: y_pred = mylm.predict(myx)
      y_pred
```

```
[24]: array([ 84.62134503,  95.07818367,  71.47560503,  76.03179901,
              89.84976435,  80.66268469,  89.47630583,  72.44659719,
              73.86573957,  90.89544822,  80.36391787,  82.08182708,
              91.41829015,  83.12751094,  78.64600867,  75.28488196,
              91.19421503,  72.22252207,  89.70038094,  86.63802106,
              84.54665333,  87.53432151,  72.37190548,  90.44729799,
              67.44225298,  92.76274083,  82.60466901,  72.37190548,
              73.26820594,  76.85340776,  71.77437185,  72.29721378,
              71.92375526,  71.84906355,  76.77871605,  80.28922617,
              75.65834048,  78.34724185,  79.1688506 ,  83.27689435,
              71.92375526,  89.40161412,  88.13185515,  71.62498844,
              75.28488196,  80.96145151,  86.78740446,  73.26820594,
              82.75405242,  89.25223072,  92.91212424,  83.79973628,
69.3842373 ,  75.43426537,  74.46327321,  85.29357037,
86.56332935,  85.81641231,  75.73303219,  79.2435423 ,
84.69603674,  70.57930457,  93.80842469,  92.09051549,
56.61195582,  96.87078458,  86.33925424,  82.00713537,
82.4552856 ,  90.82075651,  84.77072844,  75.80772389,
74.01512298,  80.36391787,  97.84177674,  74.46327321,
88.9534639 ,  89.47630583,  89.62568924,  90.5966814 ,
88.72938878,  69.98177094,  91.56767356,  78.72070037,
76.10649071,  85.06949526,  81.40960174,  84.24788651,
74.76204003,  73.04413082,  73.86573957,  80.43860958,
80.96145151,  86.26456253,  82.90343583,  71.92375526,
89.55099753,  83.57566117,  83.12751094,  85.7417206 ,
85.44295378,  94.40595833,  83.50096946,  75.73303219,
82.3805939 ,  72.96943912,  80.43860958,  77.74970821,
87.23555469,  81.26021833,  71.99844696,  77.45094139,
79.01946719,  83.65035287,  83.42627776,  80.21453446,
74.91142344,  84.99480356,  67.21817787,  70.65399628,
86.93678787,  79.1688506 ,  71.77437185,  76.55464094,
92.83743254,  79.99045935,  79.84107594,  80.58799299,
82.30590219,  84.47196162,  75.65834048,  90.89544822,
79.09415889,  75.35957367,  76.33056583,  78.12316673,
82.4552856 ,  91.86644038,  69.68300412,  89.47630583,
82.00713537,  82.4552856 ,  88.72938878,  80.8120681 ,
89.55099753,  81.55898515,  75.21019026,  77.30155798,
83.79973628,  77.5256331 ,  87.83308833,  81.26021833,
75.8824156 ,  91.64236526,  85.96579571,  73.79104787,
79.84107594,  82.60466901,  77.00279117,  93.21089106,
82.00713537,  76.10649071,  84.54665333,  84.0238114 ,
80.28922617,  71.25152991,  81.33491003,  80.8120681 ,
73.11882253,  85.51764549,  72.37190548,  84.39726992,
83.79973628,  77.82439992,  82.23121049,  73.71635617,
77.30155798,  73.64166446,  77.6003248 ,  93.36027447,
79.91576764,  83.12751094,  79.84107594,  73.41758935,
82.97812753,  76.70402435,  81.40960174,  90.5966814 ,
88.35593026,  90.97013992,  84.24788651,  80.8867598 ,
73.04413082,  90.44729799,  82.60466901,  81.78306026,
94.85410856,  85.06949526,  91.41829015,  73.04413082,
83.50096946,  74.46327321,  81.33491003,  82.90343583,
80.58799299,  95.30225879,  64.00643457,  88.05716344,
83.94911969,  68.63732025,  75.8824156 ,  75.21019026,
88.20654685,  77.5256331 ,  89.62568924,  66.62064423,
78.94477549,  83.42627776,  86.33925424,  88.50531367,
86.41394594,  88.43062197,  80.21453446,  82.4552856 ,
89.40161412,  67.8157115 ,  89.92445606,  79.61700083,
77.74970821,  77.15217458,  89.62568924,  79.46761742,
90.6713731 ,  86.41394594,  82.3805939 ,  78.64600867,
79.54230912,  86.41394594,  77.82439992,  77.97378333,
85.51764549,  78.12316673,  94.40595833,  77.82439992,
86.71271276,  80.21453446,  85.51764549,  84.62134503,
86.41394594,  92.68804913,  82.97812753,  90.14853117,
90.82075651,  72.74536401,  81.63367685,  77.89909162,
89.25223072,  91.41829015,  66.0978023 ,  84.77072844,
82.97812753,  87.68370492, 100.53067811,  75.58364878,
79.01946719,  93.28558276,  70.65399628,  79.69169253,
79.69169253,  77.07748287,  80.8120681 ,  83.20220265,
87.3849381 ,  72.37190548,  82.08182708,  88.58000537,
77.74970821,  83.94911969,  85.89110401,  82.67936071,
92.01582378,  84.17319481,  98.28992697,  77.07748287,
93.28558276,  77.97378333,  63.85705116,  82.15651878,
95.97448413,  72.89474741,  85.21887867,  78.19785844,
82.4552856 ,  77.5256331 ,  87.3102464 ,  77.67501651,
65.72434378,  91.34359844,  77.30155798,  79.69169253])
```

[27]:

```
plt.scatter(mydata['HEIGHT'], mydata['WEIGHT'])
plt.plot(mydata['HEIGHT'], y_pred)
plt.show()
```



Using the SKLearn LinearRegression class to produce a model for weight in terms of height, waist (circumference), and arm circumference. Find the intercept and coefficients for your model and display it in your Jupyter notebook.

[10]:

```
myxx = mydata[['HEIGHT','WAIST','ARM CIRC']]
```

```
[12]: mylm2 = LinearRegression()
```

```
[15]: mymodel2 = mylm.fit(myxx, myy)
```

```
[16]: mymodel2.coef_
```

```
[16]: array([0.51363607, 0.6637826 , 1.6250768 ])
```

```
[18]: mymodel2.intercept_
```

```
[18]: -124.2377897844017
```

Compare the coefficient of determination (R^2) between this model and your first model. What conclusions do you draw?

```
[19]: mymodel2.score(myxx, myy)
```

```
[19]: 0.9413235306354137
```

Model 2 is a better predictor than model 1 because model 2 explains 94.1 proportion of variance in the dependent variable. Higher the R^2 value better the model