housing_data_peerreview

November 24, 2023

[28]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from sympy import *
```

0.1 Loading the Dataset

[44]:
```
house_df = pd.read_csv('SampleDatasetHousing_Data - housing_data.csv')
house_df
```

[44]:

	size	price
0	479.75	154282.128
1	525.81	168047.264
2	549.69	171795.240
3	620.82	175716.480
4	681.07	180307.216
..	…	…
95	1300.96	444192.008
96	1496.36	454512.760
97	1275.46	458674.256
98	1842.51	494778.992
99	1508.84	500681.128

[100 rows x 2 columns]

[45]:
```
house_df.sort_values('price')
```

[45]:

	size	price
0	479.75	154282.128
1	525.81	168047.264
2	549.69	171795.240
3	620.82	175716.480
4	681.07	180307.216
..	…	…
95	1300.96	444192.008
96	1496.36	454512.760
97	1275.46	458674.256

```
98  1842.51  494778.992
99  1508.84  500681.128

[100 rows x 2 columns]
```

[46]:
```
house_df = house_df.to_numpy()
house_df
```

[46]:
```
array([[4.79750000e+02, 1.54282128e+05],
       [5.25810000e+02, 1.68047264e+05],
       [5.49690000e+02, 1.71795240e+05],
       [6.20820000e+02, 1.75716480e+05],
       [6.81070000e+02, 1.80307216e+05],
       [6.43410000e+02, 1.81587576e+05],
       [5.18380000e+02, 1.83459488e+05],
       [6.81070000e+02, 1.90909056e+05],
       [5.25810000e+02, 1.91486896e+05],
       [6.98290000e+02, 2.01778048e+05],
       [6.33190000e+02, 2.04302976e+05],
       [5.97900000e+02, 2.07742248e+05],
       [5.49800000e+02, 2.11724096e+05],
       [6.01660000e+02, 2.11904536e+05],
       [6.94520000e+02, 2.15472104e+05],
       [6.82260000e+02, 2.17468224e+05],
       [6.94520000e+02, 2.18630608e+05],
       [6.48290000e+02, 2.25145248e+05],
       [6.81070000e+02, 2.25451984e+05],
       [6.81070000e+02, 2.25452320e+05],
       [6.47500000e+02, 2.25656120e+05],
       [6.85480000e+02, 2.28313024e+05],
       [6.56220000e+02, 2.28581528e+05],
       [5.70250000e+02, 2.33493208e+05],
       [6.23940000e+02, 2.34178160e+05],
       [6.43090000e+02, 2.34314144e+05],
       [5.75190000e+02, 2.45050280e+05],
       [6.98290000e+02, 2.45747200e+05],
       [6.90540000e+02, 2.48337600e+05],
       [6.94520000e+02, 2.51140656e+05],
       [8.28160000e+02, 2.51188824e+05],
       [5.69170000e+02, 2.51332592e+05],
       [6.82260000e+02, 2.51560040e+05],
       [5.49800000e+02, 2.52460400e+05],
       [6.47500000e+02, 2.55629160e+05],
       [6.33190000e+02, 2.57828416e+05],
       [1.02195000e+03, 2.58637008e+05],
       [6.17050000e+02, 2.62423504e+05],
       [6.98290000e+02, 2.62477856e+05],
```

```
[6.98500000e+02, 2.63311696e+05],
[5.70890000e+02, 2.65129064e+05],
[6.98290000e+02, 2.66684248e+05],
[6.20710000e+02, 2.68125080e+05],
[8.23210000e+02, 2.69225920e+05],
[7.27880000e+02, 2.69523056e+05],
[5.49800000e+02, 2.71726752e+05],
[5.70890000e+02, 2.71793312e+05],
[7.05290000e+02, 2.74922856e+05],
[1.02195000e+03, 2.76875632e+05],
[1.06036000e+03, 2.79555096e+05],
[4.87290000e+02, 2.81626336e+05],
[6.43090000e+02, 2.82683544e+05],
[8.57540000e+02, 2.85223176e+05],
[6.85480000e+02, 2.86161600e+05],
[8.23210000e+02, 2.87350000e+05],
[6.85480000e+02, 2.92965216e+05],
[1.06036000e+03, 2.93044496e+05],
[6.85480000e+02, 2.94582944e+05],
[6.81070000e+02, 2.97760440e+05],
[6.94520000e+02, 2.98170880e+05],
[6.56220000e+02, 2.98926496e+05],
[1.02776000e+03, 2.99416976e+05],
[6.98290000e+02, 3.00061480e+05],
[1.01033000e+03, 3.01635728e+05],
[6.22970000e+02, 3.02000920e+05],
[6.85480000e+02, 3.02393384e+05],
[6.85480000e+02, 3.03597216e+05],
[8.27090000e+02, 3.04587272e+05],
[1.02195000e+03, 3.10045712e+05],
[1.03744000e+03, 3.10522592e+05],
[1.03206000e+03, 3.20345520e+05],
[1.02841000e+03, 3.27252112e+05],
[8.27840000e+02, 3.30677128e+05],
[6.82260000e+02, 3.31101344e+05],
[1.02195000e+03, 3.34938872e+05],
[1.07155000e+03, 3.38078168e+05],
[1.18862000e+03, 3.42988456e+05],
[1.23693000e+03, 3.54512112e+05],
[1.00925000e+03, 3.55251200e+05],
[1.20562000e+03, 3.59674440e+05],
[1.10330000e+03, 3.62519720e+05],
[1.33410000e+03, 3.65863936e+05],
[1.49636000e+03, 3.68988432e+05],
[1.00925000e+03, 3.76253808e+05],
[1.18346000e+03, 3.82120152e+05],
[1.50475000e+03, 3.83635568e+05],
```

```
[1.02195000e+03, 3.93069760e+05],
[1.28385000e+03, 3.95242096e+05],
[1.50475000e+03, 4.01255608e+05],
[1.33410000e+03, 4.06852304e+05],
[1.12234000e+03, 4.08637816e+05],
[1.20745000e+03, 4.12569472e+05],
[1.33410000e+03, 4.14682648e+05],
[1.00925000e+03, 4.18753008e+05],
[1.37972000e+03, 4.40201616e+05],
[1.30096000e+03, 4.44192008e+05],
[1.49636000e+03, 4.54512760e+05],
[1.27546000e+03, 4.58674256e+05],
[1.84251000e+03, 4.94778992e+05],
[1.50884000e+03, 5.00681128e+05]])
```

0.2 Converting the data to matrices for calculation; separating target from predictor

[47]:
```
ones = np.ones((100,1))
ones = Matrix(ones)
```

[54]:
```
X = Matrix(house_df[:,0])
X = X.col_insert(0, ones)
X
```

[54]:

$$\begin{bmatrix} 1.0 & 479.75 \\ 1.0 & 525.81 \\ 1.0 & 549.69 \\ 1.0 & 620.82 \\ 1.0 & 681.07 \\ 1.0 & 643.41 \\ 1.0 & 518.38 \\ 1.0 & 681.07 \\ 1.0 & 525.81 \\ 1.0 & 698.29 \\ 1.0 & 633.19 \\ 1.0 & 597.9 \\ 1.0 & 549.8 \\ 1.0 & 601.66 \\ 1.0 & 694.52 \\ 1.0 & 682.26 \\ 1.0 & 694.52 \\ 1.0 & 648.29 \\ 1.0 & 681.07 \\ 1.0 & 681.07 \\ 1.0 & 647.5 \\ 1.0 & 685.48 \\ 1.0 & 656.22 \\ 1.0 & 570.25 \\ 1.0 & 623.94 \\ 1.0 & 643.09 \\ 1.0 & 575.19 \\ 1.0 & 698.29 \\ 1.0 & 690.54 \\ 1.0 & 694.52 \\ 1.0 & 828.16 \\ 1.0 & 569.17 \\ 1.0 & 682.26 \\ 1.0 & 549.8 \\ 1.0 & 647.5 \\ 1.0 & 633.19 \\ 1.0 & 1021.95 \\ 1.0 & 617.05 \\ 1.0 & 698.29 \\ 1.0 & 698.5 \\ 1.0 & 570.89 \\ 1.0 & 698.29 \\ 1.0 & 620.71 \\ 1.0 & 823.21 \\ 1.0 & 727.88 \\ 1.0 & 549.8 \\ 1.0 & 570.89 \\ 1.0 & 705.29 \\ 1.0 & 1021.95 \\ 1.0 & 1060.36 \\ 1.0 & 487.29 \\ 1.0 & 643.09 \\ 1.0 & 857.54 \end{bmatrix}$$

[60]:
```
Y = Matrix(house_df[:,1])
Y
```

[60]:

154282.128
168047.264
171795.24
175716.48
180307.216
181587.576
183459.488
190909.056
191486.896
201778.048
204302.976
207742.248
211724.096
211904.536
215472.104
217468.224
218630.608
225145.248
225451.984
225452.32
225656.12
228313.024
228581.528
233493.208
234178.16
234314.144
245050.28
245747.2
248337.6
251140.656
251188.824
251332.592
251560.04
252460.4
255629.16
257828.416
258637.008
262423.504
262477.856
263311.696
265129.064
266684.248
268125.08
269225.92
269523.056
271726.752
271793.312
274922.856
276875.632
279555.096
281626.336
282683.544
285223.176

0.3 Applying the normal equation

[61]:
```
P1 = X.T*X
P1
```

[61]:
$$\begin{bmatrix} 100.0 & 85302.42 \\ 85302.42 & 81553199.7694 \end{bmatrix}$$

[62]:
```
P = P1.inv()
P
```

[62]:
$$\begin{bmatrix} 0.0927988292426997 & -9.70650413466578 \cdot 10^{-5} \\ -9.70650413466578 \cdot 10^{-5} & 1.1378931728626 \cdot 10^{-7} \end{bmatrix}$$

[63]:
```
Q = X.T*Y
Q
```

[63]:
$$\begin{bmatrix} 29228947.016 \\ 26894332141.2427 \end{bmatrix}$$

[64]:
```
A = P*Q
A
```

[64]:
$$\begin{bmatrix} 101912.60180123 \\ 223.178742594607 \end{bmatrix}$$

0.4 Finding the price prediction for size 3000

[70]:
```
pred_value = 3000
prediction = 223.18*pred_value + 101912.6
prediction
```

[70]:
```
771452.6
```

[ ]:
```

```