# sklearn1

April 23, 2024

[1]:
```
import numpy as np
```

[2]:
```
x=np.arange(10).reshape(5,2)
```

[3]:
```
y=range(5)
list(y)
```

[3]:
```
[0, 1, 2, 3, 4]
```

[4]:
```
x
```

[4]:
```
array([[0, 1],
       [2, 3],
       [4, 5],
       [6, 7],
       [8, 9]])
```

[5]:
```
from sklearn.model_selection import train_test_split
```

[6]:
```
x_train, x_test, y_train, y_test = train_test_split(x,y)
```

[7]:
```
x_train
```

[7]:
```
array([[2, 3],
       [6, 7],
       [4, 5]])
```

[8]:
```
x_test
```

[8]:
```
array([[0, 1],
       [8, 9]])
```

[9]:
```
y_train
```

[9]:
```
[1, 3, 2]
```

[10]:
```
y_test
```

```
[10]: [0, 4]
```

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

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

```
[13]: model.fit(x_train,y_train)
```

```
[13]: LinearRegression()
```

```
[14]: predict=model.predict(x_test)
      predict
```

```
[14]: array([8.04911693e-16, 4.00000000e+00])
```

```
[15]: y_test
```

```
[15]: [0, 4]
```

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

```
[17]: metrics.mean_absolute_error(y_test,predict)
```

```
[17]: 8.465450562766819e-16
```

```
[18]: import pandas as pd
```

```
[20]: ushousing=pd.read_csv('USA_Housing.csv')
      ushousing
```

```
[20]:       Avg. Area Income  Avg. Area House Age  Avg. Area Number of Rooms  \
      0         79545.458574             5.682861                   7.009188
      1         79248.642455             6.002900                   6.730821
      2         61287.067179             5.865890                   8.512727
      3         63345.240046             7.188236                   5.586729
      4         59982.197226             5.040555                   7.839388
      …                  …                  …                        …
      4995      60567.944140             7.830362                   6.137356
      4996      78491.275435             6.999135                   6.576763
      4997      63390.686886             7.250591                   4.805081
      4998      68001.331235             5.534388                   7.130144
      4999      65510.581804             5.992305                   6.792336

            Avg. Area Number of Bedrooms  Area Population         Price  \
      0                             4.09     23086.800503  1.059034e+06
      1                             3.09     40173.072174  1.505891e+06
      2                             5.13     36882.159400  1.058988e+06
```

```
3                             3.26     34310.242831  1.260617e+06
4                             4.23     26354.109472  6.309435e+05
…                            …              …           …
4995                          3.46     22837.361035  1.060194e+06
4996                          4.02     25616.115489  1.482618e+06
4997                          2.13     33266.145490  1.030730e+06
4998                          5.44     42625.620156  1.198657e+06
4999                          4.07     46501.283803  1.298950e+06

                                                Address
0     208 Michael Ferry Apt. 674\nLaurabury, NE 3701…
1     188 Johnson Views Suite 079\nLake Kathleen, CA…
2     9127 Elizabeth Stravenue\nDanieltown, WI 06482…
3                             USS Barnett\nFPO AP 44820
4                            USNS Raymond\nFPO AE 09386
…                                                   …
4995                   USNS Williams\nFPO AP 30153-7653
4996              PSC 9258, Box 8489\nAPO AA 42991-3352
4997  4215 Tracy Garden Suite 076\nJoshualand, VA 01…
4998                          USS Wallace\nFPO AE 73316
4999  37778 George Ridges Apt. 509\nEast Holly, NV 2…

[5000 rows x 7 columns]
```

[21]:

```
ushousing.columns
```

[21]:

```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Address'],
      dtype='object')
```

[22]:

```
ushousing.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Avg. Area Income              5000 non-null   float64
 1   Avg. Area House Age           5000 non-null   float64
 2   Avg. Area Number of Rooms     5000 non-null   float64
 3   Avg. Area Number of Bedrooms  5000 non-null   float64
 4   Area Population               5000 non-null   float64
 5   Price                         5000 non-null   float64
 6   Address                       5000 non-null   object
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
```

```
[23]: x = ushousing[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of␣
       ↪Rooms','Avg. Area Number of Bedrooms',
                    'Area Population']]
      y = ushousing['Price']
```

```
[24]: from sklearn.model_selection import train_test_split
```

```
[25]: x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.4,␣
       ↪random_state=101)
```

```
[26]: x_train
```

```
[26]:       Avg. Area Income  Avg. Area House Age  Avg. Area Number of Rooms  \
      1303      68091.179676             5.364208                   7.502956
      1051      75729.765546             5.580599                   7.642973
      4904      70885.420819             6.358747                   7.250241
      931       73386.407340             4.966360                   7.915453
      4976      75046.313791             5.351169                   7.797825
      …                  …                  …                        …
      4171      56610.642563             4.846832                   7.558137
      599       70596.850945             6.548274                   6.539986
      1361      55621.899104             3.735942                   6.868291
      1547      63044.460096             5.935261                   5.913454
      4959      75078.791516             7.644779                   8.440726

            Avg. Area Number of Bedrooms  Area Population
      1303                          3.10     44557.379656
      1051                          4.21     29996.018448
      4904                          5.42     38627.301473
      931                           4.30     38413.490484
      4976                          5.23     34107.888619
      …                              …              …
      4171                          3.29     25494.740298
      599                           3.10     51614.830136
      1361                          2.30     63184.613147
      1547                          4.10     32725.279544
      4959                          4.33     56148.449322

      [3000 rows x 5 columns]
```

```
[27]: x_test
```

```
[27]:       Avg. Area Income  Avg. Area House Age  Avg. Area Number of Rooms  \
      1718      66774.995817             5.717143                   7.795215
      2511      62184.539375             4.925758                   7.427689
      345       73643.057298             6.766853                   8.337085
      2521      61909.041438             6.228343                   6.593138
```

```
54        72942.705059             4.786222                   7.319886
…                 …                 …                        …
1776      65173.050438             7.679469                   6.602618
4269      42969.659393             6.295501                   7.885507
1661      48735.924512             5.543730                   6.091906
2410      65081.584048             5.433570                   9.212518
2302      65969.707036             7.325976                   8.020966

      Avg. Area Number of Bedrooms  Area Population
1718                          4.32     36788.980327
2511                          6.22     26008.309124
345                           3.34     43152.139577
2521                          4.29     28953.925377
54                            6.41     24377.909049
…                            …              …
1776                          4.23     44125.540782
4269                          4.38     29594.089863
1661                          2.43     19682.347295
2410                          5.14     37594.493458
2302                          4.09     61772.756810

[2000 rows x 5 columns]
```

[28]:
```
y_train
```

[28]:
```
1303    1.489648e+06
1051    1.183015e+06
4904    1.547889e+06
931     1.186442e+06
4976    1.340344e+06
            …
4171    7.296417e+05
599     1.599479e+06
1361    1.102641e+06
1547    8.650995e+05
4959    2.108376e+06
Name: Price, Length: 3000, dtype: float64
```

[29]:
```
y_test
```

[29]:
```
1718    1.251689e+06
2511    8.730483e+05
345     1.696978e+06
2521    1.063964e+06
54      9.487883e+05
            …
1776    1.489520e+06
```

```
4269    7.777336e+05
1661    1.515271e+05
2410    1.343824e+06
2302    1.906025e+06
Name: Price, Length: 2000, dtype: float64
```

[30]:
```
from sklearn.linear_model import LinearRegression
```

[31]:
```
model= LinearRegression()
```

[32]:
```
model.fit(x_train,y_train)
```

[32]:
```
LinearRegression()
```

[33]:
```
predict=model.predict(x_test)
predict.round(2)
```

[33]:
```
array([1260960.71,  827588.76, 1742421.24, …,  372191.41, 1365217.15,
       1914519.54])
```

[34]:
```
y_test
```

[34]:
```
1718    1.251689e+06
2511    8.730483e+05
345     1.696978e+06
2521    1.063964e+06
54      9.487883e+05
            …
1776    1.489520e+06
4269    7.777336e+05
1661    1.515271e+05
2410    1.343824e+06
2302    1.906025e+06
Name: Price, Length: 2000, dtype: float64
```

[35]:
```
from sklearn import metrics
```

[36]:
```
metrics.mean_absolute_error(y_test,predict)
```

[36]:
```
82288.22251914955
```

[37]:
```
metrics.mean_squared_error(y_test,predict)
```

[37]:
```
10460958907.209503
```

[38]:
```
model.intercept_
```

```
[38]: -2640159.7968519107
```

```
[39]: model.coef_
```

```
[39]: array([2.15282755e+01, 1.64883282e+05, 1.22368678e+05, 2.23380186e+03,
              1.51504200e+01])
```

```
[ ]:
```