Regressão Linear - TE2

April 29, 2025

[2]:

```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

[3]:

```python
casas = pd.read_csv('USA_Housing.csv')
```

[4]:

```python
casas.head()
```

[4]:

```
   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

   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

                                             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
```

0.0.1 Modelo supervisionado

- separação entre base de treino e base de teste

0.0.2 Modelo não supervisionado

- não existe a separação entre base de treino e teste

[5]:

```python
casas.info()
```

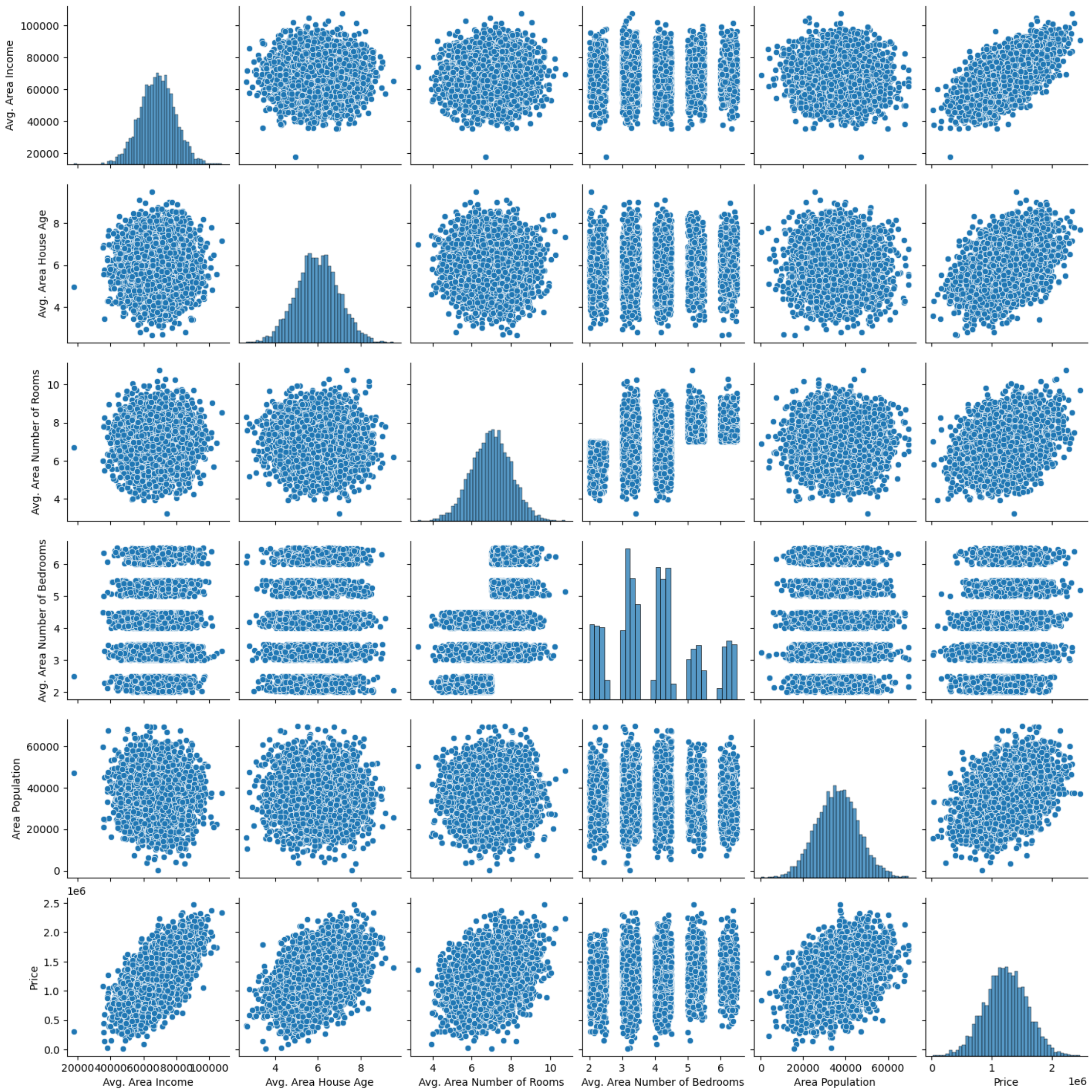
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Avg. Area Income              5000 non-null   float64
 1   Avg. Area House 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
```

[6]:

```python
sns.pairplot(casas)
```

[6]:

```
<seaborn.axisgrid.PairGrid at 0x7fa925e78c50>
```



```
[7]: casas.corr()
```

```
/tmp/ipykernel_253299/2249505781.py:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  casas.corr()
```

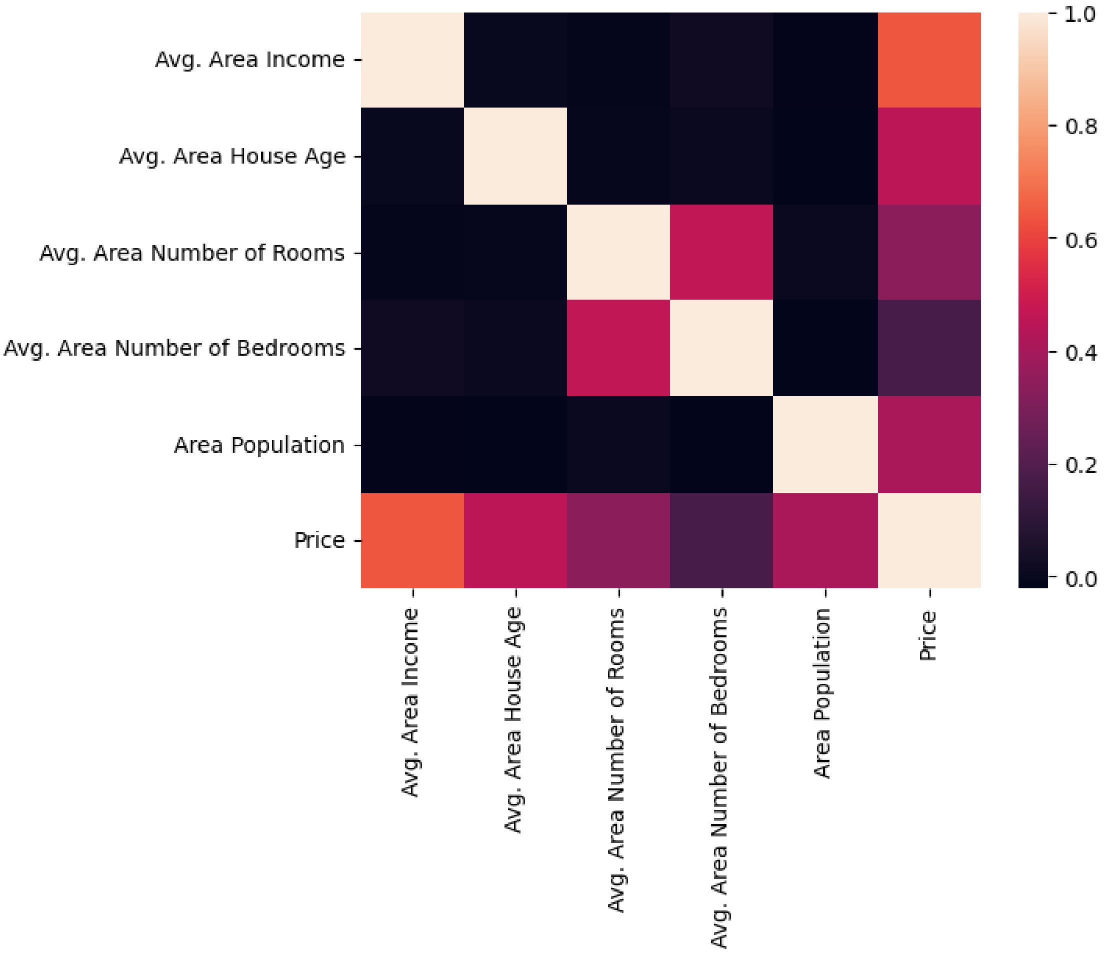
```
[7]:                               Avg. Area Income  Avg. Area House Age  \
 Avg. Area Income                      1.000000            -0.002007
 Avg. Area House Age                  -0.002007             1.000000
 Avg. Area Number of Rooms            -0.011032            -0.009428
 Avg. Area Number of Bedrooms          0.019788             0.006149
```

```
Area Population                      -0.016234            -0.018743
Price                                 0.639734             0.452543

                              Avg. Area Number of Rooms  \
Avg. Area Income                              -0.011032
Avg. Area House Age                           -0.009428
Avg. Area Number of Rooms                      1.000000
Avg. Area Number of Bedrooms                   0.462695
Area Population                                0.002040
Price                                          0.335664

                              Avg. Area Number of Bedrooms  Area Population  \
Avg. Area Income                                  0.019788        -0.016234
Avg. Area House Age                               0.006149        -0.018743
Avg. Area Number of Rooms                         0.462695         0.002040
Avg. Area Number of Bedrooms                      1.000000        -0.022168
Area Population                                  -0.022168         1.000000
Price                                             0.171071         0.408556

                                 Price
Avg. Area Income              0.639734
Avg. Area House Age           0.452543
Avg. Area Number of Rooms     0.335664
Avg. Area Number of Bedrooms  0.171071
Area Population               0.408556
Price                         1.000000
```

[8]:
```
sns.heatmap(casas.corr())
```

```
/tmp/ipykernel_253299/254642754.py:1: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
  sns.heatmap(casas.corr())
```

[8]:
```
<AxesSubplot: >
```



```
[9]: casas.columns
```

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

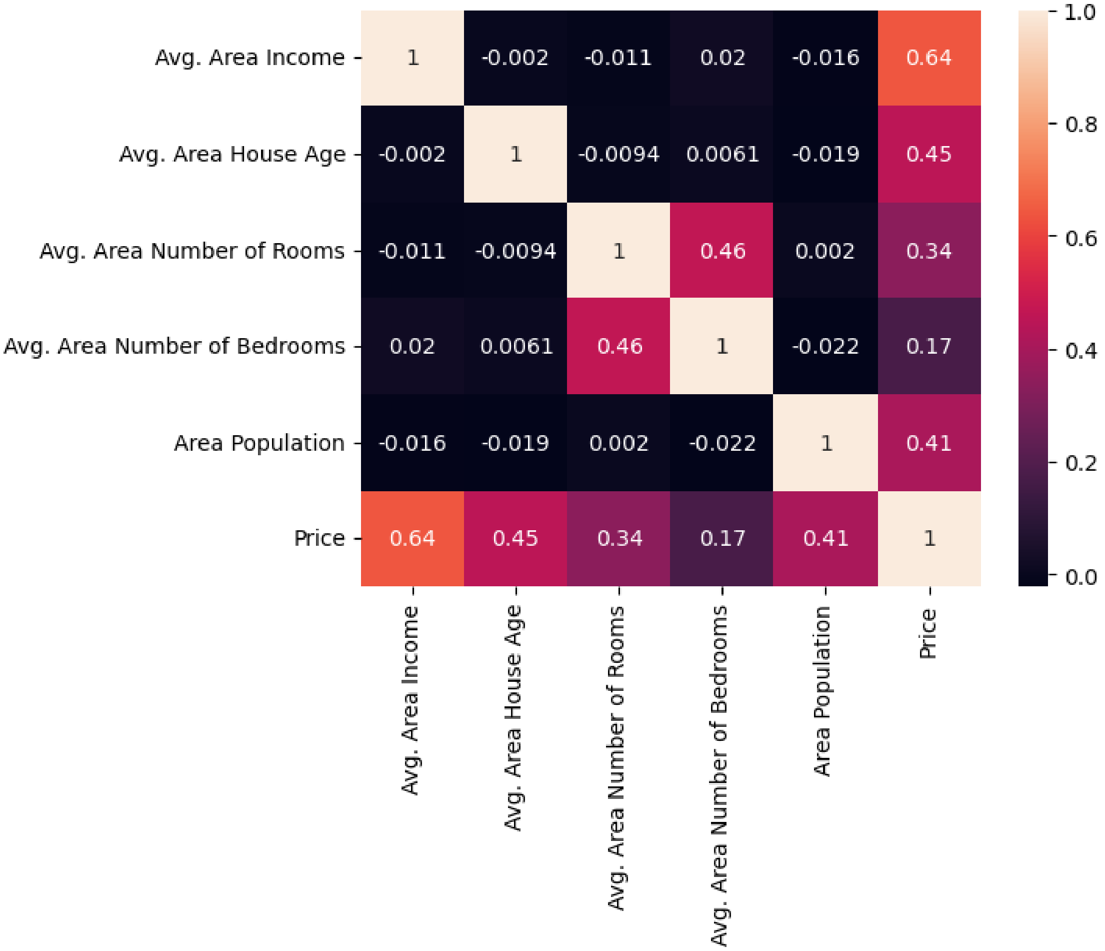
```
[10]: casas[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
             'Avg. Area Number of Bedrooms', 'Area Population', 'Price']].corr()
```

```
[10]:                               Avg. Area Income  Avg. Area House Age  \
      Avg. Area Income                      1.000000            -0.002007
      Avg. Area House Age                  -0.002007             1.000000
      Avg. Area Number of Rooms            -0.011032            -0.009428
      Avg. Area Number of Bedrooms          0.019788             0.006149
      Area Population                      -0.016234            -0.018743
      Price                                 0.639734             0.452543
```

```
                              Avg. Area Number of Rooms  \
Avg. Area Income                              -0.011032
Avg. Area House Age                           -0.009428
Avg. Area Number of Rooms                      1.000000
Avg. Area Number of Bedrooms                   0.462695
Area Population                                0.002040
Price                                          0.335664

                              Avg. Area Number of Bedrooms  Area Population  \
Avg. Area Income                                  0.019788        -0.016234
Avg. Area House Age                               0.006149        -0.018743
Avg. Area Number of Rooms                         0.462695         0.002040
Avg. Area Number of Bedrooms                      1.000000        -0.022168
Area Population                                  -0.022168         1.000000
Price                                             0.171071         0.408556

                                 Price
Avg. Area Income              0.639734
Avg. Area House Age           0.452543
Avg. Area Number of Rooms     0.335664
Avg. Area Number of Bedrooms  0.171071
Area Population               0.408556
Price                         1.000000
```

```
[12]: sns.heatmap(casas[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
             'Avg. Area Number of Bedrooms', 'Area Population', 'Price']].corr(), annot=True)
```

```
[12]: <AxesSubplot: >
```

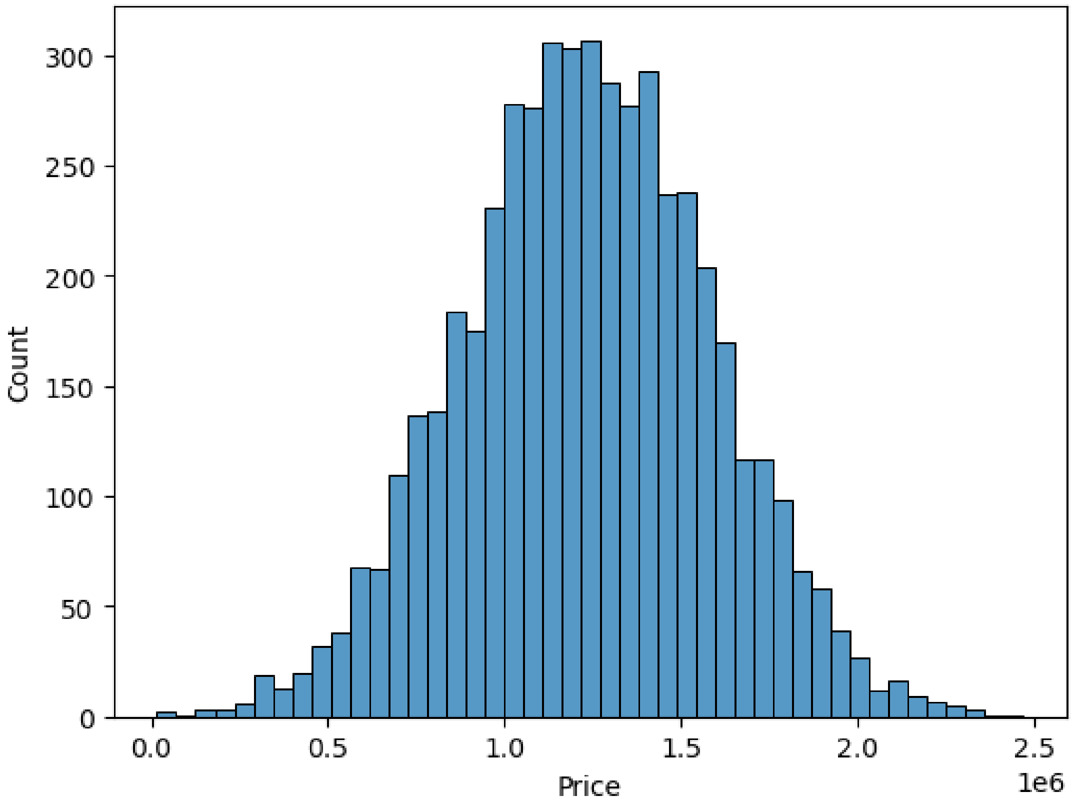


[13]:

```
sns.histplot(casas['Price'])
```

[13]:

```
<AxesSubplot: xlabel='Price', ylabel='Count'>
```



Modelo de regressão

- X -> VARIÁVEIS PREDITORAS
- Y -> VARIÁVEL QUE QUERO PREDIZER (DESCOBRIR) - ALVO

[14]:
```
casas.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
```

[15]:
```
casas.columns
```

[15]:
```
Index(['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population', 'Price', 'Address'],
      dtype='object')
```

[16]:
```
X = casas[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of␣
 ↪Rooms',
       'Avg. Area Number of Bedrooms', 'Area Population']]
Y = casas['Price']
```

[17]:
```
X
```

[17]:
```
      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
0                             4.09     23086.800503
1                             3.09     40173.072174
2                             5.13     36882.159400
3                             3.26     34310.242831
4                             4.23     26354.109472
…                              …                …
4995                          3.46     22837.361035
4996                          4.02     25616.115489
4997                          2.13     33266.145490
4998                          5.44     42625.620156
4999                          4.07     46501.283803

[5000 rows x 5 columns]
```

[18]:
```
Y
```

[18]:
```
0       1.059034e+06
1       1.505891e+06
2       1.058988e+06
3       1.260617e+06
```

```
4       6.309435e+05
            …
4995    1.060194e+06
4996    1.482618e+06
4997    1.030730e+06
4998    1.198657e+06
4999    1.298950e+06
Name: Price, Length: 5000, dtype: float64
```

0.0.3 Divisão do conjunto de dados

0.0.4 %pip install scikit-learn

[19]:
```
from sklearn.model_selection import train_test_split
```

[20]:
```
X_treino, X_teste, Y_treino, Y_teste = train_test_split(X,Y,
                                                        train_size=0.7,␣
 ↪random_state=46)
```

[21]:
```
X_treino
```

[21]:
```
      Avg. Area Income  Avg. Area House Age  Avg. Area Number of Rooms  \
456       62653.092462             6.543984                   7.884326
4480      71481.034926             5.188492                   7.152361
3564      63889.411593             5.548089                   6.357831
587       51918.546873             5.892497                   6.708809
2634      42814.993038             5.247613                   6.080981
…                  …                  …                         …
3933      71229.357964             4.850191                   6.978128
3787      50362.538095             5.582574                   4.608843
658       73829.777741             6.130263                   6.182843
2451      63421.903955             7.594954                   8.777735
2490      65010.553474             6.299952                   7.529846

      Avg. Area Number of Bedrooms  Area Population
456                           3.28     41467.867658
4480                          4.32     45246.174103
3564                          2.11     34442.099648
587                           3.37     24734.406520
2634                          4.02     41426.389765
…                              …              …
3933                          4.15     27168.842194
3787                          2.40     23611.056225
658                           3.50     51385.523241
2451                          3.09     11511.387050
2490                          6.32     34706.866627

[3500 rows x 5 columns]
```

```
[22]: Y_treino
```

```
[22]: 456     1.382110e+06
      4480    1.530013e+06
      3564    1.134126e+06
      587     6.610434e+05
      2634    4.525302e+05
                  …
      3933    6.997873e+05
      3787    3.141678e+05
      658     1.376493e+06
      2451    1.432318e+06
      2490    1.305186e+06
      Name: Price, Length: 3500, dtype: float64
```

```
[23]: X_treino.shape[0]
```

```
[23]: 3500
```

```
[24]: X_teste.shape[0]
```

```
[24]: 1500
```

```
[25]: Y_treino.shape
```

```
[25]: (3500,)
```

```
[26]: Y_teste.shape
```

```
[26]: (1500,)
```

Criando o modelo de treino

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

```
[28]: LR=LinearRegression()
```

```
[29]: LR.fit(X_treino,Y_treino )
```

```
[29]: LinearRegression()
```

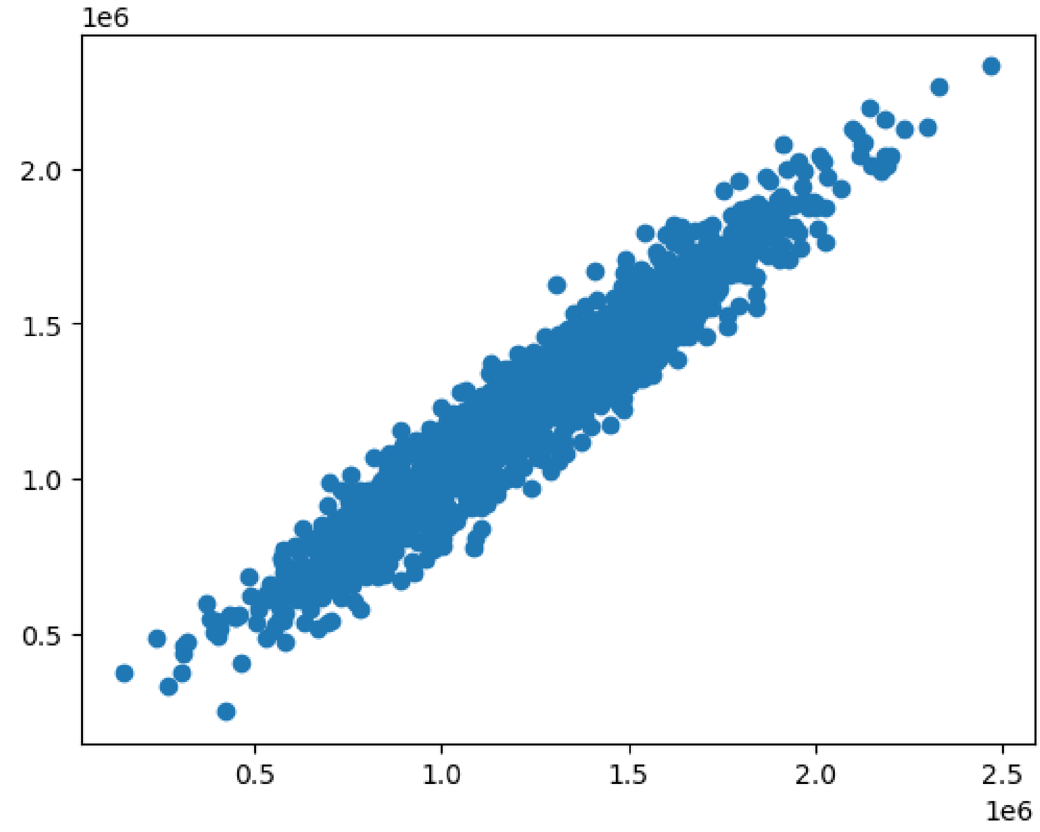
```
[30]: y_predicao = LR.predict(X_teste)
```

```
[31]: y_predicao
```

```
[31]: array([1176591.88192438,  997847.03239798, 1179674.83216237, …,
              934885.47457475, 1497530.06928932,  866365.31013406])
```

```
[32]: plt.scatter(Y_teste, y_predicao)
```

[32]:
```
<matplotlib.collections.PathCollection at 0x7fa913c68590>
```



[33]:
```
from sklearn import metrics
```

[34]:
```
metrics.mean_absolute_error(Y_teste, y_predicao)
```

[34]:
```
79806.83254169731
```

Y_teste

y_predicao

((Y_teste - y_predicao)$^{2)}$1/2

[35]:
```
metrics.mean_squared_error(Y_teste, y_predicao)
```

[35]:
```
9993926845.480902
```

Y_teste

y_predicao

((Y_teste - y_predicao)^2)

Atividade

Com base na regressão linear vista em aula, remova um dos preditores e compare o resultado obtido com esse modelo.

```
[37]: X2 = casas[['Avg. Area Income', 'Avg. Area House Age', 'Area Population']]
      Y2 = casas['Price']
```

```
[38]: X2_treino, X2_teste, Y2_treino, Y2_teste = train_test_split(X2,Y2,
                                                                  train_size=0.7,␣
        ↪random_state=46)
```

```
[39]: LR2=LinearRegression()
```

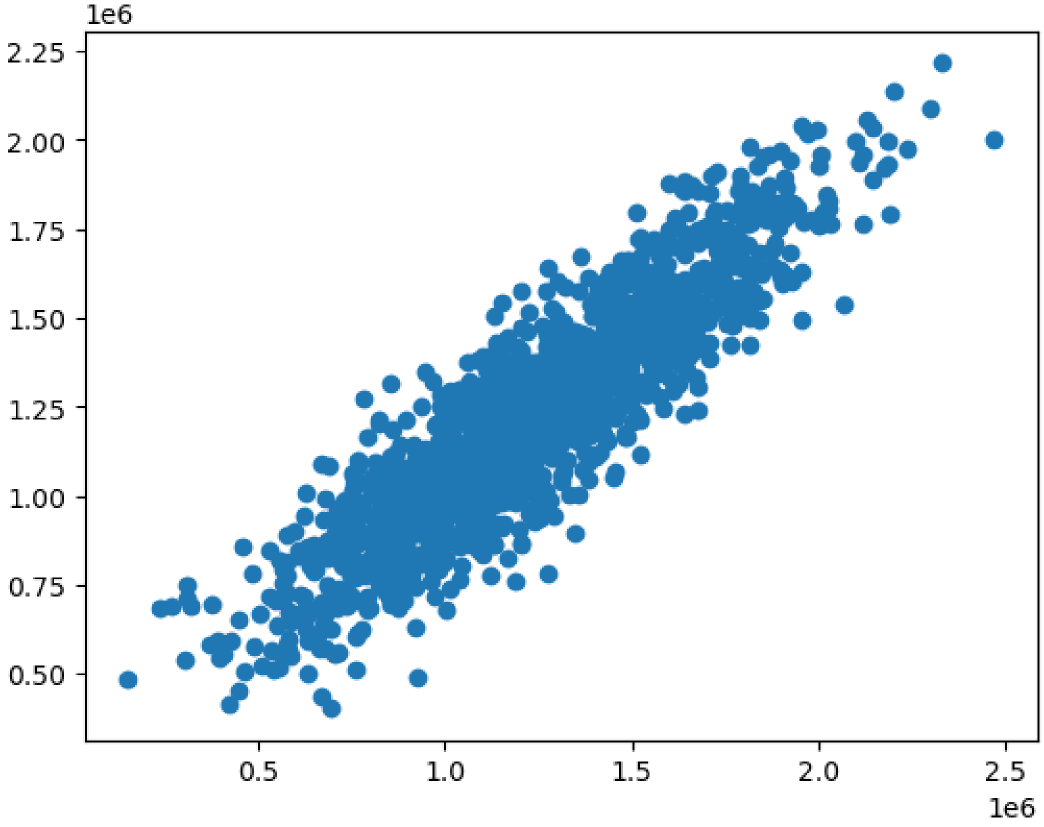
```
[40]: LR2.fit(X2_treino,Y2_treino)
```

```
[40]: LinearRegression()
```

```
[41]: Y2_predicao = LR2.predict(X2_teste)
```

```
[42]: plt.scatter(Y2_teste, Y2_predicao)
```

```
[42]: <matplotlib.collections.PathCollection at 0x7fa911a6d750>
```



```
[43]: metrics.mean_absolute_error(Y2_teste, Y2_predicao)
```

```
[43]: 127067.82406794395
```

```
[ ]:
```