regressao_linear_casas

February 27, 2021

In [1]:
```
# Regressão Linear- Avaliação de Preços de Casas/ USA

# Utilizando Base de Dados do Kaggle

https://www.kaggle.com/harlfoxem/housesalesprediction

import pandas as pd
```

1 PANDAS/TENSORFLOW

In [2]:
```
# Carregando o conjunto de Dados

base = pd.read_csv('house_prices.csv')
```

In [3]:
```
#Realizando a primeira análise do cabeçalho

base.head()
```

Out[3]:

	id	date	price	bedrooms	bathrooms	sqft_living	\
0	7129300520	20141013T000000	221900.0	3	1.00	1180	
1	6414100192	20141209T000000	538000.0	3	2.25	2570	
2	5631500400	20150225T000000	180000.0	2	1.00	770	
3	2487200875	20141209T000000	604000.0	4	3.00	1960	
4	1954400510	20150218T000000	510000.0	3	2.00	1680	

	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basement	\
0	5650	1.0	0	0	...	7	1180	0	
1	7242	2.0	0	0	...	7	2170	400	
2	10000	1.0	0	0	...	6	770	0	
3	5000	1.0	0	0	...	7	1050	910	
4	8080	1.0	0	0	...	8	1680	0	

	yr_built	yr_renovated	zipcode	lat	long	sqft_living15	\
0	1955	0	98178	47.5112	-122.257	1340	
1	1951	1991	98125	47.7210	-122.319	1690	
2	1933	0	98028	47.7379	-122.233	2720	
3	1965	0	98136	47.5208	-122.393	1360	

```
4      1987             0    98074  47.6168 -122.045          1800

   sqft_lot15
0        5650
1        7639
2        8062
3        5000
4        7503

[5 rows x 21 columns]
```

In [4]:
```
# quantidade de Registros no Dataset
base.count()
```

Out[4]:
```
id               21613
date             21613
price            21613
bedrooms         21613
bathrooms        21613
sqft_living      21613
sqft_lot         21613
floors           21613
waterfront       21613
view             21613
condition        21613
grade            21613
sqft_above       21613
sqft_basement    21613
yr_built         21613
yr_renovated     21613
zipcode          21613
lat              21613
long             21613
sqft_living15    21613
sqft_lot15       21613
dtype: int64
```

In [5]:
```
base.shape
```

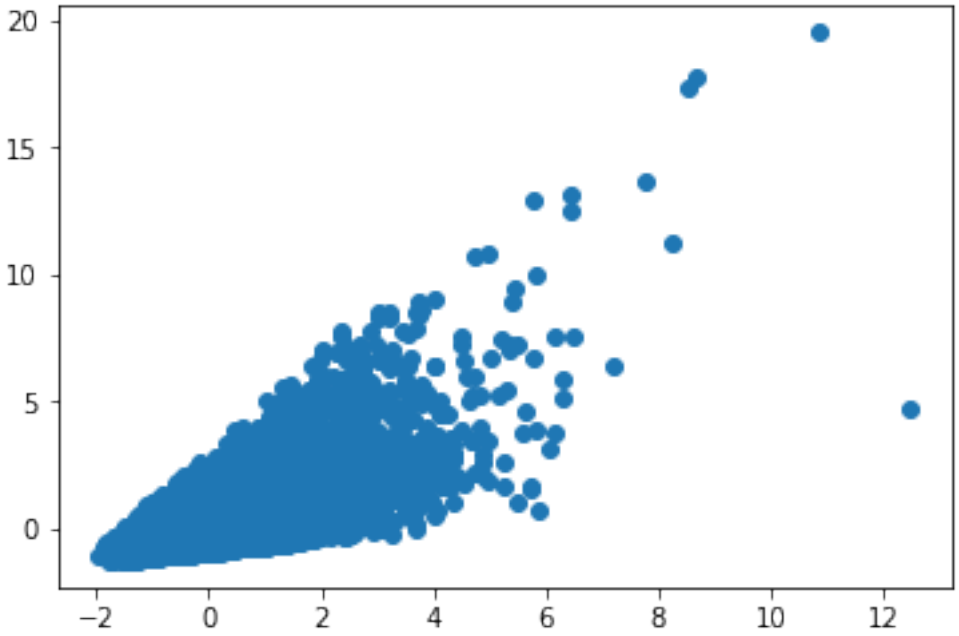
Out[5]:
```
(21613, 21)
```

In [13]:
```
# Análise da metragem da casa
x = base.iloc[:, 5].values
x = x.reshape(-1, 1)
```

In [14]:
```
x.shape
```

Out[14]:
```
(21613, 1)
```

```
In [15]: # Análise do preço da casa
         y = base.iloc[:, 2:3].values

In [16]: y.shape

Out[16]: (21613, 1)

In [17]: from sklearn.preprocessing import StandardScaler
         scaler_x = StandardScaler()
         x = scaler_x.fit_transform(x)
         scaler_y = StandardScaler()
         y = scaler_y.fit_transform(y)

C:\Users\rique\Anaconda3\lib\site-packages\sklearn\utils\validation.py:595: DataConversionWarni
  warnings.warn(msg, DataConversionWarning)
C:\Users\rique\Anaconda3\lib\site-packages\sklearn\utils\validation.py:595: DataConversionWarni
  warnings.warn(msg, DataConversionWarning)


In [18]: x

Out[18]: array([[-0.97983502],
                [ 0.53363434],
                [-1.42625404],
                ...,
                [-1.15404732],
                [-0.52252773],
                [-1.15404732]])

In [19]: y

Out[19]: array([[-0.86671733],
                [-0.00568792],
                [-0.98084935],
                ...,
                [-0.37586519],
                [-0.38158814],
                [-0.58588173]])

In [20]: #visualizando os Dados

         import matplotlib.pyplot as plt
         %matplotlib inline
         plt.scatter(x, y)

Out[20]: <matplotlib.collections.PathCollection at 0x281fdb47dd8>
```



#Fórmula da regressão Linear simples
y = b0 + b1 * x

In [21]:
```
import numpy as np
np.random.seed(1)
np.random.rand(2)
```

Out[21]:
```
array([0.417022  , 0.72032449])
```

In [22]:
```
# Começando com o TensorFlow, passando os dados de Pandas para Placeholders
import tensorflow as tf
```

In [23]:
```
# Construindo o Modelo

b0 = tf.Variable(0.41)
b1 = tf.Variable(0.72)
```

In [24]:
```
#Treinando o modelo/ Placeholders
# A coluna simbolizada com 1 refere-se ao preço da Casa

batch_size = 32
xph = tf.placeholder(tf.float32, [batch_size, 1])
yph = tf.placeholder(tf.float32, [batch_size, 1])
```

In [26]:
```
# Criando o Modelo

y_modelo = b0 + b1 * xph
```

```
        erro = tf.losses.mean_squared_error(yph, y_modelo)
        otimizador = tf.train.GradientDescentOptimizer(learning_rate = 0.001)
        treinamento = otimizador.minimize(erro)
        init = tf.global_variables_initializer()
```

```
WARNING:tensorflow:From C:\Users\rique\Anaconda3\lib\site-packages\tensorflow_core\python\ops\l
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
```

In [27]:
```
# Criando uma sessão

with tf.Session() as sess:
    sess.run(init)
    for i in range(10000):
        indices = np.random.randint(len(x), size = batch_size)
        feed = {xph: x [indices], yph: y[indices]}
        sess.run(treinamento, feed_dict = feed)
    b0_final, b1_final = sess.run([b0, b1])
```

In [28]:
```
b0_final
```

Out[28]:
```
-0.0030732849
```

In [29]:
```
b1_final
```

Out[29]:
```
0.69893813
```

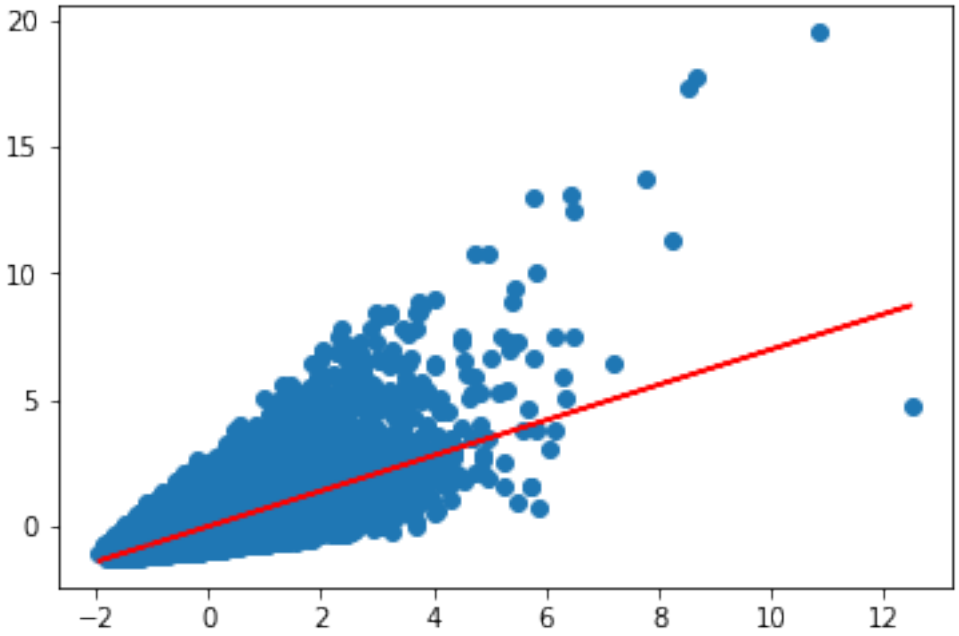
In [30]:
```
previsoes = b0_final + b1_final * x
```

In [31]:
```
previsoes
```

Out[31]:
```
array([[-0.68791734],
       [ 0.3699041 ],
       [-0.99993662],
       ...,
       [-0.80968096],
       [-0.36828784],
       [-0.80968096]])
```

In [32]:
```
# Plotando no gráfico com a linha de regressão

plt.plot(x, y, 'o')
plt.plot(x, previsoes, color = 'red')
```

Out[32]:
```
[<matplotlib.lines.Line2D at 0x28194e4a358>]
```




```
In [33]: # MAE - Mean Absolute Error
         y1 = scaler_y.inverse_transform(y)
         previsoes1 = scaler_y.inverse_transform(previsoes)

In [34]: y1

Out[34]: array([[221900.],
                [538000.],
                [180000.],
                ...,
                [402101.],
                [400000.],
                [325000.]])

In [35]: previsoes1

Out[35]: array([[287540.81878705],
                [675886.85664255],
                [172992.70690162],
                ...,
                [242839.11658786],
                [404882.78705994],
                [242839.11658786]])

In [36]: from sklearn.metrics import mean_absolute_error
         mae = mean_absolute_error(y1, previsoes1)
         mae

Out[36]: 173392.88203921868
```

2 O modelo errou nas previsões apresentando um erro grande em relação aos preços das casas

3 Para alcançar melhores resultados será necessário aplicar novas abordagens além da regressão linear simples

In [ ]:

In [ ]: