In [6]:

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

from sklearn import linear_model
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_boston

houses_boston = load_boston()
# Dados originais sobre mercado imobiliario de Boston (506 linhas x 13 atributo
s)
df_x = pd.DataFrame(houses_boston.data, columns=houses_boston.feature_names)
df_x.head()
```

Out[6]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	E
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	3
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	3
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	3
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	3
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	3

In [7]:

```
# Preços conforme os atributos acima
df_target = pd.DataFrame(houses_boston.target)
df_target.head()
```

Out[7]:

	0
0	24.0
1	21.6
2	34.7
3	33.4
4	36.2

In [10]:

```
# Atributos da base de dados
# ---------------------------
# CRIM = crime rate
# ZN = land zone
# INDUS = proportion of non retail business acres
# CHAS = Charles River Dummy variable
# NOX = Nitric Oxides Concentration
# RM  = average rooms
# AGE = owner occupied units
# DIS = weighted distances to five boston employment centers
# RAD = index of accessibility
# TAX = tax rate
# PTRATIO = pupil teacher ratio
# B = proportion of people of afracan american descent
# LSTAT = percentage lower status
# Price = housing prices
```

In [11]:

```
# DESAFIO
# Como analisar e descobrir quais atributos são melhores para treinar o modelo d
e forma a fazer a melhor predição?
# Campos com melhor relação, predição com melhor score!
df_x.describe()
```

Out[11]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AG
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	50
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.9
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	94
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	10

In [12]:

```
df_x.corr()
```

Out[12]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE
CRIM	1.000000	-0.200469	0.406583	-0.055892	0.420972	-0.219247	0.352734
ZN	-0.200469	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537
INDUS	0.406583	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779
CHAS	-0.055892	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518
NOX	0.420972	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470
RM	-0.219247	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265
AGE	0.352734	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000
DIS	-0.379670	0.664408	-0.708027	-0.099176	-0.769230	0.205246	-0.747881
RAD	0.625505	-0.311948	0.595129	-0.007368	0.611441	-0.209847	0.456022
TAX	0.582764	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456
PTRATIO	0.289946	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515
B	-0.385064	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534
LSTAT	0.455621	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339