PROJECT 2: REGRESSION

In this project, we analyse the Boston housing dataset. The goal is to compare the relative importance of the numerical factors to see which have greater influence on the median value of houses.

Data description

The data set has 506 rows and 14 features:

In [3]:
```
print(df.shape)
df.head()
```

```
(506, 14)
```

Out[3]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	medv
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2

The explanations of the feature columns are as follows:

CRIM: Per capita crime rate by town
ZN: Proportion of residential land zoned for lots over 25,000 sq. ft
INDUS: Proportion of non-retail business acres per town
CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
NOX: Nitric oxide concentration (parts per 10 million)
RM: Average number of rooms per dwelling
AGE: Proportion of owner-occupied units built prior to 1940
DIS: Weighted distances to five Boston employment centers
RAD: Index of accessibility to radial highways
TAX: Full-value property tax rate per $10,000
PTRATIO: Pupil-teacher ratio by town
B: $1000(Bk — 0.63)^2$, where Bk is the proportion of [people of African American descent] by town
LSTAT: Percentage of lower status of the population

MEDV is the target: the median value of houses (in thousands)

Exploration

We checked that there is no missing data

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 506 entries, 0 to 505
Data columns (total 14 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   crim     506 non-null    float64
 1   zn       506 non-null    float64
 2   indus    506 non-null    float64
 3   chas     506 non-null    int64  
 4   nox      506 non-null    float64
 5   rm       506 non-null    float64
 6   age      506 non-null    float64
 7   dis      506 non-null    float64
 8   rad      506 non-null    int64  
 9   tax      506 non-null    int64  
 10  ptratio  506 non-null    float64
 11  b        506 non-null    float64
 12  lstat    506 non-null    float64
 13  medv     506 non-null    float64
dtypes: float64(11), int64(3)
memory usage: 55.5 KB
```

Then we looked at the correlation between the columns and medv

```
In [15]:  df.corr()['medv'].sort_values()

Out[15]: lstat      -0.737663
         ptratio    -0.507787
         indus      -0.483725
         tax        -0.468536
         nox        -0.427321
         crim       -0.388305
         rad        -0.381626
         age        -0.376955
         chas        0.175260
         dis         0.249929
         b           0.333461
         zn          0.360445
         rm          0.695360
         medv        1.000000
         Name: medv, dtype: float64
```

We observe that rm has highest positive correlation, lstat has highest negative correlation, and chas has correlation close to 0. We'll perform 2 analyses, one with just these features, and one with all the features.

```
Xall = df.drop('medv',1)
Xsome = df[['lstat','chas','rm']]
y = df['medv']
```

To improve interpretability, we do not scale the features.

Linear regression models

We train the two linear regression models and note their coefficients

```
from sklearn.linear_model import LinearRegression

lr1 = LinearRegression()
lr1.fit(Xall,y)

lr2 = LinearRegression()
lr2.fit(Xsome,y)
```

```
LinearRegression()
```

```
vars = ['lstat','chas','rm']

pd.DataFrame({'Model 1 coefficients': lr1.coef_[[12, 3, 5]],
              'Model 2 coefficients': lr2.coef_}, index = vars)
```

	Model 1 coefficients	Model 2 coefficients
lstat	-0.524758	-0.642848
chas	2.686734	4.120479
rm	3.809865	4.955812

We compare only the coefficients for the three features of interest.

Conclusion

We note that the two models broadly agree as to the dependence of medv on these variables: medv is negatively correlated with lstat and positively correlated with both chas and rm. The coefficients corresponding to lstat are also quite similar, which suggests that there are unlikely to be confounding variables between lstat and medv not considered here.

We note that the coefficients for chas and rm differ a bit between the models. All factors considered, a linear model suggests that the median value of a house increases by 2.7 thousand if it is closer to Charles River and increases by 3.8 thousand if it has one more room. When only these three factors are considered, a linear model suggests that the median value of a house increases by 4.1 thousand if it is closer to Charles River and increases by 5.0 thousand if it has one more room. This suggests the presence of confounds, which might be worth exploring.

Another point worth noting is that the model 2 coefficients for chas and rm are quite similar, which might be surprising given that the (all-factors-considered) correlation between rm and medv is much higher than that between chas and medv.