In [247…
```
import pandas as pd
import numpy as np
```

In [249…
```
df = pd.read_csv("/home/admin1/Downloads/BostonHousing (3).csv")
df
```

Out[249…

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	ls
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5
...	...	...	...	...	...	...	...	...	...	...	...	...	
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273	21.0	391.99	9
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90	9
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45	6
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273	21.0	396.90	7

506 rows × 14 columns

In [251…
```
df.describe()
```

Out[251…

	crim	zn	indus	chas	nox	rm	a
count	506.000000	506.000000	506.000000	506.000000	506.000000	501.000000	506.000(
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284341	68.574(
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.705587	28.148(
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900(
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.884000	45.025(
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208000	77.500(
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.625000	94.075(
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000(

In [253…
```
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       501 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
```

In [255…
```
df.dropna(inplace = True)
```

In [257…
```
x = df.drop(columns = ['medv'])
y = df['medv']
```

In [259…
```
x
```

Out[259…

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	ls
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5
...	...	...	...	...	...	...	...	...	...	...	...	...	
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273	21.0	391.99	9
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90	9
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45	6
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273	21.0	396.90	7

501 rows × 13 columns

In [261…
```
y
```

Out[261…
```
0      24.0
1      21.6
2      34.7
3      33.4
4      36.2
       ...
501    22.4
502    20.6
503    23.9
504    22.0
505    11.9
Name: medv, Length: 501, dtype: float64
```

In [263…
```
from sklearn.model_selection import train_test_split
```

In [378…
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)
```

In [380…
```
from sklearn.linear_model import LinearRegression
```

In [382…
```
LR = LinearRegression()

LR.fit(x_train, y_train)
```

Out[382…
▾ LinearRegression

LinearRegression()

In [384…
```
y_pred = LR.predict(x_test)
```

In [386…
```
y_pred
```

Out[386…
```
array([17.55024241, 36.69868483, 36.83593832, 28.91444614, 19.30660719,
       25.09469614, 29.50320998, 15.6784396 , 12.26957839, 21.21984009,
       23.34250624, 13.43080574, 13.12136922, 37.25294764, 22.64990136,
       16.04657012, 27.57120912, 15.6647531 , 16.14548717, 12.95503092,
       27.04407439, 24.13088354, 24.69785809, 25.1864416 , 30.95072393,
       20.72393034, 19.44207148, 23.66332237, 20.57547046, 12.87883835,
       24.71104271, 13.04431616, 23.79421395, 27.71115036, 24.68885797,
       22.59030373, 17.31530608,  6.79374741, 13.68307385, 26.11249035,
       30.28898152, 25.89162543, 42.28421197, 20.53007249, 20.56449961,
       42.24948464, 15.76117744, 22.98581497, 12.27467514, 18.76755199,
       38.39764948, 19.5223973 , 17.65979565, 27.5039948 , 20.96978157,
       27.79746959, 18.16106061, 20.92569859, 37.59339549, 15.44839744,
       15.65469141, 13.74821633, 21.37118256, 26.06208549, 19.85756396,
        8.03926181, 33.37776007, 10.71832487, 35.51977422, 18.44587131,
       19.08047192, 23.42183085, 36.26166784, 28.63564993, 28.51610572,
       27.98389578, 10.3797045 , 23.95455056, 27.19809954, 28.51254532,
       19.87587612, 20.4671337 , 21.81396578, 31.1348068 , 18.17848536,
       15.23649566, 22.52257776, 36.36627231, 20.81900418, 14.62534743,
       22.14066611, 21.66551337, 25.05700294, 21.60370201, 18.07332864,
       20.85605926, 21.26237237, 13.78135961, 14.76041694, 33.22764118,
       18.21115347])
```

In [388…
```
LR.score(x_test, y_test)
```

Out[388…
```
0.6384679227914916
```

In [390…
```
LR.coef_
```

Out[390…
```
array([-9.79066779e-02,  3.96169071e-02,  1.04207341e-02,  3.15116912e+00,
       -1.59502630e+01,  4.60395187e+00, -2.23025532e-02, -1.53147842e+00,
        2.81333530e-01, -1.33272907e-02, -9.33347726e-01,  8.90217627e-03,
       -3.99171771e-01])
```

In [392…
```
LR.intercept_
```

Out[392…
```
31.126447405446584
```

In [394…
```
from sklearn import metrics
```

In [407…
```
MAE = metrics.mean_absolute_error(y_test, y_pred)
MSE = metrics.mean_squared_error(y_test, y_pred)
RMSE = np.sqrt(metrics.mean_squared_error(y_test, y_pred))
```

In [398…
```
MAE
```

Out[398…
```
3.6617830498047734
```

In [400…
```
MSE
```

Out[400…
```
32.715235780930996
```

In [409…
```
RMSE
```

Out[409…
```
5.71972340073635
```