

# Lab\_5\_python\_B4

December 8, 2022

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

[2]: column_names = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS',
↳ 'RAD', 'TAX', 'PTRATIO', 'B', 'LSTAT', 'MEDV']

[3]: housing = pd.read_csv('/home/spark/lab05/housing.csv', header=None,
↳ delimiter=r"\s+", names=column_names)

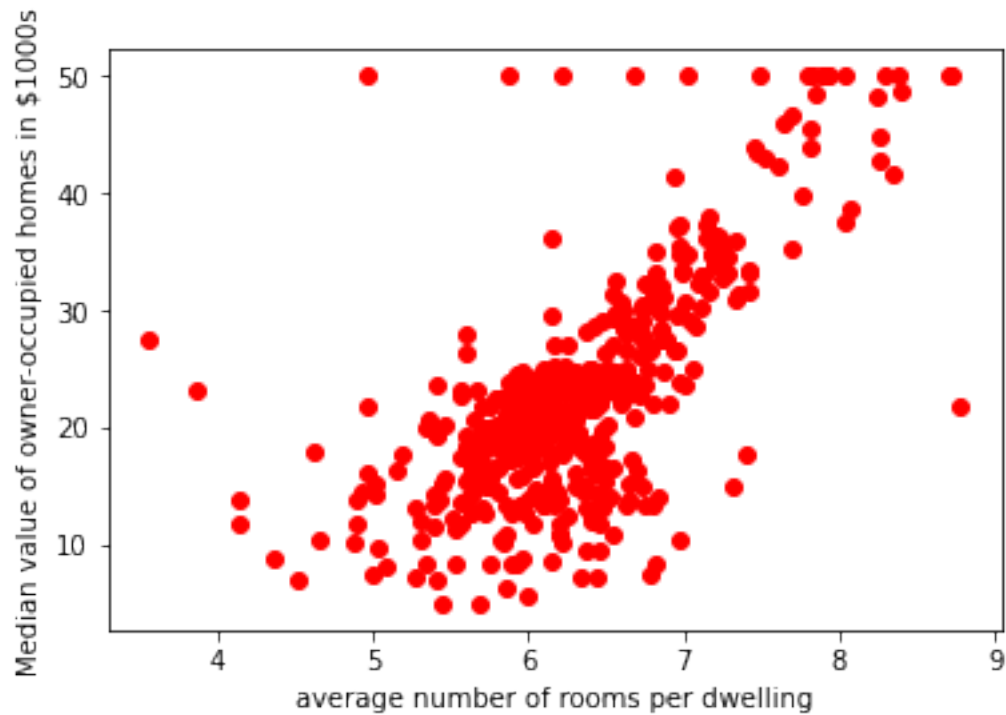
[4]: housing.head()
```

```
[4]:      CRIM      ZN  INDUS  CHAS      NOX      RM      AGE      DIS  RAD      TAX  \
0  0.00632  18.0    2.31     0  0.538  6.575  65.2  4.0900    1  296.0
1  0.02731   0.0    7.07     0  0.469  6.421  78.9  4.9671    2  242.0
2  0.02729   0.0    7.07     0  0.469  7.185  61.1  4.9671    2  242.0
3  0.03237   0.0    2.18     0  0.458  6.998  45.8  6.0622    3  222.0
4  0.06905   0.0    2.18     0  0.458  7.147  54.2  6.0622    3  222.0

      PTRATIO      B  LSTAT  MEDV
0      15.3  396.90   4.98  24.0
1      17.8  396.90   9.14  21.6
2      17.8  392.83   4.03  34.7
3      18.7  394.63   2.94  33.4
4      18.7  396.90   5.33  36.2
```

```
[5]: plt.scatter(x=housing['RM'], y=housing['MEDV'], color='red')
plt.xlabel('average number of rooms per dwelling')
plt.ylabel('Median value of owner-occupied homes in $1000s')
```

```
[5]: Text(0, 0.5, 'Median value of owner-occupied homes in $1000s')
```



```
[6]: x=pd.DataFrame(housing['RM'])
     y=pd.DataFrame(housing['MEDV'])
```

```
[7]: from sklearn.model_selection import train_test_split
     x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20)
```

```
[8]: from sklearn.tree import DecisionTreeRegressor
```

```
[9]: regressor = DecisionTreeRegressor(criterion='mse',random_state=100,max_depth=4,
    ↪min_samples_leaf=1)
```

```
[10]: regressor.fit(x_train,y_train)
```

```
[10]: DecisionTreeRegressor(max_depth=4, random_state=100)
```

```
[11]: from sklearn.tree import export_graphviz
     import pydotplus
```

```
[12]: export_graphviz (regressor, out_file='reg_tree.dot')
```

```
[13]: y_pred = regressor.predict(x_test)
```

```
[14]: print(y_pred[4:9])
      print(y_test[4:9])
```

```
[19.83652968 19.83652968 19.83652968 19.83652968 19.83652968]
      MEDV
215    25.0
83     22.9
491    13.6
425     8.3
145    13.8
```

```
[15]: from sklearn.metrics import mean_squared_error
      mse = mean_squared_error(y_pred,y_test)
      rmse = np.sqrt(mse)
      rmse
```

```
[15]: 7.512962059971756
```

```
[ ]:
```