{

"nbformat": 4,

"nbformat\_minor": 0,

"metadata": {

"colab": {

"name": "Vaishali V",

"provenance": [],

"collapsed\_sections": []

},

"kernelspec": {

"name": "python3",

"display\_name": "Python 3"

},

"language\_info": {

"name": "python"

}

},

"cells": [

{

"cell\_type": "code",

"metadata": {

"id": "HPcpZoJ4eqmm"

},

"source": [

"import numpy as np\n",

"import pandas as pd\n",

"import sklearn"

],

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"id": "LAAeeno8gm5G"

},

"source": [

"from sklearn.datasets import load\_boston\n",

"df=load\_boston()"

],

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "h3WkQGMjilhZ",

"outputId": "fcf35edd-9734-45c1-c9d0-f5beca6734a3"

},

"source": [

"df.keys() #return all the keys of the dataset dictionary"

],

"execution\_count": null,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"dict\_keys(['data', 'target', 'feature\_names', 'DESCR', 'filename'])"

]

},

"metadata": {

"tags": []

},

"execution\_count": 7

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "tLCU0TpZmyi8",

"outputId": "a0e67021-6ebd-4c55-d719-ce16b83321ce"

},

"source": [

"print(df.DESCR) #info about the datset"

],

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"text": [

".. \_boston\_dataset:\n",

"\n",

"Boston house prices dataset\n",

"---------------------------\n",

"\n",

"\*\*Data Set Characteristics:\*\* \n",

"\n",

" :Number of Instances: 506 \n",

"\n",

" :Number of Attributes: 13 numeric/categorical predictive. Median Value (attribute 14) is usually the target.\n",

"\n",

" :Attribute Information (in order):\n",

" - CRIM per capita crime rate by town\n",

" - ZN proportion of residential land zoned for lots over 25,000 sq.ft.\n",

" - INDUS proportion of non-retail business acres per town\n",

" - CHAS Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)\n",

" - NOX nitric oxides concentration (parts per 10 million)\n",

" - RM average number of rooms per dwelling\n",

" - AGE proportion of owner-occupied units built prior to 1940\n",

" - DIS weighted distances to five Boston employment centres\n",

" - RAD index of accessibility to radial highways\n",

" - TAX full-value property-tax rate per $10,000\n",

" - PTRATIO pupil-teacher ratio by town\n",

" - B 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town\n",

" - LSTAT % lower status of the population\n",

" - MEDV Median value of owner-occupied homes in $1000's\n",

"\n",

" :Missing Attribute Values: None\n",

"\n",

" :Creator: Harrison, D. and Rubinfeld, D.L.\n",

"\n",

"This is a copy of UCI ML housing dataset.\n",

"https://archive.ics.uci.edu/ml/machine-learning-databases/housing/\n",

"\n",

"\n",

"This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University.\n",

"\n",

"The Boston house-price data of Harrison, D. and Rubinfeld, D.L. 'Hedonic\n",

"prices and the demand for clean air', J. Environ. Economics & Management,\n",

"vol.5, 81-102, 1978. Used in Belsley, Kuh & Welsch, 'Regression diagnostics\n",

"...', Wiley, 1980. N.B. Various transformations are used in the table on\n",

"pages 244-261 of the latter.\n",

"\n",

"The Boston house-price data has been used in many machine learning papers that address regression\n",

"problems. \n",

" \n",

".. topic:: References\n",

"\n",

" - Belsley, Kuh & Welsch, 'Regression diagnostics: Identifying Influential Data and Sources of Collinearity', Wiley, 1980. 244-261.\n",

" - Quinlan,R. (1993). Combining Instance-Based and Model-Based Learning. In Proceedings on the Tenth International Conference of Machine Learning, 236-243, University of Massachusetts, Amherst. Morgan Kaufmann.\n",

"\n"

],

"name": "stdout"

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "Ezv2ch0pn8Hg",

"outputId": "4c5bd9ab-16f6-40e2-c429-ad63e99a6390"

},

"source": [

" print(df.feature\_names) #info about the datset"

],

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"text": [

"['CRIM' 'ZN' 'INDUS' 'CHAS' 'NOX' 'RM' 'AGE' 'DIS' 'RAD' 'TAX' 'PTRATIO'\n",

" 'B' 'LSTAT']\n"

],

"name": "stdout"

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "sGLlELqEoZSM",

"outputId": "aa5d15d5-f004-412a-ff91-2da80c700dfc"

},

"source": [

" print(df.filename) #info about the datset"

],

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"text": [

"/usr/local/lib/python3.7/dist-packages/sklearn/datasets/data/boston\_house\_prices.csv\n"

],

"name": "stdout"

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "cUX\_3MmOouyo",

"outputId": "f878cf63-d46d-4ae1-ad62-095fb454cbc5"

},

"source": [

" print(df.target) #info about the datset"

],

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"text": [

"[24. 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 15. 18.9 21.7 20.4\n",

" 18.2 19.9 23.1 17.5 20.2 18.2 13.6 19.6 15.2 14.5 15.6 13.9 16.6 14.8\n",

" 18.4 21. 12.7 14.5 13.2 13.1 13.5 18.9 20. 21. 24.7 30.8 34.9 26.6\n",

" 25.3 24.7 21.2 19.3 20. 16.6 14.4 19.4 19.7 20.5 25. 23.4 18.9 35.4\n",

" 24.7 31.6 23.3 19.6 18.7 16. 22.2 25. 33. 23.5 19.4 22. 17.4 20.9\n",

" 24.2 21.7 22.8 23.4 24.1 21.4 20. 20.8 21.2 20.3 28. 23.9 24.8 22.9\n",

" 23.9 26.6 22.5 22.2 23.6 28.7 22.6 22. 22.9 25. 20.6 28.4 21.4 38.7\n",

" 43.8 33.2 27.5 26.5 18.6 19.3 20.1 19.5 19.5 20.4 19.8 19.4 21.7 22.8\n",

" 18.8 18.7 18.5 18.3 21.2 19.2 20.4 19.3 22. 20.3 20.5 17.3 18.8 21.4\n",

" 15.7 16.2 18. 14.3 19.2 19.6 23. 18.4 15.6 18.1 17.4 17.1 13.3 17.8\n",

" 14. 14.4 13.4 15.6 11.8 13.8 15.6 14.6 17.8 15.4 21.5 19.6 15.3 19.4\n",

" 17. 15.6 13.1 41.3 24.3 23.3 27. 50. 50. 50. 22.7 25. 50. 23.8\n",

" 23.8 22.3 17.4 19.1 23.1 23.6 22.6 29.4 23.2 24.6 29.9 37.2 39.8 36.2\n",

" 37.9 32.5 26.4 29.6 50. 32. 29.8 34.9 37. 30.5 36.4 31.1 29.1 50.\n",

" 33.3 30.3 34.6 34.9 32.9 24.1 42.3 48.5 50. 22.6 24.4 22.5 24.4 20.\n",

" 21.7 19.3 22.4 28.1 23.7 25. 23.3 28.7 21.5 23. 26.7 21.7 27.5 30.1\n",

" 44.8 50. 37.6 31.6 46.7 31.5 24.3 31.7 41.7 48.3 29. 24. 25.1 31.5\n",

" 23.7 23.3 22. 20.1 22.2 23.7 17.6 18.5 24.3 20.5 24.5 26.2 24.4 24.8\n",

" 29.6 42.8 21.9 20.9 44. 50. 36. 30.1 33.8 43.1 48.8 31. 36.5 22.8\n",

" 30.7 50. 43.5 20.7 21.1 25.2 24.4 35.2 32.4 32. 33.2 33.1 29.1 35.1\n",

" 45.4 35.4 46. 50. 32.2 22. 20.1 23.2 22.3 24.8 28.5 37.3 27.9 23.9\n",

" 21.7 28.6 27.1 20.3 22.5 29. 24.8 22. 26.4 33.1 36.1 28.4 33.4 28.2\n",

" 22.8 20.3 16.1 22.1 19.4 21.6 23.8 16.2 17.8 19.8 23.1 21. 23.8 23.1\n",

" 20.4 18.5 25. 24.6 23. 22.2 19.3 22.6 19.8 17.1 19.4 22.2 20.7 21.1\n",

" 19.5 18.5 20.6 19. 18.7 32.7 16.5 23.9 31.2 17.5 17.2 23.1 24.5 26.6\n",

" 22.9 24.1 18.6 30.1 18.2 20.6 17.8 21.7 22.7 22.6 25. 19.9 20.8 16.8\n",

" 21.9 27.5 21.9 23.1 50. 50. 50. 50. 50. 13.8 13.8 15. 13.9 13.3\n",

" 13.1 10.2 10.4 10.9 11.3 12.3 8.8 7.2 10.5 7.4 10.2 11.5 15.1 23.2\n",

" 9.7 13.8 12.7 13.1 12.5 8.5 5. 6.3 5.6 7.2 12.1 8.3 8.5 5.\n",

" 11.9 27.9 17.2 27.5 15. 17.2 17.9 16.3 7. 7.2 7.5 10.4 8.8 8.4\n",

" 16.7 14.2 20.8 13.4 11.7 8.3 10.2 10.9 11. 9.5 14.5 14.1 16.1 14.3\n",

" 11.7 13.4 9.6 8.7 8.4 12.8 10.5 17.1 18.4 15.4 10.8 11.8 14.9 12.6\n",

" 14.1 13. 13.4 15.2 16.1 17.8 14.9 14.1 12.7 13.5 14.9 20. 16.4 17.7\n",

" 19.5 20.2 21.4 19.9 19. 19.1 19.1 20.1 19.9 19.6 23.2 29.8 13.8 13.3\n",

" 16.7 12. 14.6 21.4 23. 23.7 25. 21.8 20.6 21.2 19.1 20.6 15.2 7.\n",

" 8.1 13.6 20.1 21.8 24.5 23.1 19.7 18.3 21.2 17.5 16.8 22.4 20.6 23.9\n",

" 22. 11.9]\n"

],

"name": "stdout"

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "\_66q3rbeo7NK",

"outputId": "1b3006cf-f0a3-4ced-8198-62261da6232b"

},

"source": [

" print(df.data) #info about the datset"

],

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"text": [

"[[6.3200e-03 1.8000e+01 2.3100e+00 ... 1.5300e+01 3.9690e+02 4.9800e+00]\n",

" [2.7310e-02 0.0000e+00 7.0700e+00 ... 1.7800e+01 3.9690e+02 9.1400e+00]\n",

" [2.7290e-02 0.0000e+00 7.0700e+00 ... 1.7800e+01 3.9283e+02 4.0300e+00]\n",

" ...\n",

" [6.0760e-02 0.0000e+00 1.1930e+01 ... 2.1000e+01 3.9690e+02 5.6400e+00]\n",

" [1.0959e-01 0.0000e+00 1.1930e+01 ... 2.1000e+01 3.9345e+02 6.4800e+00]\n",

" [4.7410e-02 0.0000e+00 1.1930e+01 ... 2.1000e+01 3.9690e+02 7.8800e+00]]\n"

],

"name": "stdout"

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 226

},

"id": "kVy7OacXpi8D",

"outputId": "d47c6b7d-6997-4827-f331-2755bbd89978"

},

"source": [

"boston = pd.DataFrame(df.data,columns=df.feature\_names)\n",

"boston.head()"

],

"execution\_count": null,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>CRIM</th>\n",

" <th>ZN</th>\n",

" <th>INDUS</th>\n",

" <th>CHAS</th>\n",

" <th>NOX</th>\n",

" <th>RM</th>\n",

" <th>AGE</th>\n",

" <th>DIS</th>\n",

" <th>RAD</th>\n",

" <th>TAX</th>\n",

" <th>PTRATIO</th>\n",

" <th>B</th>\n",

" <th>LSTAT</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>0.00632</td>\n",

" <td>18.0</td>\n",

" <td>2.31</td>\n",

" <td>0.0</td>\n",

" <td>0.538</td>\n",

" <td>6.575</td>\n",

" <td>65.2</td>\n",

" <td>4.0900</td>\n",

" <td>1.0</td>\n",

" <td>296.0</td>\n",

" <td>15.3</td>\n",

" <td>396.90</td>\n",

" <td>4.98</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>0.02731</td>\n",

" <td>0.0</td>\n",

" <td>7.07</td>\n",

" <td>0.0</td>\n",

" <td>0.469</td>\n",

" <td>6.421</td>\n",

" <td>78.9</td>\n",

" <td>4.9671</td>\n",

" <td>2.0</td>\n",

" <td>242.0</td>\n",

" <td>17.8</td>\n",

" <td>396.90</td>\n",

" <td>9.14</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>0.02729</td>\n",

" <td>0.0</td>\n",

" <td>7.07</td>\n",

" <td>0.0</td>\n",

" <td>0.469</td>\n",

" <td>7.185</td>\n",

" <td>61.1</td>\n",

" <td>4.9671</td>\n",

" <td>2.0</td>\n",

" <td>242.0</td>\n",

" <td>17.8</td>\n",

" <td>392.83</td>\n",

" <td>4.03</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>0.03237</td>\n",

" <td>0.0</td>\n",

" <td>2.18</td>\n",

" <td>0.0</td>\n",

" <td>0.458</td>\n",

" <td>6.998</td>\n",

" <td>45.8</td>\n",

" <td>6.0622</td>\n",

" <td>3.0</td>\n",

" <td>222.0</td>\n",

" <td>18.7</td>\n",

" <td>394.63</td>\n",

" <td>2.94</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>0.06905</td>\n",

" <td>0.0</td>\n",

" <td>2.18</td>\n",

" <td>0.0</td>\n",

" <td>0.458</td>\n",

" <td>7.147</td>\n",

" <td>54.2</td>\n",

" <td>6.0622</td>\n",

" <td>3.0</td>\n",

" <td>222.0</td>\n",

" <td>18.7</td>\n",

" <td>396.90</td>\n",

" <td>5.33</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" CRIM ZN INDUS CHAS NOX ... RAD TAX PTRATIO B LSTAT\n",

"0 0.00632 18.0 2.31 0.0 0.538 ... 1.0 296.0 15.3 396.90 4.98\n",

"1 0.02731 0.0 7.07 0.0 0.469 ... 2.0 242.0 17.8 396.90 9.14\n",

"2 0.02729 0.0 7.07 0.0 0.469 ... 2.0 242.0 17.8 392.83 4.03\n",

"3 0.03237 0.0 2.18 0.0 0.458 ... 3.0 222.0 18.7 394.63 2.94\n",

"4 0.06905 0.0 2.18 0.0 0.458 ... 3.0 222.0 18.7 396.90 5.33\n",

"\n",

"[5 rows x 13 columns]"

]

},

"metadata": {

"tags": []

},

"execution\_count": 21

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 226

},

"id": "UX0gV32Cx5mU",

"outputId": "46d44f77-5e70-407f-d378-860e3b79ca8c"

},

"source": [

"boston['MEDV']=df.target\n",

"boston.head()"

],

"execution\_count": null,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>CRIM</th>\n",

" <th>ZN</th>\n",

" <th>INDUS</th>\n",

" <th>CHAS</th>\n",

" <th>NOX</th>\n",

" <th>RM</th>\n",

" <th>AGE</th>\n",

" <th>DIS</th>\n",

" <th>RAD</th>\n",

" <th>TAX</th>\n",

" <th>PTRATIO</th>\n",

" <th>B</th>\n",

" <th>LSTAT</th>\n",

" <th>MEDV</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>0.00632</td>\n",

" <td>18.0</td>\n",

" <td>2.31</td>\n",

" <td>0.0</td>\n",

" <td>0.538</td>\n",

" <td>6.575</td>\n",

" <td>65.2</td>\n",

" <td>4.0900</td>\n",

" <td>1.0</td>\n",

" <td>296.0</td>\n",

" <td>15.3</td>\n",

" <td>396.90</td>\n",

" <td>4.98</td>\n",

" <td>24.0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>0.02731</td>\n",

" <td>0.0</td>\n",

" <td>7.07</td>\n",

" <td>0.0</td>\n",

" <td>0.469</td>\n",

" <td>6.421</td>\n",

" <td>78.9</td>\n",

" <td>4.9671</td>\n",

" <td>2.0</td>\n",

" <td>242.0</td>\n",

" <td>17.8</td>\n",

" <td>396.90</td>\n",

" <td>9.14</td>\n",

" <td>21.6</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>0.02729</td>\n",

" <td>0.0</td>\n",

" <td>7.07</td>\n",

" <td>0.0</td>\n",

" <td>0.469</td>\n",

" <td>7.185</td>\n",

" <td>61.1</td>\n",

" <td>4.9671</td>\n",

" <td>2.0</td>\n",

" <td>242.0</td>\n",

" <td>17.8</td>\n",

" <td>392.83</td>\n",

" <td>4.03</td>\n",

" <td>34.7</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>0.03237</td>\n",

" <td>0.0</td>\n",

" <td>2.18</td>\n",

" <td>0.0</td>\n",

" <td>0.458</td>\n",

" <td>6.998</td>\n",

" <td>45.8</td>\n",

" <td>6.0622</td>\n",

" <td>3.0</td>\n",

" <td>222.0</td>\n",

" <td>18.7</td>\n",

" <td>394.63</td>\n",

" <td>2.94</td>\n",

" <td>33.4</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>0.06905</td>\n",

" <td>0.0</td>\n",

" <td>2.18</td>\n",

" <td>0.0</td>\n",

" <td>0.458</td>\n",

" <td>7.147</td>\n",

" <td>54.2</td>\n",

" <td>6.0622</td>\n",

" <td>3.0</td>\n",

" <td>222.0</td>\n",

" <td>18.7</td>\n",

" <td>396.90</td>\n",

" <td>5.33</td>\n",

" <td>36.2</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" CRIM ZN INDUS CHAS NOX ... TAX PTRATIO B LSTAT MEDV\n",

"0 0.00632 18.0 2.31 0.0 0.538 ... 296.0 15.3 396.90 4.98 24.0\n",

"1 0.02731 0.0 7.07 0.0 0.469 ... 242.0 17.8 396.90 9.14 21.6\n",

"2 0.02729 0.0 7.07 0.0 0.469 ... 242.0 17.8 392.83 4.03 34.7\n",

"3 0.03237 0.0 2.18 0.0 0.458 ... 222.0 18.7 394.63 2.94 33.4\n",

"4 0.06905 0.0 2.18 0.0 0.458 ... 222.0 18.7 396.90 5.33 36.2\n",

"\n",

"[5 rows x 14 columns]"

]

},

"metadata": {

"tags": []

},

"execution\_count": 22

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 443

},

"id": "mnQPnStDyziw",

"outputId": "a690f0f7-9b38-48ab-edf1-da76fbc516d5"

},

"source": [

" boston.isnull()"

],

"execution\_count": null,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>CRIM</th>\n",

" <th>ZN</th>\n",

" <th>INDUS</th>\n",

" <th>CHAS</th>\n",

" <th>NOX</th>\n",

" <th>RM</th>\n",

" <th>AGE</th>\n",

" <th>DIS</th>\n",

" <th>RAD</th>\n",

" <th>TAX</th>\n",

" <th>PTRATIO</th>\n",

" <th>B</th>\n",

" <th>LSTAT</th>\n",

" <th>MEDV</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>...</th>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" <td>...</td>\n",

" </tr>\n",

" <tr>\n",

" <th>501</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>502</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>503</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>504</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" <tr>\n",

" <th>505</th>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" <td>False</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>506 rows נ14 columns</p>\n",

"</div>"

],

"text/plain": [

" CRIM ZN INDUS CHAS NOX ... TAX PTRATIO B LSTAT MEDV\n",

"0 False False False False False ... False False False False False\n",

"1 False False False False False ... False False False False False\n",

"2 False False False False False ... False False False False False\n",

"3 False False False False False ... False False False False False\n",

"4 False False False False False ... False False False False False\n",

".. ... ... ... ... ... ... ... ... ... ... ...\n",

"501 False False False False False ... False False False False False\n",

"502 False False False False False ... False False False False False\n",

"503 False False False False False ... False False False False False\n",

"504 False False False False False ... False False False False False\n",

"505 False False False False False ... False False False False False\n",

"\n",

"[506 rows x 14 columns]"

]

},

"metadata": {

"tags": []

},

"execution\_count": 23

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "rpvTqAvjzN6w",

"outputId": "7ddcb9a2-f107-4f83-a4ed-803f728f7330"

},

"source": [

" boston.isnull().sum()"

],

"execution\_count": null,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"CRIM 0\n",

"ZN 0\n",

"INDUS 0\n",

"CHAS 0\n",

"NOX 0\n",

"RM 0\n",

"AGE 0\n",

"DIS 0\n",

"RAD 0\n",

"TAX 0\n",

"PTRATIO 0\n",

"B 0\n",

"LSTAT 0\n",

"MEDV 0\n",

"dtype: int64"

]

},

"metadata": {

"tags": []

},

"execution\_count": 24

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "HUHfXcUD0A2K",

"outputId": "61a88ae9-8265-4bce-82a8-39d03b0f2ca2"

},

"source": [

"from sklearn.model\_selection import train\_test\_split \n",

"X = boston.drop( 'MEDV', axis=1)\n",

"Y = boston ['MEDV'] \n",

"X\_train, X\_test, Y\_train, Y\_test= train\_test\_split(X, Y, test\_size=0.15, random\_state=5)\n",

"print(X\_train.shape)\n",

"print (X\_test.shape) \n",

"print (Y\_train.shape)\n",

"print (Y\_test.shape)"

],

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"text": [

"(430, 13)\n",

"(76, 13)\n",

"(430,)\n",

"(76,)\n"

],

"name": "stdout"

}

]

},

{

"cell\_type": "code",

"metadata": {

"id": "2969Hmi64rOi"

},

"source": [

" from sklearn.linear\_model import LinearRegression \n",

"from sklearn.metrics import mean\_squared\_error"

],

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "gWS3e2B45BWc",

"outputId": "a0a9e6de-481b-4dcf-df4a-c7076985d841"

},

"source": [

" \n",

"## FITTING MODEL ON THE TRAINING DATASET\n",

" \n",

"lin\_model= LinearRegression()\n",

" \n",

"lin\_model.fit(X\_train, Y\_train)"

],

"execution\_count": null,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"LinearRegression(copy\_X=True, fit\_intercept=True, n\_jobs=None, normalize=False)"

]

},

"metadata": {

"tags": []

},

"execution\_count": 32

}

]

},

{

"cell\_type": "code",

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "OZJGNx1Q5-Q\_",

"outputId": "bb91563d-ad8d-4f02-b6f1-fe8d5cce45fb"

},

"source": [

"y\_train\_predict = lin\_model.predict(X\_train) \n",

"rmse = (np.sqrt (mean\_squared\_error(Y\_train, y\_train\_predict)))\n",

" \n",

"print(\"The model performance for training set\")\n",

"print('RMSE is {}'.format(rmse)) \n",

"print(\"\\n\")\n",

" \n",

"# on testing set\n",

" \n",

"y\_test\_predict = lin\_model.predict (X\_test) \n",

"rmse= (np.sqrt (mean\_squared\_error(Y\_test, y\_test\_predict)))\n",

" \n",

"print(\"The model performance for testing set\")\n",

"print('RMSE is {}'.format(rmse))"

],

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"text": [

"The model performance for training set\n",

"RMSE is 4.710901797319796\n",

"\n",

"\n",

"The model performance for testing set\n",

"RMSE is 4.687543527902972\n"

],

"name": "stdout"

}

]

}

]

}