linear-regression-boston

April 23, 2024

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

0.0.1 Load Dataset

[8]:
```python
df = pd.read_csv("boston.csv")
df.head(10)
```

[8]:
```
      CRIM    ZN  INDUS  CHAS    NOX     RM    AGE     DIS  RAD  TAX  PTRATIO  \
0  0.00632  18.0   2.31     0  0.538  6.575   65.2  4.0900    1  296     15.3
1  0.02731   0.0   7.07     0  0.469  6.421   78.9  4.9671    2  242     17.8
2  0.02729   0.0   7.07     0  0.469  7.185   61.1  4.9671    2  242     17.8
3  0.03237   0.0   2.18     0  0.458  6.998   45.8  6.0622    3  222     18.7
4  0.06905   0.0   2.18     0  0.458  7.147   54.2  6.0622    3  222     18.7
5  0.02985   0.0   2.18     0  0.458  6.430   58.7  6.0622    3  222     18.7
6  0.08829  12.5   7.87     0  0.524  6.012   66.6  5.5605    5  311     15.2
7  0.14455  12.5   7.87     0  0.524  6.172   96.1  5.9505    5  311     15.2
8  0.21124  12.5   7.87     0  0.524  5.631  100.0  6.0821    5  311     15.2
9  0.17004  12.5   7.87     0  0.524  6.004   85.9  6.5921    5  311     15.2

        B  LSTAT  MEDV  CAT. MEDV  Unnamed: 15  Unnamed: 16
0  396.90   4.98  24.0          0          NaN          NaN
1  396.90   9.14  21.6          0          NaN          NaN
2  392.83   4.03  34.7          1          NaN          NaN
3  394.63   2.94  33.4          1          NaN          NaN
4  396.90   5.33  36.2          1          NaN          NaN
5  394.12   5.21  28.7          0          NaN          NaN
6  395.60  12.43  22.9          0          NaN          NaN
7  396.90  19.15  27.1          0          NaN          NaN
8  386.63  29.93  16.5          0          NaN          NaN
9  386.71  17.10  18.9          0          NaN          NaN
```

[9]:
```python
df.drop(columns=['Unnamed: 15','Unnamed: 16'],inplace=True)
```

[10]:
```python
df.drop(columns=['CAT. MEDV'],inplace=True)
```

Checking for null values

[11]:
```
df.isnull().sum()
```

[11]:
```
CRIM       0
ZN         0
INDUS      0
CHAS       0
NOX        0
RM         0
AGE        0
DIS        0
RAD        0
TAX        0
PTRATIO    0
B          0
LSTAT      0
MEDV       0
dtype: int64
```

[12]:
```
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
```

[13]:
```
df.describe()
```

[13]:
```
             CRIM          ZN       INDUS        CHAS         NOX          RM  \
count  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000
mean     3.613524   11.363636   11.136779    0.069170    0.554695    6.284634
```

```
std       8.601545   23.322453    6.860353    0.253994    0.115878    0.702617
min       0.006320    0.000000    0.460000    0.000000    0.385000    3.561000
25%       0.082045    0.000000    5.190000    0.000000    0.449000    5.885500
50%       0.256510    0.000000    9.690000    0.000000    0.538000    6.208500
75%       3.677083   12.500000   18.100000    0.000000    0.624000    6.623500
max      88.976200  100.000000   27.740000    1.000000    0.871000    8.780000

              AGE         DIS         RAD         TAX     PTRATIO           B  \
count  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000
mean    68.574901    3.795043    9.549407  408.237154   18.455534  356.674032
std     28.148861    2.105710    8.707259  168.537116    2.164946   91.294864
min      2.900000    1.129600    1.000000  187.000000   12.600000    0.320000
25%     45.025000    2.100175    4.000000  279.000000   17.400000  375.377500
50%     77.500000    3.207450    5.000000  330.000000   19.050000  391.440000
75%     94.075000    5.188425   24.000000  666.000000   20.200000  396.225000
max    100.000000   12.126500   24.000000  711.000000   22.000000  396.900000

            LSTAT        MEDV
count  506.000000  506.000000
mean    12.653063   22.532806
std      7.141062    9.197104
min      1.730000    5.000000
25%      6.950000   17.025000
50%     11.360000   21.200000
75%     16.955000   25.000000
max     37.970000   50.000000
```

Checking correlation with target variable MEDV

[14]:
```python
df.corr()['MEDV'].sort_values()
```

[14]:
```
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
```

[15]:
```
X = df.loc[:,['LSTAT','PTRATIO','RM']]
Y = df.loc[:,"MEDV"]
X.shape,Y.shape
```

[15]:
```
((506, 3), (506,))
```

0.0.2 Preparing training and testing data set

[16]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(X,Y,test_size=0.
 ↪25,random_state=10)
```

0.0.3 Normalizing training and testing dataset

[17]:
```
from sklearn.preprocessing import StandardScaler
```

[18]:
```
scaler = StandardScaler()
```

[19]:
```
scaler.fit(x_train)
```

[19]:
```
StandardScaler()
```

[20]:
```
x_train = scaler.transform(x_train)
x_test = scaler.transform(x_test)
```

0.0.4 Preparing model

[21]:
```
from keras.models import Sequential
from keras.layers import Dense
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[21], line 1
----> 1 from keras.models import Sequential
      2 from keras.layers import Dense

ModuleNotFoundError: No module named 'keras'
```

[22]:
```
model = Sequential()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[22], line 1
----> 1 model = Sequential()
```

```
NameError: name 'Sequential' is not defined
```

[37]:
```
model.add(Dense(128,input_shape=(3,),activation='relu',name='input'))
model.add(Dense(64,activation='relu',name='layer_1'))
model.add(Dense(1,activation='linear',name='output'))
model.compile(optimizer='adam', loss='mse', metrics=['mae'])
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input (Dense)               (None, 128)               512

 layer_1 (Dense)             (None, 64)                8256

 output (Dense)              (None, 1)                 65

=================================================================
Total params: 8,833
Trainable params: 8,833
Non-trainable params: 0
_________________________________________________________________
```

[38]:
```
model.fit(x_train,y_train,epochs=100,validation_split=0.05)
```

```
/home/pratik/.local/lib/python3.8/site-
packages/keras/engine/data_adapter.py:1699: FutureWarning: The behavior of
`series[i:j]` with an integer-dtype index is deprecated. In a future version,
this will be treated as *label-based* indexing, consistent with e.g. `series[i]`
lookups. To retain the old behavior, use `series.iloc[i:j]`. To get the future
behavior, use `series.loc[i:j]`.
  return t[start:end]

Epoch 1/100
12/12 [==============================] - 1s 16ms/step - loss: 521.6520 - mae:
21.1973 - val_loss: 684.2971 - val_mae: 23.4446
Epoch 2/100
12/12 [==============================] - 0s 3ms/step - loss: 480.6119 - mae:
20.2731 - val_loss: 630.4888 - val_mae: 22.2464
Epoch 3/100
12/12 [==============================] - 0s 3ms/step - loss: 423.7050 - mae:
18.9328 - val_loss: 557.4312 - val_mae: 20.5203
Epoch 4/100
12/12 [==============================] - 0s 3ms/step - loss: 347.1570 - mae:
16.9609 - val_loss: 464.9811 - val_mae: 18.3199
Epoch 5/100
```

```
12/12 [==============================] - 0s 3ms/step - loss: 251.9777 - mae:
14.3374 - val_loss: 361.5852 - val_mae: 15.6702
Epoch 6/100
12/12 [==============================] - 0s 3ms/step - loss: 158.9359 - mae:
11.2933 - val_loss: 259.4221 - val_mae: 12.6659
Epoch 7/100
12/12 [==============================] - 0s 3ms/step - loss: 85.3764 - mae:
8.0840 - val_loss: 184.6476 - val_mae: 10.3077
Epoch 8/100
12/12 [==============================] - 0s 3ms/step - loss: 51.1009 - mae:
5.8868 - val_loss: 143.0784 - val_mae: 8.6210
Epoch 9/100
12/12 [==============================] - 0s 3ms/step - loss: 41.0831 - mae:
5.0207 - val_loss: 122.0487 - val_mae: 7.6780
Epoch 10/100
12/12 [==============================] - 0s 3ms/step - loss: 34.8150 - mae:
4.4855 - val_loss: 109.6800 - val_mae: 7.2366
Epoch 11/100
12/12 [==============================] - 0s 3ms/step - loss: 29.9512 - mae:
4.0640 - val_loss: 102.8176 - val_mae: 7.0559
Epoch 12/100
12/12 [==============================] - 0s 3ms/step - loss: 26.9342 - mae:
3.8241 - val_loss: 98.0677 - val_mae: 6.8996
Epoch 13/100
12/12 [==============================] - 0s 3ms/step - loss: 24.9706 - mae:
3.6743 - val_loss: 93.2265 - val_mae: 6.7188
Epoch 14/100
12/12 [==============================] - 0s 3ms/step - loss: 23.6977 - mae:
3.5606 - val_loss: 91.2786 - val_mae: 6.6632
Epoch 15/100
12/12 [==============================] - 0s 3ms/step - loss: 22.7392 - mae:
3.4870 - val_loss: 89.9420 - val_mae: 6.5950
Epoch 16/100
12/12 [==============================] - 0s 3ms/step - loss: 21.9554 - mae:
3.4222 - val_loss: 87.4618 - val_mae: 6.5106
Epoch 17/100
12/12 [==============================] - 0s 3ms/step - loss: 21.3945 - mae:
3.3762 - val_loss: 86.6438 - val_mae: 6.4625
Epoch 18/100
12/12 [==============================] - 0s 3ms/step - loss: 20.7621 - mae:
3.3364 - val_loss: 86.2997 - val_mae: 6.4570
Epoch 19/100
12/12 [==============================] - 0s 3ms/step - loss: 20.3429 - mae:
3.3059 - val_loss: 87.6115 - val_mae: 6.5001
Epoch 20/100
12/12 [==============================] - 0s 3ms/step - loss: 19.7910 - mae:
3.2573 - val_loss: 86.8414 - val_mae: 6.4395
Epoch 21/100
```

```
12/12 [==============================] - 0s 3ms/step - loss: 19.3040 - mae:
3.2172 - val_loss: 85.3897 - val_mae: 6.3410
Epoch 22/100
12/12 [==============================] - 0s 3ms/step - loss: 18.9326 - mae:
3.1857 - val_loss: 83.8950 - val_mae: 6.2748
Epoch 23/100
12/12 [==============================] - 0s 3ms/step - loss: 18.4641 - mae:
3.1427 - val_loss: 85.9416 - val_mae: 6.2838
Epoch 24/100
12/12 [==============================] - 0s 3ms/step - loss: 17.9731 - mae:
3.0871 - val_loss: 85.2962 - val_mae: 6.2192
Epoch 25/100
12/12 [==============================] - 0s 3ms/step - loss: 17.5960 - mae:
3.0524 - val_loss: 84.0756 - val_mae: 6.1301
Epoch 26/100
12/12 [==============================] - 0s 3ms/step - loss: 17.2496 - mae:
3.0313 - val_loss: 83.8474 - val_mae: 6.0809
Epoch 27/100
12/12 [==============================] - 0s 3ms/step - loss: 16.8987 - mae:
3.0019 - val_loss: 82.9085 - val_mae: 6.0096
Epoch 28/100
12/12 [==============================] - 0s 3ms/step - loss: 16.6120 - mae:
2.9855 - val_loss: 82.4742 - val_mae: 5.9599
Epoch 29/100
12/12 [==============================] - 0s 3ms/step - loss: 16.3804 - mae:
2.9552 - val_loss: 84.0461 - val_mae: 5.9848
Epoch 30/100
12/12 [==============================] - 0s 4ms/step - loss: 16.0876 - mae:
2.9228 - val_loss: 82.8573 - val_mae: 5.8955
Epoch 31/100
12/12 [==============================] - 0s 3ms/step - loss: 15.8775 - mae:
2.9189 - val_loss: 82.3173 - val_mae: 5.8456
Epoch 32/100
12/12 [==============================] - 0s 3ms/step - loss: 15.7003 - mae:
2.9055 - val_loss: 82.2009 - val_mae: 5.8318
Epoch 33/100
12/12 [==============================] - 0s 3ms/step - loss: 15.5040 - mae:
2.8817 - val_loss: 81.1925 - val_mae: 5.7864
Epoch 34/100
12/12 [==============================] - 0s 3ms/step - loss: 15.2819 - mae:
2.8577 - val_loss: 82.7803 - val_mae: 5.8049
Epoch 35/100
12/12 [==============================] - 0s 3ms/step - loss: 15.1625 - mae:
2.8552 - val_loss: 82.5307 - val_mae: 5.7775
Epoch 36/100
12/12 [==============================] - 0s 3ms/step - loss: 14.9914 - mae:
2.8294 - val_loss: 82.5536 - val_mae: 5.7522
Epoch 37/100
```

```
12/12 [==============================] - 0s 3ms/step - loss: 14.7915 - mae:
2.8238 - val_loss: 81.7447 - val_mae: 5.7054
Epoch 38/100
12/12 [==============================] - 0s 3ms/step - loss: 14.6339 - mae:
2.8152 - val_loss: 80.9904 - val_mae: 5.6606
Epoch 39/100
12/12 [==============================] - 0s 3ms/step - loss: 14.5984 - mae:
2.7850 - val_loss: 83.0082 - val_mae: 5.7130
Epoch 40/100
12/12 [==============================] - 0s 3ms/step - loss: 14.4250 - mae:
2.7908 - val_loss: 79.6669 - val_mae: 5.6064
Epoch 41/100
12/12 [==============================] - 0s 3ms/step - loss: 14.5862 - mae:
2.8150 - val_loss: 84.1153 - val_mae: 5.7513
Epoch 42/100
12/12 [==============================] - 0s 3ms/step - loss: 14.1320 - mae:
2.7555 - val_loss: 81.3843 - val_mae: 5.6236
Epoch 43/100
12/12 [==============================] - 0s 3ms/step - loss: 14.0879 - mae:
2.7372 - val_loss: 79.7218 - val_mae: 5.5161
Epoch 44/100
12/12 [==============================] - 0s 3ms/step - loss: 13.9245 - mae:
2.7283 - val_loss: 82.5691 - val_mae: 5.5710
Epoch 45/100
12/12 [==============================] - 0s 3ms/step - loss: 13.8294 - mae:
2.7071 - val_loss: 83.5797 - val_mae: 5.5885
Epoch 46/100
12/12 [==============================] - 0s 3ms/step - loss: 13.7279 - mae:
2.6897 - val_loss: 81.4584 - val_mae: 5.5139
Epoch 47/100
12/12 [==============================] - 0s 3ms/step - loss: 13.6327 - mae:
2.6876 - val_loss: 81.5313 - val_mae: 5.5337
Epoch 48/100
12/12 [==============================] - 0s 3ms/step - loss: 13.4871 - mae:
2.6831 - val_loss: 81.8829 - val_mae: 5.5197
Epoch 49/100
12/12 [==============================] - 0s 3ms/step - loss: 13.3905 - mae:
2.6598 - val_loss: 81.6146 - val_mae: 5.5109
Epoch 50/100
12/12 [==============================] - 0s 3ms/step - loss: 13.3846 - mae:
2.6507 - val_loss: 82.3006 - val_mae: 5.5114
Epoch 51/100
12/12 [==============================] - 0s 3ms/step - loss: 13.3731 - mae:
2.6582 - val_loss: 79.2186 - val_mae: 5.4195
Epoch 52/100
12/12 [==============================] - 0s 3ms/step - loss: 13.0755 - mae:
2.6232 - val_loss: 82.1192 - val_mae: 5.4674
Epoch 53/100
```

```
12/12 [==============================] - 0s 3ms/step - loss: 13.1333 - mae:
2.6301 - val_loss: 82.2511 - val_mae: 5.4621
Epoch 54/100
12/12 [==============================] - 0s 3ms/step - loss: 12.8883 - mae:
2.6066 - val_loss: 80.3709 - val_mae: 5.3996
Epoch 55/100
12/12 [==============================] - 0s 3ms/step - loss: 12.8645 - mae:
2.5896 - val_loss: 80.9426 - val_mae: 5.3828
Epoch 56/100
12/12 [==============================] - 0s 3ms/step - loss: 12.9274 - mae:
2.5858 - val_loss: 80.2519 - val_mae: 5.3328
Epoch 57/100
12/12 [==============================] - 0s 3ms/step - loss: 12.7558 - mae:
2.5847 - val_loss: 82.8984 - val_mae: 5.3706
Epoch 58/100
12/12 [==============================] - 0s 3ms/step - loss: 12.7283 - mae:
2.5830 - val_loss: 80.4384 - val_mae: 5.3213
Epoch 59/100
12/12 [==============================] - 0s 3ms/step - loss: 12.5721 - mae:
2.5648 - val_loss: 81.8696 - val_mae: 5.3709
Epoch 60/100
12/12 [==============================] - 0s 3ms/step - loss: 12.4515 - mae:
2.5413 - val_loss: 80.1730 - val_mae: 5.3064
Epoch 61/100
12/12 [==============================] - 0s 3ms/step - loss: 12.3504 - mae:
2.5372 - val_loss: 82.3537 - val_mae: 5.3312
Epoch 62/100
12/12 [==============================] - 0s 3ms/step - loss: 12.2713 - mae:
2.5263 - val_loss: 81.8209 - val_mae: 5.3155
Epoch 63/100
12/12 [==============================] - 0s 3ms/step - loss: 12.1914 - mae:
2.5186 - val_loss: 81.0043 - val_mae: 5.2819
Epoch 64/100
12/12 [==============================] - 0s 3ms/step - loss: 12.0838 - mae:
2.5141 - val_loss: 82.2268 - val_mae: 5.2991
Epoch 65/100
12/12 [==============================] - 0s 3ms/step - loss: 12.0705 - mae:
2.5192 - val_loss: 80.9062 - val_mae: 5.2588
Epoch 66/100
12/12 [==============================] - 0s 3ms/step - loss: 12.1463 - mae:
2.4995 - val_loss: 81.0292 - val_mae: 5.2457
Epoch 67/100
12/12 [==============================] - 0s 3ms/step - loss: 11.9510 - mae:
2.4905 - val_loss: 81.9456 - val_mae: 5.2709
Epoch 68/100
12/12 [==============================] - 0s 3ms/step - loss: 11.9244 - mae:
2.5124 - val_loss: 80.4283 - val_mae: 5.2259
Epoch 69/100
```

```
12/12 [==============================] - 0s 3ms/step - loss: 11.8111 - mae:
2.4702 - val_loss: 80.2744 - val_mae: 5.2222
Epoch 70/100
12/12 [==============================] - 0s 3ms/step - loss: 11.8484 - mae:
2.4784 - val_loss: 82.1776 - val_mae: 5.2416
Epoch 71/100
12/12 [==============================] - 0s 3ms/step - loss: 11.6163 - mae:
2.4718 - val_loss: 79.9181 - val_mae: 5.1496
Epoch 72/100
12/12 [==============================] - 0s 3ms/step - loss: 11.6516 - mae:
2.4528 - val_loss: 81.3688 - val_mae: 5.1745
Epoch 73/100
12/12 [==============================] - 0s 3ms/step - loss: 11.4880 - mae:
2.4364 - val_loss: 81.5457 - val_mae: 5.1724
Epoch 74/100
12/12 [==============================] - 0s 3ms/step - loss: 11.6022 - mae:
2.4720 - val_loss: 82.5531 - val_mae: 5.1809
Epoch 75/100
12/12 [==============================] - 0s 3ms/step - loss: 11.3654 - mae:
2.4368 - val_loss: 81.2617 - val_mae: 5.1176
Epoch 76/100
12/12 [==============================] - 0s 3ms/step - loss: 11.3965 - mae:
2.4344 - val_loss: 82.1688 - val_mae: 5.1470
Epoch 77/100
12/12 [==============================] - 0s 3ms/step - loss: 11.4002 - mae:
2.4252 - val_loss: 81.4868 - val_mae: 5.1287
Epoch 78/100
12/12 [==============================] - 0s 3ms/step - loss: 11.2641 - mae:
2.4045 - val_loss: 80.9666 - val_mae: 5.1395
Epoch 79/100
12/12 [==============================] - 0s 3ms/step - loss: 11.1954 - mae:
2.3919 - val_loss: 81.4386 - val_mae: 5.0960
Epoch 80/100
12/12 [==============================] - 0s 3ms/step - loss: 11.2026 - mae:
2.4224 - val_loss: 80.0435 - val_mae: 5.1421
Epoch 81/100
12/12 [==============================] - 0s 3ms/step - loss: 11.2060 - mae:
2.4191 - val_loss: 81.8591 - val_mae: 5.1589
Epoch 82/100
12/12 [==============================] - 0s 3ms/step - loss: 11.0078 - mae:
2.3819 - val_loss: 81.2953 - val_mae: 5.0771
Epoch 83/100
12/12 [==============================] - 0s 3ms/step - loss: 11.0476 - mae:
2.3804 - val_loss: 81.5968 - val_mae: 5.0807
Epoch 84/100
12/12 [==============================] - 0s 3ms/step - loss: 11.0166 - mae:
2.3936 - val_loss: 83.5097 - val_mae: 5.1785
Epoch 85/100
```

```
12/12 [==============================] - 0s 3ms/step - loss: 10.9730 - mae:
2.3777 - val_loss: 81.7998 - val_mae: 5.1038
Epoch 86/100
12/12 [==============================] - 0s 3ms/step - loss: 10.8896 - mae:
2.3799 - val_loss: 82.7827 - val_mae: 5.1333
Epoch 87/100
12/12 [==============================] - 0s 3ms/step - loss: 10.8807 - mae:
2.3697 - val_loss: 83.3495 - val_mae: 5.1081
Epoch 88/100
12/12 [==============================] - 0s 3ms/step - loss: 10.8414 - mae:
2.3782 - val_loss: 84.2799 - val_mae: 5.2340
Epoch 89/100
12/12 [==============================] - 0s 3ms/step - loss: 10.8551 - mae:
2.3657 - val_loss: 82.8966 - val_mae: 5.1426
Epoch 90/100
12/12 [==============================] - 0s 3ms/step - loss: 10.7487 - mae:
2.3737 - val_loss: 82.9162 - val_mae: 5.1396
Epoch 91/100
12/12 [==============================] - 0s 3ms/step - loss: 10.7692 - mae:
2.3567 - val_loss: 81.7738 - val_mae: 5.0534
Epoch 92/100
12/12 [==============================] - 0s 3ms/step - loss: 10.8826 - mae:
2.3858 - val_loss: 83.4745 - val_mae: 5.1617
Epoch 93/100
12/12 [==============================] - 0s 3ms/step - loss: 10.7417 - mae:
2.3593 - val_loss: 80.3109 - val_mae: 5.0397
Epoch 94/100
12/12 [==============================] - 0s 3ms/step - loss: 10.6165 - mae:
2.3462 - val_loss: 83.9402 - val_mae: 5.1520
Epoch 95/100
12/12 [==============================] - 0s 3ms/step - loss: 10.5802 - mae:
2.3409 - val_loss: 83.6694 - val_mae: 5.1339
Epoch 96/100
12/12 [==============================] - 0s 3ms/step - loss: 10.6581 - mae:
2.3396 - val_loss: 82.1335 - val_mae: 5.0487
Epoch 97/100
12/12 [==============================] - 0s 3ms/step - loss: 10.6420 - mae:
2.3562 - val_loss: 83.7102 - val_mae: 5.1258
Epoch 98/100
12/12 [==============================] - 0s 3ms/step - loss: 10.6508 - mae:
2.3370 - val_loss: 81.2809 - val_mae: 5.0286
Epoch 99/100
12/12 [==============================] - 0s 3ms/step - loss: 10.5333 - mae:
2.3380 - val_loss: 83.6725 - val_mae: 5.1323
Epoch 100/100
12/12 [==============================] - 0s 3ms/step - loss: 10.3605 - mae:
2.3148 - val_loss: 83.1757 - val_mae: 5.0693
```

[38]: <keras.callbacks.History at 0x7fbddc67a490>

[39]:
```
output = model.evaluate(x_test,y_test)
```

```
4/4 [==============================] - 0s 4ms/step - loss: 22.2640 - mae: 3.1030
```

[44]:
```
print(f"Mean Squared Error: {output[0]}"
      ,f"Mean Absolute Error: {output[1]}",sep="\n")
```

```
Mean Squared Error: 22.26400375366211
Mean Absolute Error: 3.1030352115631104
```

[45]:
```
y_pred = model.predict(x=x_test)
```

```
4/4 [==============================] - 0s 4ms/step
```

[46]:
```
print(*zip(y_pred,y_test))
```

```
(array([24.506329], dtype=float32), 28.4) (array([30.56254], dtype=float32),
31.1) (array([25.646534], dtype=float32), 23.5) (array([27.445961],
dtype=float32), 26.6) (array([19.707462], dtype=float32), 19.6)
(array([16.464933], dtype=float32), 14.3) (array([42.08562], dtype=float32),
50.0) (array([14.898803], dtype=float32), 14.3) (array([20.1403],
dtype=float32), 20.7) (array([43.237473], dtype=float32), 37.6)
(array([17.841496], dtype=float32), 20.4) (array([26.564915], dtype=float32),
27.5) (array([22.473684], dtype=float32), 36.2) (array([32.409435],
dtype=float32), 32.0) (array([31.079502], dtype=float32), 33.1)
(array([51.951847], dtype=float32), 48.8) (array([25.474497], dtype=float32),
24.6) (array([19.781612], dtype=float32), 26.4) (array([21.237524],
dtype=float32), 23.2) (array([19.808071], dtype=float32), 17.0)
(array([33.196445], dtype=float32), 41.3) (array([15.7997875], dtype=float32),
14.9) (array([22.308857], dtype=float32), 18.5) (array([24.506542],
dtype=float32), 25.0) (array([36.95174], dtype=float32), 36.4)
(array([21.02158], dtype=float32), 19.5) (array([19.072449], dtype=float32),
27.1) (array([16.65482], dtype=float32), 14.9) (array([42.864365],
dtype=float32), 46.0) (array([11.255093], dtype=float32), 17.9)
(array([34.67793], dtype=float32), 30.3) (array([32.396557], dtype=float32),
31.6) (array([26.159986], dtype=float32), 23.1) (array([23.886639],
dtype=float32), 24.7) (array([15.139406], dtype=float32), 16.7)
(array([19.983408], dtype=float32), 18.3) (array([8.552556], dtype=float32),
8.4) (array([32.28726], dtype=float32), 37.3) (array([24.598032],
dtype=float32), 22.1) (array([24.136978], dtype=float32), 22.0)
(array([39.275646], dtype=float32), 46.7) (array([25.97865], dtype=float32),
30.1) (array([13.960388], dtype=float32), 12.1) (array([29.030474],
dtype=float32), 29.1) (array([17.66833], dtype=float32), 16.6)
(array([27.10108], dtype=float32), 23.9) (array([17.861822], dtype=float32),
19.9) (array([18.575834], dtype=float32), 21.4) (array([44.4922],
dtype=float32), 45.4) (array([16.563892], dtype=float32), 15.6)
```

(array([20.46928], dtype=float32), 22.7) (array([14.620689], dtype=float32), 12.5) (array([20.612724], dtype=float32), 24.3) (array([39.50388], dtype=float32), 43.8) (array([24.186283], dtype=float32), 22.0) (array([34.64388], dtype=float32), 33.8) (array([19.930775], dtype=float32), 19.3) (array([18.89568], dtype=float32), 22.6) (array([21.176815], dtype=float32), 16.1) (array([21.034569], dtype=float32), 15.0) (array([18.339993], dtype=float32), 19.6) (array([21.137554], dtype=float32), 21.2) (array([51.624172], dtype=float32), 50.0) (array([56.14203], dtype=float32), 50.0) (array([27.36555], dtype=float32), 29.4) (array([15.268709], dtype=float32), 17.8) (array([24.735828], dtype=float32), 22.8) (array([12.591748], dtype=float32), 8.8) (array([27.126993], dtype=float32), 32.5) (array([40.339428], dtype=float32), 42.8) (array([16.689852], dtype=float32), 12.6) (array([27.98465], dtype=float32), 28.6) (array([17.789898], dtype=float32), 19.1) (array([21.564999], dtype=float32), 50.0) (array([21.122467], dtype=float32), 27.5) (array([12.93322], dtype=float32), 23.7) (array([48.340298], dtype=float32), 50.0) (array([10.2400875], dtype=float32), 7.2) (array([20.198032], dtype=float32), 18.7) (array([32.495087], dtype=float32), 37.0) (array([20.165474], dtype=float32), 22.9) (array([24.815548], dtype=float32), 22.9) (array([20.013338], dtype=float32), 17.1) (array([24.288055], dtype=float32), 22.0) (array([31.427582], dtype=float32), 23.6) (array([25.81147], dtype=float32), 23.9) (array([25.682194], dtype=float32), 27.1) (array([34.150513], dtype=float32), 29.0) (array([23.863184], dtype=float32), 22.2) (array([11.1325245], dtype=float32), 7.0) (array([23.009584], dtype=float32), 20.7) (array([21.158339], dtype=float32), 18.5) (array([23.660124], dtype=float32), 21.6) (array([24.155087], dtype=float32), 23.0) (array([18.824171], dtype=float32), 16.0) (array([19.678234], dtype=float32), 15.0) (array([25.252113], dtype=float32), 23.9) (array([19.542976], dtype=float32), 24.4) (array([21.263361], dtype=float32), 22.6) (array([18.398333], dtype=float32), 19.8) (array([21.704134], dtype=float32), 22.2) (array([19.568924], dtype=float32), 18.6) (array([19.732323], dtype=float32), 19.7) (array([23.8418], dtype=float32), 23.1) (array([13.837982], dtype=float32), 13.5) (array([21.036032], dtype=float32), 21.2) (array([19.110723], dtype=float32), 23.1) (array([15.809771], dtype=float32), 13.6) (array([28.556469], dtype=float32), 22.8) (array([23.521175], dtype=float32), 18.2) (array([11.745223], dtype=float32), 13.1) (array([17.627365], dtype=float32), 23.2) (array([24.611477], dtype=float32), 22.8) (array([24.487736], dtype=float32), 25.1) (array([21.870634], dtype=float32), 18.9) (array([13.559553], dtype=float32), 10.9) (array([14.871121], dtype=float32), 19.3) (array([20.880054], dtype=float32), 17.4) (array([18.778835], dtype=float32), 15.6) (array([20.01767], dtype=float32), 20.6) (array([30.024704], dtype=float32), 50.0) (array([35.32085], dtype=float32), 32.7) (array([21.411596], dtype=float32), 21.8) (array([16.723742], dtype=float32), 13.4) (array([17.538132], dtype=float32), 16.6) (array([23.341366], dtype=float32), 23.6) (array([13.014972], dtype=float32), 11.0)

[ ]:

```

```