Name :- Omkar Balwade

#Roll n.o 4107

#Div A

#Practical 1:- Predicting Boston House Prices: A Comparative Analysis of Linear Regression and Neural Networks

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_error
# Importing Keras for neural networks
import keras
from keras.models import Sequential
from keras.layers import Dense
```

```
WARNING:tensorflow:From C:\Users\Omkar\AppData\Local\anaconda3\Lib\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_e
```

In [2]:

```
# Load Boston dataset
data = pd.read_csv("boston.csv")
```

In [3]:

```
data
```

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	296.0	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	242.0	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	242.0	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	222.0	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	222.0	18.7	396.90	5.33	36.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273.0	21.0	391.99	9.67	22.4
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273.0	21.0	396.90	9.08	20.6
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273.0	21.0	396.90	5.64	23.9
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273.0	21.0	393.45	6.48	22.0
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273.0	21.0	396.90	7.88	11.9

506 rows × 14 columns

In [4]:

```
# Checking the null values in the dataset
data.isnull().sum()
```

Out [4]:

```
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
```

In [5]:

```
# Checking the statistics of the data
data.describe()
```

Out [5]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATI
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.00000
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68.574901	3.795043	9.549407	408.237154	18.455534
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148861	2.105710	8.707259	168.537116	2.164946
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.129600	1.000000	187.000000	12.600000
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025000	2.100175	4.000000	279.000000	17.400000
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500000	3.207450	5.000000	330.000000	19.050000
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	94.075000	5.188425	24.000000	666.000000	20.200000

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATI
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.126500	24.000000	711.000000	22.000000

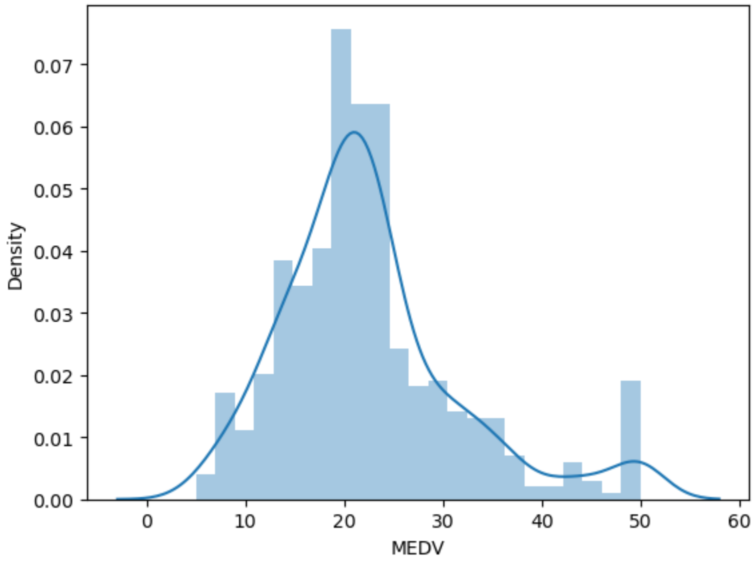
In [6]:

```
# Checking the distribution of the target variable
sns.distplot(data.MEDV)
```

```
C:\Users\Omkar\AppData\Local\Temp\ipykernel_13348\2421881879.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data.MEDV)
```

Out [6]:

```
<Axes: xlabel='MEDV', ylabel='Density'>
```

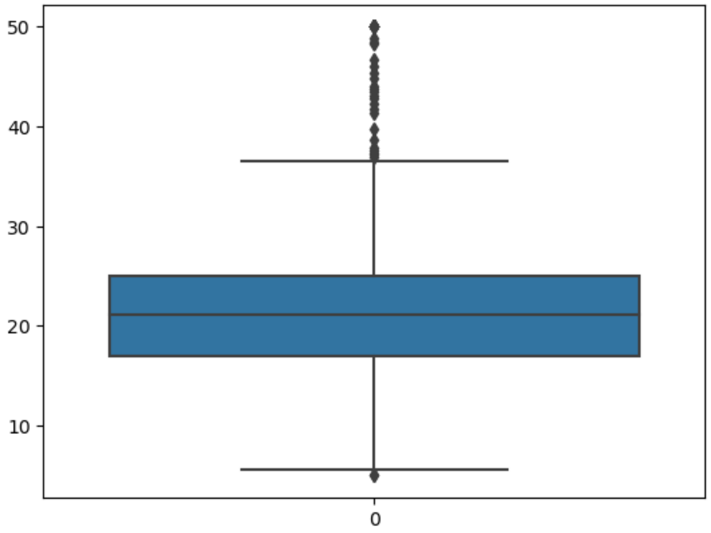


In [7]:

```
# Distribution using box plot
sns.boxplot(data.MEDV)
```

Out [7]:

```
<Axes: >
```



In [8]:

```
# Checking the correlation of the independent feature with the dependent feature
correlation = data.corr()
correlation.loc['MEDV']
```

Out [8]:

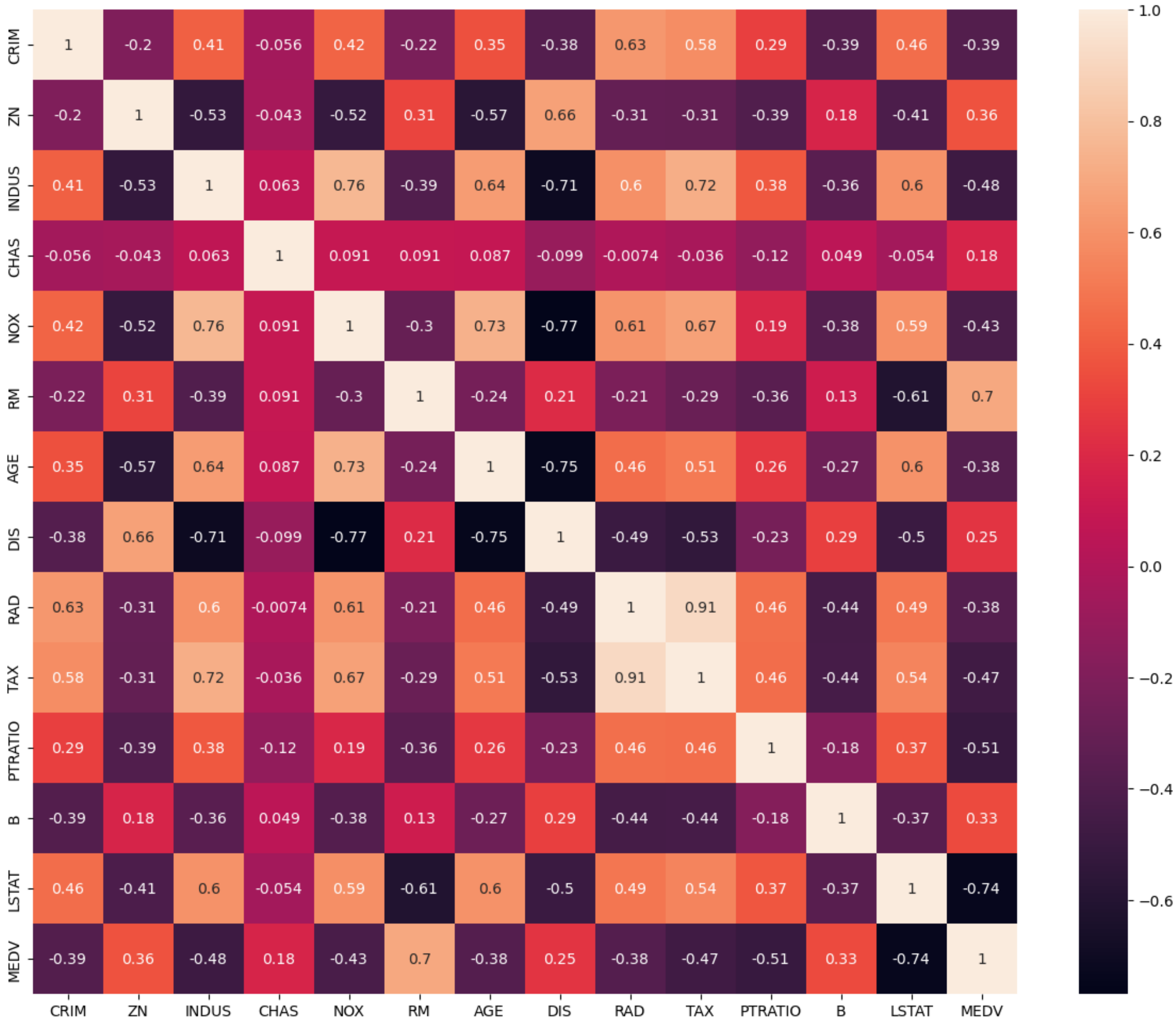
```
CRIM      -0.388305
ZN         0.360445
INDUS     -0.483725
CHAS       0.175260
NOX       -0.427321
RM         0.695360
AGE       -0.376955
DIS        0.249929
RAD       -0.381626
TAX       -0.468536
PTRATIO   -0.507787
B          0.333461
LSTAT     -0.737663
MEDV       1.000000
Name: MEDV, dtype: float64
```

In [9]:

```
# Plotting the heatmap
fig, axes = plt.subplots(figsize=(15, 12))
```

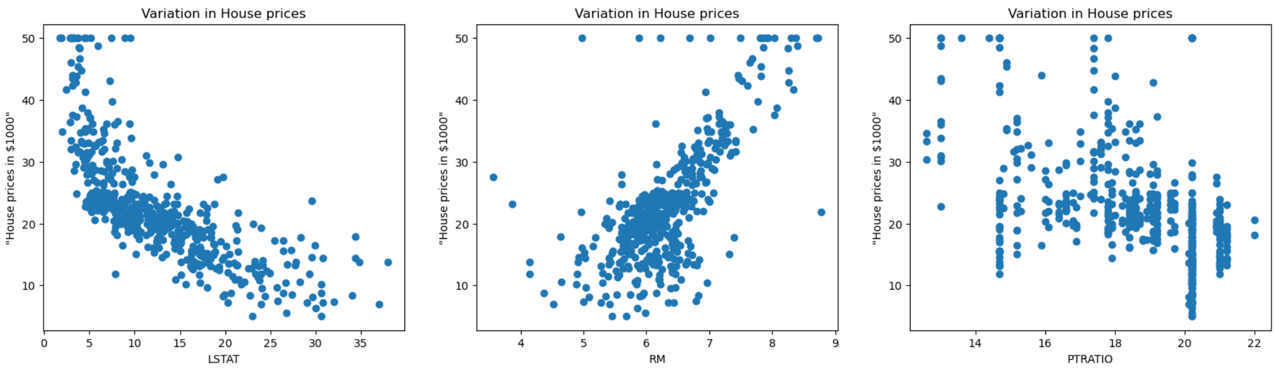
```
sns.heatmap(correlation, square=True, annot=True)
```

Out [9]: <Axes: >



In [10]:

```
# Checking the scatter plot with the most correlated features
plt.figure(figsize=(20, 5))
features = ['LSTAT', 'RM', 'PTRATIO']
for i, col in enumerate(features):
    plt.subplot(1, len(features), i+1)
    x = data[col]
    y = data.MEDV
    plt.scatter(x, y, marker='o')
    plt.title("Variation in House prices")
    plt.xlabel(col)
    plt.ylabel('"House prices in $1000"')
```



In [11]:

```
# Splitting the dependent feature and independent feature
X = data.iloc[:, :-1]
y = data.MEDV
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [12]:

```
# Standardizing the dataset
scaler = StandardScaler()
```

```
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

In [13]:
```
# Linear Regression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
y_pred_lr = regressor.predict(X_test)
```

In [14]:
```
# Model Evaluation
mse_lr = mean_squared_error(y_test, y_pred_lr)
mae_lr = mean_absolute_error(y_test, y_pred_lr)
r2_lr = r2_score(y_test, y_pred_lr)
```

In [15]:
```
# Neural Networks
model = Sequential()
model.add(Dense(128, activation='relu', input_dim=13))
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(1))
```

```
WARNING:tensorflow:From C:\Users\Omkar\AppData\Local\anaconda3\Lib\site-packages\keras\src\backend.py:873: The name tf.get_default_graph is deprecat
```

In [16]:
```
model.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae'])
```

```
WARNING:tensorflow:From C:\Users\Omkar\AppData\Local\anaconda3\Lib\site-packages\keras\src\optimizers\__init__.py:309: The name tf.train.Optimizer i
```

In [17]:
```
history = model.fit(X_train, y_train, epochs=100, validation_split=0.05)
```

```
Epoch 1/100
WARNING:tensorflow:From C:\Users\Omkar\AppData\Local\anaconda3\Lib\site-packages\keras\src\utils\tf_utils.py:492: The name tf.ragged.RaggedTensorVal

WARNING:tensorflow:From C:\Users\Omkar\AppData\Local\anaconda3\Lib\site-packages\keras\src\engine\base_layer_utils.py:384: The name tf.executing_eag

12/12 [==============================] - 3s 44ms/step - loss: 600.1306 - mae: 22.6068 - val_loss: 502.6846 - val_mae: 20.9355
Epoch 2/100
12/12 [==============================] - 0s 11ms/step - loss: 549.3881 - mae: 21.4522 - val_loss: 432.5534 - val_mae: 19.2446
Epoch 3/100
12/12 [==============================] - 0s 9ms/step - loss: 430.7413 - mae: 18.5542 - val_loss: 275.8118 - val_mae: 14.8036
Epoch 4/100
12/12 [==============================] - 0s 13ms/step - loss: 207.1008 - mae: 11.9711 - val_loss: 76.5751 - val_mae: 6.5055
Epoch 5/100
12/12 [==============================] - 0s 11ms/step - loss: 94.1821 - mae: 7.7540 - val_loss: 65.5131 - val_mae: 5.3271
Epoch 6/100
12/12 [==============================] - 0s 11ms/step - loss: 56.1305 - mae: 5.7674 - val_loss: 67.9639 - val_mae: 5.4303
Epoch 7/100
12/12 [==============================] - 0s 14ms/step - loss: 34.9353 - mae: 4.4432 - val_loss: 57.5597 - val_mae: 4.8712
Epoch 8/100
12/12 [==============================] - 0s 12ms/step - loss: 26.5227 - mae: 3.8596 - val_loss: 57.8510 - val_mae: 4.8976
Epoch 9/100
12/12 [==============================] - 0s 11ms/step - loss: 22.5743 - mae: 3.4850 - val_loss: 56.4505 - val_mae: 4.6971
Epoch 10/100
12/12 [==============================] - 0s 9ms/step - loss: 20.3374 - mae: 3.3084 - val_loss: 53.5390 - val_mae: 4.5857
Epoch 11/100
12/12 [==============================] - 0s 14ms/step - loss: 18.6829 - mae: 3.1660 - val_loss: 49.5476 - val_mae: 4.3183
Epoch 12/100
12/12 [==============================] - 0s 11ms/step - loss: 17.2778 - mae: 3.0777 - val_loss: 46.5235 - val_mae: 4.1202
Epoch 13/100
12/12 [==============================] - 0s 9ms/step - loss: 15.8904 - mae: 2.9149 - val_loss: 45.4987 - val_mae: 4.0214
Epoch 14/100
12/12 [==============================] - 0s 14ms/step - loss: 15.0165 - mae: 2.8484 - val_loss: 44.4792 - val_mae: 3.9450
Epoch 15/100
12/12 [==============================] - 0s 11ms/step - loss: 14.1592 - mae: 2.7534 - val_loss: 42.4556 - val_mae: 3.8419
Epoch 16/100
12/12 [==============================] - 0s 9ms/step - loss: 13.5510 - mae: 2.7063 - val_loss: 40.8982 - val_mae: 3.7539
Epoch 17/100
12/12 [==============================] - 0s 9ms/step - loss: 12.8428 - mae: 2.6160 - val_loss: 40.1941 - val_mae: 3.7384
Epoch 18/100
12/12 [==============================] - 0s 11ms/step - loss: 12.4407 - mae: 2.5760 - val_loss: 40.2403 - val_mae: 3.7248
Epoch 19/100
12/12 [==============================] - 0s 12ms/step - loss: 12.0987 - mae: 2.5344 - val_loss: 37.1383 - val_mae: 3.5365
Epoch 20/100
12/12 [==============================] - 0s 12ms/step - loss: 11.5824 - mae: 2.4594 - val_loss: 38.5864 - val_mae: 3.6473
Epoch 21/100
12/12 [==============================] - 0s 9ms/step - loss: 11.5743 - mae: 2.4970 - val_loss: 33.6557 - val_mae: 3.3846
Epoch 22/100
12/12 [==============================] - 0s 12ms/step - loss: 11.0689 - mae: 2.4057 - val_loss: 36.9645 - val_mae: 3.5664
Epoch 23/100
12/12 [==============================] - 0s 11ms/step - loss: 10.6962 - mae: 2.3870 - val_loss: 33.9775 - val_mae: 3.3919
Epoch 24/100
12/12 [==============================] - 0s 9ms/step - loss: 10.5388 - mae: 2.3726 - val_loss: 33.8313 - val_mae: 3.3700
Epoch 25/100
12/12 [==============================] - 0s 10ms/step - loss: 10.2317 - mae: 2.3245 - val_loss: 34.1691 - val_mae: 3.4135
Epoch 26/100
12/12 [==============================] - 0s 9ms/step - loss: 10.0453 - mae: 2.3219 - val_loss: 34.3386 - val_mae: 3.3816
Epoch 27/100
12/12 [==============================] - 0s 14ms/step - loss: 9.7600 - mae: 2.2751 - val_loss: 32.3325 - val_mae: 3.2715
Epoch 28/100
12/12 [==============================] - 0s 12ms/step - loss: 9.6396 - mae: 2.2606 - val_loss: 32.0110 - val_mae: 3.2989
Epoch 29/100
12/12 [==============================] - 0s 11ms/step - loss: 9.4730 - mae: 2.2324 - val_loss: 32.4374 - val_mae: 3.2716
Epoch 30/100
12/12 [==============================] - 0s 10ms/step - loss: 9.2617 - mae: 2.2213 - val_loss: 31.5458 - val_mae: 3.2369
Epoch 31/100
12/12 [==============================] - 0s 15ms/step - loss: 9.1483 - mae: 2.1953 - val_loss: 30.6773 - val_mae: 3.1786
Epoch 32/100
12/12 [==============================] - 0s 14ms/step - loss: 9.0073 - mae: 2.1795 - val_loss: 30.8302 - val_mae: 3.2135
Epoch 33/100
12/12 [==============================] - 0s 12ms/step - loss: 8.8303 - mae: 2.1775 - val_loss: 31.1382 - val_mae: 3.1905
Epoch 34/100
12/12 [==============================] - 0s 11ms/step - loss: 8.5699 - mae: 2.1300 - val_loss: 30.1554 - val_mae: 3.1505
```

```
Epoch 35/100
12/12 [==============================] - 0s 10ms/step - loss: 8.4070 - mae: 2.1203 - val_loss: 28.7764 - val_mae: 3.0854
Epoch 36/100
12/12 [==============================] - 0s 10ms/step - loss: 8.3610 - mae: 2.1348 - val_loss: 29.5295 - val_mae: 3.1783
Epoch 37/100
12/12 [==============================] - 0s 12ms/step - loss: 8.6397 - mae: 2.1339 - val_loss: 29.9310 - val_mae: 3.1288
Epoch 38/100
12/12 [==============================] - 0s 11ms/step - loss: 8.1604 - mae: 2.1164 - val_loss: 28.5666 - val_mae: 3.0967
Epoch 39/100
12/12 [==============================] - 0s 11ms/step - loss: 7.9047 - mae: 2.0618 - val_loss: 28.9429 - val_mae: 3.1288
Epoch 40/100
12/12 [==============================] - 0s 10ms/step - loss: 7.7069 - mae: 2.0415 - val_loss: 28.0041 - val_mae: 3.1405
Epoch 41/100
12/12 [==============================] - 0s 9ms/step - loss: 7.5815 - mae: 2.0268 - val_loss: 28.0550 - val_mae: 3.1049
Epoch 42/100
12/12 [==============================] - 0s 11ms/step - loss: 7.4925 - mae: 2.0051 - val_loss: 26.5498 - val_mae: 3.0387
Epoch 43/100
12/12 [==============================] - 0s 9ms/step - loss: 7.4094 - mae: 1.9922 - val_loss: 26.7096 - val_mae: 3.0056
Epoch 44/100
12/12 [==============================] - 0s 10ms/step - loss: 7.2063 - mae: 1.9897 - val_loss: 26.6955 - val_mae: 3.0510
Epoch 45/100
12/12 [==============================] - 0s 10ms/step - loss: 7.0729 - mae: 1.9634 - val_loss: 27.3698 - val_mae: 3.0516
Epoch 46/100
12/12 [==============================] - 0s 15ms/step - loss: 6.9537 - mae: 1.9530 - val_loss: 26.0075 - val_mae: 3.0069
Epoch 47/100
12/12 [==============================] - 0s 11ms/step - loss: 6.7810 - mae: 1.9386 - val_loss: 26.0204 - val_mae: 3.0453
Epoch 48/100
12/12 [==============================] - 0s 11ms/step - loss: 7.0939 - mae: 1.9771 - val_loss: 26.3599 - val_mae: 3.0412
Epoch 49/100
12/12 [==============================] - 0s 11ms/step - loss: 6.7323 - mae: 1.9315 - val_loss: 24.3894 - val_mae: 2.9854
Epoch 50/100
12/12 [==============================] - 0s 12ms/step - loss: 6.6278 - mae: 1.9304 - val_loss: 25.1735 - val_mae: 2.9238
Epoch 51/100
12/12 [==============================] - 0s 12ms/step - loss: 6.6365 - mae: 1.8959 - val_loss: 24.1109 - val_mae: 2.8762
Epoch 52/100
12/12 [==============================] - 0s 9ms/step - loss: 6.3529 - mae: 1.8924 - val_loss: 24.1354 - val_mae: 2.9699
Epoch 53/100
12/12 [==============================] - 0s 13ms/step - loss: 6.6247 - mae: 1.9232 - val_loss: 23.8500 - val_mae: 2.8957
Epoch 54/100
12/12 [==============================] - 0s 10ms/step - loss: 6.1001 - mae: 1.8347 - val_loss: 22.5117 - val_mae: 2.8657
Epoch 55/100
12/12 [==============================] - 0s 9ms/step - loss: 5.9832 - mae: 1.8205 - val_loss: 23.1864 - val_mae: 2.9161
Epoch 56/100
12/12 [==============================] - 0s 15ms/step - loss: 5.8629 - mae: 1.7946 - val_loss: 23.3148 - val_mae: 2.8649
Epoch 57/100
12/12 [==============================] - 0s 11ms/step - loss: 5.7530 - mae: 1.7965 - val_loss: 22.0594 - val_mae: 2.8308
Epoch 58/100

12/12 [==============================] - 0s 11ms/step - loss: 5.6349 - mae: 1.7581 - val_loss: 22.7982 - val_mae: 2.8742
Epoch 59/100
12/12 [==============================] - 0s 12ms/step - loss: 5.5776 - mae: 1.7764 - val_loss: 21.7256 - val_mae: 2.8081
Epoch 60/100
12/12 [==============================] - 0s 10ms/step - loss: 5.4245 - mae: 1.7417 - val_loss: 22.3153 - val_mae: 2.8704
Epoch 61/100
12/12 [==============================] - 0s 9ms/step - loss: 5.4828 - mae: 1.7533 - val_loss: 20.6305 - val_mae: 2.7861
Epoch 62/100
12/12 [==============================] - 0s 10ms/step - loss: 5.4752 - mae: 1.7506 - val_loss: 21.1294 - val_mae: 2.7957
Epoch 63/100
12/12 [==============================] - 0s 12ms/step - loss: 5.7759 - mae: 1.8232 - val_loss: 21.8055 - val_mae: 2.8174
Epoch 64/100
12/12 [==============================] - 0s 9ms/step - loss: 5.1775 - mae: 1.6953 - val_loss: 18.9503 - val_mae: 2.6734
Epoch 65/100
12/12 [==============================] - 0s 10ms/step - loss: 5.1540 - mae: 1.7179 - val_loss: 21.1852 - val_mae: 2.7780
Epoch 66/100
12/12 [==============================] - 0s 10ms/step - loss: 5.1692 - mae: 1.7136 - val_loss: 19.6200 - val_mae: 2.7473
Epoch 67/100
12/12 [==============================] - 0s 10ms/step - loss: 5.0599 - mae: 1.7088 - val_loss: 19.4162 - val_mae: 2.7261
Epoch 68/100
12/12 [==============================] - 0s 15ms/step - loss: 5.4447 - mae: 1.7696 - val_loss: 20.7696 - val_mae: 2.7643
Epoch 69/100
12/12 [==============================] - 0s 12ms/step - loss: 4.9718 - mae: 1.6586 - val_loss: 18.5168 - val_mae: 2.6485
Epoch 70/100
12/12 [==============================] - 0s 9ms/step - loss: 4.7495 - mae: 1.6233 - val_loss: 19.1487 - val_mae: 2.7227
Epoch 71/100
12/12 [==============================] - 0s 12ms/step - loss: 4.7983 - mae: 1.6537 - val_loss: 19.1453 - val_mae: 2.6943
Epoch 72/100
12/12 [==============================] - 0s 10ms/step - loss: 4.6085 - mae: 1.5947 - val_loss: 19.0938 - val_mae: 2.6799
Epoch 73/100
12/12 [==============================] - 0s 10ms/step - loss: 4.4951 - mae: 1.5851 - val_loss: 17.6942 - val_mae: 2.6326
Epoch 74/100
12/12 [==============================] - 0s 14ms/step - loss: 4.5494 - mae: 1.6166 - val_loss: 16.8372 - val_mae: 2.5308
Epoch 75/100
12/12 [==============================] - 0s 11ms/step - loss: 4.5297 - mae: 1.5840 - val_loss: 18.1104 - val_mae: 2.6570
Epoch 76/100
12/12 [==============================] - 0s 9ms/step - loss: 4.4338 - mae: 1.5952 - val_loss: 16.9451 - val_mae: 2.5823
Epoch 77/100
12/12 [==============================] - 0s 13ms/step - loss: 4.4248 - mae: 1.5768 - val_loss: 17.3997 - val_mae: 2.5920
Epoch 78/100
12/12 [==============================] - 0s 9ms/step - loss: 4.3965 - mae: 1.5679 - val_loss: 18.2531 - val_mae: 2.7175
Epoch 79/100
12/12 [==============================] - 0s 9ms/step - loss: 4.2552 - mae: 1.5498 - val_loss: 16.7504 - val_mae: 2.5941
Epoch 80/100
12/12 [==============================] - 0s 14ms/step - loss: 4.1289 - mae: 1.5134 - val_loss: 16.7982 - val_mae: 2.5623
Epoch 81/100
12/12 [==============================] - 0s 10ms/step - loss: 3.9600 - mae: 1.4885 - val_loss: 17.3512 - val_mae: 2.6392
Epoch 82/100
12/12 [==============================] - 0s 9ms/step - loss: 4.1210 - mae: 1.5276 - val_loss: 17.3269 - val_mae: 2.6270
Epoch 83/100
12/12 [==============================] - 0s 10ms/step - loss: 4.0899 - mae: 1.5428 - val_loss: 15.0631 - val_mae: 2.5234
Epoch 84/100
12/12 [==============================] - 0s 9ms/step - loss: 3.9750 - mae: 1.5083 - val_loss: 17.7207 - val_mae: 2.6694
Epoch 85/100
12/12 [==============================] - 0s 13ms/step - loss: 4.1706 - mae: 1.5581 - val_loss: 15.9041 - val_mae: 2.5296
Epoch 86/100
12/12 [==============================] - 0s 9ms/step - loss: 3.9401 - mae: 1.4596 - val_loss: 15.3481 - val_mae: 2.5288
Epoch 87/100
12/12 [==============================] - 0s 10ms/step - loss: 3.7431 - mae: 1.4625 - val_loss: 16.6077 - val_mae: 2.5585
Epoch 88/100
12/12 [==============================] - 0s 11ms/step - loss: 3.8480 - mae: 1.4637 - val_loss: 15.8704 - val_mae: 2.5609
Epoch 89/100
12/12 [==============================] - 0s 11ms/step - loss: 3.8408 - mae: 1.4644 - val_loss: 16.9072 - val_mae: 2.6138
Epoch 90/100
12/12 [==============================] - 0s 9ms/step - loss: 3.8666 - mae: 1.4779 - val_loss: 14.7184 - val_mae: 2.5807
Epoch 91/100
12/12 [==============================] - 0s 12ms/step - loss: 3.6789 - mae: 1.4345 - val_loss: 15.8979 - val_mae: 2.5526
Epoch 92/100
12/12 [==============================] - 0s 15ms/step - loss: 3.6236 - mae: 1.4088 - val_loss: 14.3161 - val_mae: 2.4826
Epoch 93/100
12/12 [==============================] - 0s 13ms/step - loss: 3.5439 - mae: 1.4184 - val_loss: 17.1565 - val_mae: 2.6651
Epoch 94/100
```

```
12/12 [==============================] - 0s 13ms/step - loss: 3.5255 - mae: 1.3985 - val_loss: 14.2670 - val_mae: 2.4286
Epoch 95/100
12/12 [==============================] - 0s 14ms/step - loss: 3.4028 - mae: 1.3592 - val_loss: 16.1651 - val_mae: 2.5888
Epoch 96/100
12/12 [==============================] - 0s 12ms/step - loss: 3.3904 - mae: 1.3630 - val_loss: 14.1217 - val_mae: 2.4847
Epoch 97/100
12/12 [==============================] - 0s 11ms/step - loss: 3.5001 - mae: 1.4016 - val_loss: 14.4484 - val_mae: 2.4992
Epoch 98/100
12/12 [==============================] - 0s 13ms/step - loss: 3.2599 - mae: 1.3252 - val_loss: 14.6851 - val_mae: 2.4873
Epoch 99/100
12/12 [==============================] - 0s 9ms/step - loss: 3.3359 - mae: 1.3434 - val_loss: 13.5714 - val_mae: 2.4654
Epoch 100/100
12/12 [==============================] - 0s 11ms/step - loss: 3.2209 - mae: 1.3004 - val_loss: 14.7126 - val_mae: 2.5417
```

In [18]:

```python
# Evaluation of the model
y_pred_nn = model.predict(X_test)
mse_nn, mae_nn = model.evaluate(X_test, y_test)
```

```
4/4 [==============================] - 0s 6ms/step
4/4 [==============================] - 0s 8ms/step - loss: 10.6461 - mae: 2.1673
```

In [19]:

```python
# Make predictions on new data
new_data = scaler.transform([[0.1, 10.0, 5.0, 0, 0.4, 6.0, 50, 6.0, 1, 400, 20, 300, 10]])
prediction = model.predict(new_data)
```

```
1/1 [==============================] - 0s 55ms/step
```

```
C:\Users\Omkar\AppData\Local\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names, but
StandardScaler was fitted with feature names
  warnings.warn(
```

In [20]:

```python
# Outputting results
print('Linear Regression - Mean squared error on test data:', mse_lr)
print('Linear Regression - Mean absolute error on test data:', mae_lr)
print('Linear Regression - R2 score:', r2_lr)

print('Neural Network - Mean squared error on test data:', mse_nn)
print('Neural Network - Mean absolute error on test data:', mae_nn)

print('Predicted house price:', prediction)
```

```
Linear Regression - Mean squared error on test data: 24.291119474973527
Linear Regression - Mean absolute error on test data: 3.1890919658878483
Linear Regression - R2 score: 0.6687594935356318
Neural Network - Mean squared error on test data: 10.64611530303955
Neural Network - Mean absolute error on test data: 2.167318820953369
Predicted house price: [[15.888279]]
```

In [ ]:

In [ ]: