In [1]:
```
# Importing libraries
import numpy as np
import pandas as pd
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score
import matplotlib.pyplot as plt
```

In [2]:
```
# Loading the Boston Housing dataset
boston_dataset = pd.read_csv('Boston.csv')
boston = pd.DataFrame(boston_dataset, columns=boston_dataset.columns)
boston['MEDV'] = boston_dataset['medv']
```

In [3]:
```
boston_dataset.shape
```

Out[3]:
```
(506, 16)
```

In [4]:
```
print(boston_dataset.head(5))
```

```
   Unnamed: 0     crim    zn  indus  chas    nox     rm   age     dis  rad  \
0           1  0.00632  18.0   2.31     0  0.538  6.575  65.2  4.0900    1
1           2  0.02731   0.0   7.07     0  0.469  6.421  78.9  4.9671    2
2           3  0.02729   0.0   7.07     0  0.469  7.185  61.1  4.9671    2
3           4  0.03237   0.0   2.18     0  0.458  6.998  45.8  6.0622    3
4           5  0.06905   0.0   2.18     0  0.458  7.147  54.2  6.0622    3

   tax  ptratio   black  lstat  medv  MEDV
0  296     15.3  396.90   4.98  24.0  24.0
1  242     17.8  396.90   9.14  21.6  21.6
2  242     17.8  392.83   4.03  34.7  34.7
3  222     18.7  394.63   2.94  33.4  33.4
4  222     18.7  396.90   5.33  36.2  36.2
```

In [5]:
```
print(np.shape(boston_dataset))
```

```
(506, 16)
```

In [6]:
```
print(boston_dataset.describe())
```

```
       Unnamed: 0        crim          zn       indus        chas         nox  \
count  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000
mean   253.500000    3.613524   11.363636   11.136779    0.069170    0.554695
std    146.213884    8.601545   23.322453    6.860353    0.253994    0.115878
min      1.000000    0.006320    0.000000    0.460000    0.000000    0.385000
25%    127.250000    0.082045    0.000000    5.190000    0.000000    0.449000
50%    253.500000    0.256510    0.000000    9.690000    0.000000    0.538000
75%    379.750000    3.677083   12.500000   18.100000    0.000000    0.624000
max    506.000000   88.976200  100.000000   27.740000    1.000000    0.871000

              rm         age         dis         rad         tax     ptratio  \
count  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000
mean     6.284634   68.574901    3.795043    9.549407  408.237154   18.455534
std      0.702617   28.148861    2.105710    8.707259  168.537116    2.164946
min      3.561000    2.900000    1.129600    1.000000  187.000000   12.600000
25%      5.885500   45.025000    2.100175    4.000000  279.000000   17.400000
50%      6.208500   77.500000    3.207450    5.000000  330.000000   19.050000
75%      6.623500   94.075000    5.188425   24.000000  666.000000   20.200000
max      8.780000  100.000000   12.126500   24.000000  711.000000   22.000000

            black       lstat        medv        MEDV
count  506.000000  506.000000  506.000000  506.000000
mean   356.674032   12.653063   22.532806   22.532806
std     91.294864    7.141062    9.197104    9.197104
min      0.320000    1.730000    5.000000    5.000000
25%    375.377500    6.950000   17.025000   17.025000
50%    391.440000   11.360000   21.200000   21.200000
75%    396.225000   16.955000   25.000000   25.000000
max    396.900000   37.970000   50.000000   50.000000
```

In [7]:
```
# Split the data into training and testing sets
X = boston.drop('MEDV', axis=1)
Y = boston['MEDV']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_st
```

In [8]:
```
# Scale the data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [9]:
```
# Define the model
model = tf.keras.models.Sequential([
  tf.keras.layers.Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
  tf.keras.layers.Dense(64, activation='relu'),
  tf.keras.layers.Dense(1)
])
```

In [10]:
```
# Compile the model
model.compile(optimizer='adam', loss='mse')
```

In [11]:
```
# Train the model
history = model.fit(X_train_scaled, Y_train, validation_data=(X_test_scaled, Y_tes
```

```
Epoch 1/100
13/13 [==============================] - 1s 20ms/step - loss: 544.7557 - val_loss:
528.2176
Epoch 2/100
13/13 [==============================] - 0s 6ms/step - loss: 474.2619 - val_loss:
450.4079
Epoch 3/100
13/13 [==============================] - 0s 6ms/step - loss: 386.4602 - val_loss:
351.7784
Epoch 4/100
13/13 [==============================] - 0s 7ms/step - loss: 278.7653 - val_loss:
237.4474
Epoch 5/100
13/13 [==============================] - 0s 6ms/step - loss: 166.2076 - val_loss:
133.5319
Epoch 6/100
13/13 [==============================] - 0s 6ms/step - loss: 88.7835 - val_loss: 7
4.4762
Epoch 7/100
13/13 [==============================] - 0s 7ms/step - loss: 56.3596 - val_loss: 5
0.7624
Epoch 8/100
13/13 [==============================] - 0s 6ms/step - loss: 41.2469 - val_loss: 3
5.8589
Epoch 9/100
13/13 [==============================] - 0s 6ms/step - loss: 29.9458 - val_loss: 2
6.2243
Epoch 10/100
13/13 [==============================] - 0s 5ms/step - loss: 22.9147 - val_loss: 2
0.6316
Epoch 11/100
13/13 [==============================] - 0s 6ms/step - loss: 18.7007 - val_loss: 1
7.2213
Epoch 12/100
13/13 [==============================] - 0s 6ms/step - loss: 15.8012 - val_loss: 1
5.3307
Epoch 13/100
13/13 [==============================] - 0s 6ms/step - loss: 13.9863 - val_loss: 1
3.8547
Epoch 14/100
13/13 [==============================] - 0s 6ms/step - loss: 12.3514 - val_loss: 1
2.7239
Epoch 15/100
13/13 [==============================] - 0s 7ms/step - loss: 11.1191 - val_loss: 1
1.8005
Epoch 16/100
13/13 [==============================] - 0s 6ms/step - loss: 10.0941 - val_loss: 1
0.9234
Epoch 17/100
13/13 [==============================] - 0s 5ms/step - loss: 9.1952 - val_loss: 1
0.0745
Epoch 18/100
13/13 [==============================] - 0s 7ms/step - loss: 8.3820 - val_loss: 9.
3967
Epoch 19/100
13/13 [==============================] - 0s 6ms/step - loss: 7.7622 - val_loss: 8.
8787
Epoch 20/100
13/13 [==============================] - 0s 6ms/step - loss: 7.2135 - val_loss: 8.
2846
Epoch 21/100
13/13 [==============================] - 0s 6ms/step - loss: 6.6660 - val_loss: 7.
6962
Epoch 22/100
```

```
13/13 [==============================] - 0s 6ms/step - loss: 6.2597 - val_loss: 7.
2311
Epoch 23/100
13/13 [==============================] - 0s 6ms/step - loss: 5.8609 - val_loss: 6.
8703
Epoch 24/100
13/13 [==============================] - 0s 6ms/step - loss: 5.5483 - val_loss: 6.
5370
Epoch 25/100
13/13 [==============================] - 0s 6ms/step - loss: 5.2237 - val_loss: 6.
1241
Epoch 26/100
13/13 [==============================] - 0s 6ms/step - loss: 4.9893 - val_loss: 5.
8749
Epoch 27/100
13/13 [==============================] - 0s 6ms/step - loss: 4.7292 - val_loss: 5.
5789
Epoch 28/100
13/13 [==============================] - 0s 5ms/step - loss: 4.5355 - val_loss: 5.
4220
Epoch 29/100
13/13 [==============================] - 0s 6ms/step - loss: 4.3526 - val_loss: 5.
1536
Epoch 30/100
13/13 [==============================] - 0s 6ms/step - loss: 4.1909 - val_loss: 4.
9802
Epoch 31/100
13/13 [==============================] - 0s 6ms/step - loss: 4.0084 - val_loss: 4.
7379
Epoch 32/100
13/13 [==============================] - 0s 7ms/step - loss: 3.8600 - val_loss: 4.
6386
Epoch 33/100
13/13 [==============================] - 0s 7ms/step - loss: 3.7354 - val_loss: 4.
4787
Epoch 34/100
13/13 [==============================] - 0s 6ms/step - loss: 3.5799 - val_loss: 4.
2570
Epoch 35/100
13/13 [==============================] - 0s 5ms/step - loss: 3.5040 - val_loss: 4.
1388
Epoch 36/100
13/13 [==============================] - 0s 6ms/step - loss: 3.3625 - val_loss: 3.
9860
Epoch 37/100
13/13 [==============================] - 0s 6ms/step - loss: 3.2475 - val_loss: 3.
8835
Epoch 38/100
13/13 [==============================] - 0s 6ms/step - loss: 3.1808 - val_loss: 3.
7203
Epoch 39/100
13/13 [==============================] - 0s 6ms/step - loss: 3.0265 - val_loss: 3.
6454
Epoch 40/100
13/13 [==============================] - 0s 6ms/step - loss: 2.9619 - val_loss: 3.
5450
Epoch 41/100
13/13 [==============================] - 0s 6ms/step - loss: 2.8546 - val_loss: 3.
4093
Epoch 42/100
13/13 [==============================] - 0s 5ms/step - loss: 2.7699 - val_loss: 3.
3024
Epoch 43/100
13/13 [==============================] - 0s 6ms/step - loss: 2.6705 - val_loss: 3.
```

```
2194
Epoch 44/100
13/13 [==============================] - 0s 6ms/step - loss: 2.5973 - val_loss: 3.
0859
Epoch 45/100
13/13 [==============================] - 0s 6ms/step - loss: 2.5165 - val_loss: 3.
0507
Epoch 46/100
13/13 [==============================] - 0s 7ms/step - loss: 2.4570 - val_loss: 2.
9426
Epoch 47/100
13/13 [==============================] - 0s 6ms/step - loss: 2.3949 - val_loss: 2.
8561
Epoch 48/100
13/13 [==============================] - 0s 5ms/step - loss: 2.3214 - val_loss: 2.
7287
Epoch 49/100
13/13 [==============================] - 0s 6ms/step - loss: 2.2469 - val_loss: 2.
7258
Epoch 50/100
13/13 [==============================] - 0s 6ms/step - loss: 2.1974 - val_loss: 2.
6972
Epoch 51/100
13/13 [==============================] - 0s 6ms/step - loss: 2.1071 - val_loss: 2.
5647
Epoch 52/100
13/13 [==============================] - 0s 7ms/step - loss: 2.0570 - val_loss: 2.
4770
Epoch 53/100
13/13 [==============================] - 0s 6ms/step - loss: 2.0121 - val_loss: 2.
4425
Epoch 54/100
13/13 [==============================] - 0s 6ms/step - loss: 1.9369 - val_loss: 2.
3968
Epoch 55/100
13/13 [==============================] - 0s 5ms/step - loss: 1.8858 - val_loss: 2.
3230
Epoch 56/100
13/13 [==============================] - 0s 5ms/step - loss: 1.8329 - val_loss: 2.
2876
Epoch 57/100
13/13 [==============================] - 0s 5ms/step - loss: 1.7771 - val_loss: 2.
2050
Epoch 58/100
13/13 [==============================] - 0s 5ms/step - loss: 1.7226 - val_loss: 2.
1694
Epoch 59/100
13/13 [==============================] - 0s 5ms/step - loss: 1.6924 - val_loss: 2.
0998
Epoch 60/100
13/13 [==============================] - 0s 5ms/step - loss: 1.6443 - val_loss: 2.
0467
Epoch 61/100
13/13 [==============================] - 0s 5ms/step - loss: 1.5914 - val_loss: 2.
0238
Epoch 62/100
13/13 [==============================] - 0s 5ms/step - loss: 1.5386 - val_loss: 1.
9952
Epoch 63/100
13/13 [==============================] - 0s 5ms/step - loss: 1.5007 - val_loss: 1.
9350
Epoch 64/100
13/13 [==============================] - 0s 5ms/step - loss: 1.4812 - val_loss: 1.
8878
```

```
Epoch 65/100
13/13 [==============================] - 0s 5ms/step - loss: 1.4316 - val_loss: 1.
8689
Epoch 66/100
13/13 [==============================] - 0s 5ms/step - loss: 1.3845 - val_loss: 1.
8155
Epoch 67/100
13/13 [==============================] - 0s 5ms/step - loss: 1.3612 - val_loss: 1.
7486
Epoch 68/100
13/13 [==============================] - 0s 5ms/step - loss: 1.3136 - val_loss: 1.
7383
Epoch 69/100
13/13 [==============================] - 0s 5ms/step - loss: 1.2854 - val_loss: 1.
7089
Epoch 70/100
13/13 [==============================] - 0s 5ms/step - loss: 1.2479 - val_loss: 1.
6705
Epoch 71/100
13/13 [==============================] - 0s 6ms/step - loss: 1.2126 - val_loss: 1.
6209
Epoch 72/100
13/13 [==============================] - 0s 5ms/step - loss: 1.1965 - val_loss: 1.
6077
Epoch 73/100
13/13 [==============================] - 0s 5ms/step - loss: 1.1602 - val_loss: 1.
5491
Epoch 74/100
13/13 [==============================] - 0s 5ms/step - loss: 1.1356 - val_loss: 1.
5389
Epoch 75/100
13/13 [==============================] - 0s 5ms/step - loss: 1.0942 - val_loss: 1.
4873
Epoch 76/100
13/13 [==============================] - 0s 5ms/step - loss: 1.0611 - val_loss: 1.
4602
Epoch 77/100
13/13 [==============================] - 0s 5ms/step - loss: 1.0367 - val_loss: 1.
4562
Epoch 78/100
13/13 [==============================] - 0s 5ms/step - loss: 1.0096 - val_loss: 1.
4292
Epoch 79/100
13/13 [==============================] - 0s 5ms/step - loss: 0.9759 - val_loss: 1.
3894
Epoch 80/100
13/13 [==============================] - 0s 5ms/step - loss: 0.9548 - val_loss: 1.
3679
Epoch 81/100
13/13 [==============================] - 0s 5ms/step - loss: 0.9433 - val_loss: 1.
3508
Epoch 82/100
13/13 [==============================] - 0s 5ms/step - loss: 0.9110 - val_loss: 1.
3194
Epoch 83/100
13/13 [==============================] - 0s 5ms/step - loss: 0.8921 - val_loss: 1.
2953
Epoch 84/100
13/13 [==============================] - 0s 5ms/step - loss: 0.8668 - val_loss: 1.
2811
Epoch 85/100
13/13 [==============================] - 0s 5ms/step - loss: 0.8610 - val_loss: 1.
2553
Epoch 86/100
```

```
13/13 [==============================] - 0s 5ms/step - loss: 0.8586 - val_loss: 1.
2562
Epoch 87/100
13/13 [==============================] - 0s 5ms/step - loss: 0.8326 - val_loss: 1.
2146
Epoch 88/100
13/13 [==============================] - 0s 5ms/step - loss: 0.8056 - val_loss: 1.
2069
Epoch 89/100
13/13 [==============================] - 0s 5ms/step - loss: 0.7737 - val_loss: 1.
2073
Epoch 90/100
13/13 [==============================] - 0s 6ms/step - loss: 0.7694 - val_loss: 1.
1611
Epoch 91/100
13/13 [==============================] - 0s 6ms/step - loss: 0.7533 - val_loss: 1.
1673
Epoch 92/100
13/13 [==============================] - 0s 5ms/step - loss: 0.7296 - val_loss: 1.
1512
Epoch 93/100
13/13 [==============================] - 0s 5ms/step - loss: 0.7264 - val_loss: 1.
1182
Epoch 94/100
13/13 [==============================] - 0s 5ms/step - loss: 0.7041 - val_loss: 1.
1141
Epoch 95/100
13/13 [==============================] - 0s 5ms/step - loss: 0.6780 - val_loss: 1.
0933
Epoch 96/100
13/13 [==============================] - 0s 5ms/step - loss: 0.6640 - val_loss: 1.
0887
Epoch 97/100
13/13 [==============================] - 0s 5ms/step - loss: 0.6420 - val_loss: 1.
0658
Epoch 98/100
13/13 [==============================] - 0s 5ms/step - loss: 0.6284 - val_loss: 1.
0482
Epoch 99/100
13/13 [==============================] - 0s 5ms/step - loss: 0.6183 - val_loss: 1.
0343
Epoch 100/100
13/13 [==============================] - 0s 5ms/step - loss: 0.6055 - val_loss: 1.
0332
```

In [12]:

```
# Evaluate the model
Y_pred = model.predict(X_test_scaled)
r2 = r2_score(Y_test, Y_pred)
print("R^2 score:", r2)
```

```
4/4 [==============================] - 0s 4ms/step
R^2 score: 0.9895458804730384
```

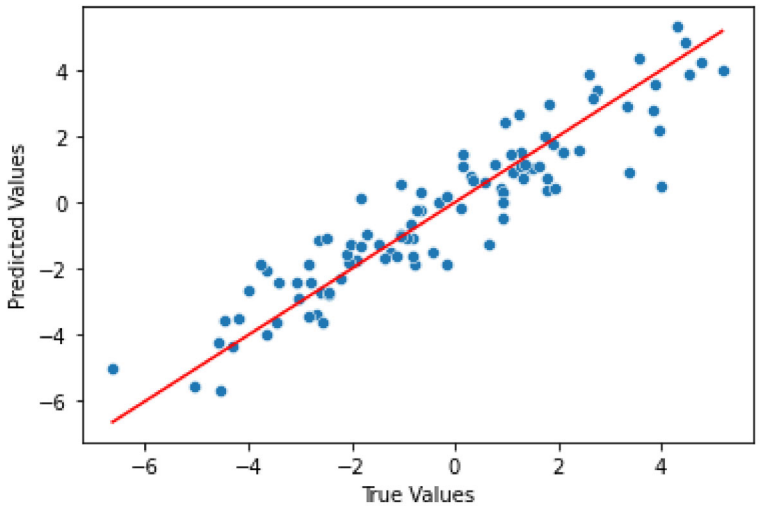
In [13]:

```
import numpy as np
import seaborn as sns

# Generate some sample data
X = np.random.normal(0, 1, 100)
Y = 2 * X + np.random.normal(0, 1, 100)

# Fit a linear regression model
model = np.polyfit(X, Y, 1)

# Make predictions on the training data
```

```
Y_pred = np.polyval(model, X)

# Add axis labels
plt.xlabel('True Values')
plt.ylabel('Predicted Values')

# Create a scatter plot of predicted vs true values
sns.scatterplot(np.squeeze(Y), np.squeeze(Y_pred))

# Add a diagonal line to show perfect correlation
sns.lineplot(np.squeeze(Y), np.squeeze(Y), color='red')
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarnin
g: Pass the following variables as keyword args: x, y. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an e
xplicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarnin
g: Pass the following variables as keyword args: x, y. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an e
xplicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[13]:
```
<AxesSubplot:xlabel='True Values', ylabel='Predicted Values'>
```



In [ ]: