In [ ]:
```
# 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
```

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

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

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

```
(506, 16)
```

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

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

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

```
Epoch 1/100
13/13 [==============================] - 2s 22ms/step - loss: 579.7935 - val_loss: 
566.8345
Epoch 2/100
13/13 [==============================] - 0s 4ms/step - loss: 519.9058 - val_loss: 
499.8945
Epoch 3/100
13/13 [==============================] - 0s 4ms/step - loss: 446.0014 - val_loss: 
410.6792
Epoch 4/100
13/13 [==============================] - 0s 4ms/step - loss: 346.0971 - val_loss: 
295.0309
Epoch 5/100
13/13 [==============================] - 0s 4ms/step - loss: 228.5537 - val_loss: 
172.4942
Epoch 6/100
13/13 [==============================] - 0s 4ms/step - loss: 117.2006 - val_loss: 
86.2677
Epoch 7/100
13/13 [==============================] - 0s 4ms/step - loss: 60.7959 - val_loss: 5
0.8605
Epoch 8/100
13/13 [==============================] - 0s 4ms/step - loss: 41.9353 - val_loss: 3
7.0392
Epoch 9/100
13/13 [==============================] - 0s 4ms/step - loss: 31.4906 - val_loss: 2
7.4767
Epoch 10/100
13/13 [==============================] - 0s 3ms/step - loss: 24.2087 - val_loss: 2
1.7079
Epoch 11/100
13/13 [==============================] - 0s 4ms/step - loss: 19.9854 - val_loss: 1
8.1820
Epoch 12/100
13/13 [==============================] - 0s 4ms/step - loss: 17.1946 - val_loss: 1
5.7325
Epoch 13/100
13/13 [==============================] - 0s 4ms/step - loss: 15.0696 - val_loss: 1
4.0107
Epoch 14/100
13/13 [==============================] - 0s 4ms/step - loss: 13.4463 - val_loss: 1
2.7971
Epoch 15/100
13/13 [==============================] - 0s 4ms/step - loss: 12.3516 - val_loss: 1
1.7113
Epoch 16/100
13/13 [==============================] - 0s 4ms/step - loss: 11.1735 - val_loss: 1
0.7977
Epoch 17/100
13/13 [==============================] - 0s 4ms/step - loss: 10.3415 - val_loss: 1
0.2638
Epoch 18/100
13/13 [==============================] - 0s 4ms/step - loss: 9.5420 - val_loss: 9.
5540
Epoch 19/100
13/13 [==============================] - 0s 4ms/step - loss: 8.9082 - val_loss: 9.
0734
Epoch 20/100
13/13 [==============================] - 0s 4ms/step - loss: 8.3447 - val_loss: 8.
7130
Epoch 21/100
13/13 [==============================] - 0s 4ms/step - loss: 7.7903 - val_loss: 8.
1594
Epoch 22/100
```

```
13/13 [==============================] - 0s 4ms/step - loss: 7.3412 - val_loss: 7.
7143
Epoch 23/100
13/13 [==============================] - 0s 4ms/step - loss: 6.9347 - val_loss: 7.
3838
Epoch 24/100
13/13 [==============================] - 0s 4ms/step - loss: 6.5723 - val_loss: 7.
0898
Epoch 25/100
13/13 [==============================] - 0s 4ms/step - loss: 6.2431 - val_loss: 6.
6430
Epoch 26/100
13/13 [==============================] - 0s 4ms/step - loss: 5.9225 - val_loss: 6.
4520
Epoch 27/100
13/13 [==============================] - 0s 4ms/step - loss: 5.6786 - val_loss: 6.
1964
Epoch 28/100
13/13 [==============================] - 0s 4ms/step - loss: 5.3824 - val_loss: 5.
9290
Epoch 29/100
13/13 [==============================] - 0s 3ms/step - loss: 5.1993 - val_loss: 5.
7668
Epoch 30/100
13/13 [==============================] - 0s 4ms/step - loss: 4.9399 - val_loss: 5.
5288
Epoch 31/100
13/13 [==============================] - 0s 3ms/step - loss: 4.7501 - val_loss: 5.
3834
Epoch 32/100
13/13 [==============================] - 0s 4ms/step - loss: 4.5536 - val_loss: 5.
1878
Epoch 33/100
13/13 [==============================] - 0s 4ms/step - loss: 4.3981 - val_loss: 5.
0531
Epoch 34/100
13/13 [==============================] - 0s 4ms/step - loss: 4.2246 - val_loss: 4.
8554
Epoch 35/100
13/13 [==============================] - 0s 4ms/step - loss: 4.1005 - val_loss: 4.
7853
Epoch 36/100
13/13 [==============================] - 0s 4ms/step - loss: 3.9351 - val_loss: 4.
5902
Epoch 37/100
13/13 [==============================] - 0s 4ms/step - loss: 3.7831 - val_loss: 4.
4526
Epoch 38/100
13/13 [==============================] - 0s 4ms/step - loss: 3.6974 - val_loss: 4.
3520
Epoch 39/100
13/13 [==============================] - 0s 4ms/step - loss: 3.5651 - val_loss: 4.
1889
Epoch 40/100
13/13 [==============================] - 0s 4ms/step - loss: 3.4270 - val_loss: 4.
0594
Epoch 41/100
13/13 [==============================] - 0s 4ms/step - loss: 3.3219 - val_loss: 3.
9709
Epoch 42/100
13/13 [==============================] - 0s 4ms/step - loss: 3.2117 - val_loss: 3.
8503
Epoch 43/100
13/13 [==============================] - 0s 4ms/step - loss: 3.0983 - val_loss: 3.
```

```
7659
Epoch 44/100
13/13 [==============================] - 0s 4ms/step - loss: 3.0200 - val_loss: 3.
6303
Epoch 45/100
13/13 [==============================] - 0s 4ms/step - loss: 3.0036 - val_loss: 3.
6176
Epoch 46/100
13/13 [==============================] - 0s 4ms/step - loss: 2.8331 - val_loss: 3.
4714
Epoch 47/100
13/13 [==============================] - 0s 4ms/step - loss: 2.7135 - val_loss: 3.
4201
Epoch 48/100
13/13 [==============================] - 0s 4ms/step - loss: 2.6206 - val_loss: 3.
2971
Epoch 49/100
13/13 [==============================] - 0s 4ms/step - loss: 2.5344 - val_loss: 3.
2256
Epoch 50/100
13/13 [==============================] - 0s 4ms/step - loss: 2.4815 - val_loss: 3.
1975
Epoch 51/100
13/13 [==============================] - 0s 4ms/step - loss: 2.3792 - val_loss: 3.
0637
Epoch 52/100
13/13 [==============================] - 0s 4ms/step - loss: 2.3216 - val_loss: 3.
0033
Epoch 53/100
13/13 [==============================] - 0s 4ms/step - loss: 2.2419 - val_loss: 2.
9665
Epoch 54/100
13/13 [==============================] - 0s 4ms/step - loss: 2.1967 - val_loss: 2.
8615
Epoch 55/100
13/13 [==============================] - 0s 4ms/step - loss: 2.1231 - val_loss: 2.
8480
Epoch 56/100
13/13 [==============================] - 0s 3ms/step - loss: 2.0490 - val_loss: 2.
7432
Epoch 57/100
13/13 [==============================] - 0s 4ms/step - loss: 1.9867 - val_loss: 2.
6608
Epoch 58/100
13/13 [==============================] - 0s 4ms/step - loss: 1.9340 - val_loss: 2.
6377
Epoch 59/100
13/13 [==============================] - 0s 4ms/step - loss: 1.8673 - val_loss: 2.
5568
Epoch 60/100
13/13 [==============================] - 0s 4ms/step - loss: 1.8072 - val_loss: 2.
4932
Epoch 61/100
13/13 [==============================] - 0s 4ms/step - loss: 1.7626 - val_loss: 2.
4221
Epoch 62/100
13/13 [==============================] - 0s 4ms/step - loss: 1.7002 - val_loss: 2.
3837
Epoch 63/100
13/13 [==============================] - 0s 3ms/step - loss: 1.6564 - val_loss: 2.
3385
Epoch 64/100
13/13 [==============================] - 0s 4ms/step - loss: 1.6168 - val_loss: 2.
2838
```

```
Epoch 65/100
13/13 [==============================] - 0s 4ms/step - loss: 1.5665 - val_loss: 2.
2466
Epoch 66/100
13/13 [==============================] - 0s 3ms/step - loss: 1.5442 - val_loss: 2.
1940
Epoch 67/100
13/13 [==============================] - 0s 3ms/step - loss: 1.4927 - val_loss: 2.
1336
Epoch 68/100
13/13 [==============================] - 0s 4ms/step - loss: 1.4522 - val_loss: 2.
1053
Epoch 69/100
13/13 [==============================] - 0s 4ms/step - loss: 1.4215 - val_loss: 2.
0470
Epoch 70/100
13/13 [==============================] - 0s 5ms/step - loss: 1.3719 - val_loss: 2.
0393
Epoch 71/100
13/13 [==============================] - 0s 3ms/step - loss: 1.3441 - val_loss: 1.
9835
Epoch 72/100
13/13 [==============================] - 0s 3ms/step - loss: 1.3089 - val_loss: 1.
9623
Epoch 73/100
13/13 [==============================] - 0s 4ms/step - loss: 1.2761 - val_loss: 1.
8952
Epoch 74/100
13/13 [==============================] - 0s 4ms/step - loss: 1.2514 - val_loss: 1.
8818
Epoch 75/100
13/13 [==============================] - 0s 4ms/step - loss: 1.2050 - val_loss: 1.
8480
Epoch 76/100
13/13 [==============================] - 0s 4ms/step - loss: 1.1727 - val_loss: 1.
8131
Epoch 77/100
13/13 [==============================] - 0s 4ms/step - loss: 1.1533 - val_loss: 1.
7807
Epoch 78/100
13/13 [==============================] - 0s 4ms/step - loss: 1.1223 - val_loss: 1.
7512
Epoch 79/100
13/13 [==============================] - 0s 4ms/step - loss: 1.1093 - val_loss: 1.
7289
Epoch 80/100
13/13 [==============================] - 0s 4ms/step - loss: 1.0749 - val_loss: 1.
6943
Epoch 81/100
13/13 [==============================] - 0s 4ms/step - loss: 1.0587 - val_loss: 1.
6859
Epoch 82/100
13/13 [==============================] - 0s 4ms/step - loss: 1.0236 - val_loss: 1.
6370
Epoch 83/100
13/13 [==============================] - 0s 4ms/step - loss: 0.9885 - val_loss: 1.
6404
Epoch 84/100
13/13 [==============================] - 0s 4ms/step - loss: 0.9649 - val_loss: 1.
5829
Epoch 85/100
13/13 [==============================] - 0s 4ms/step - loss: 0.9514 - val_loss: 1.
5622
Epoch 86/100
```

```
13/13 [==============================] - 0s 4ms/step - loss: 0.9222 - val_loss: 1.
5437
Epoch 87/100
13/13 [==============================] - 0s 4ms/step - loss: 0.9147 - val_loss: 1.
5493
Epoch 88/100
13/13 [==============================] - 0s 4ms/step - loss: 0.8950 - val_loss: 1.
5184
Epoch 89/100
13/13 [==============================] - 0s 3ms/step - loss: 0.8701 - val_loss: 1.
4658
Epoch 90/100
13/13 [==============================] - 0s 4ms/step - loss: 0.8495 - val_loss: 1.
5099
Epoch 91/100
13/13 [==============================] - 0s 3ms/step - loss: 0.8492 - val_loss: 1.
4326
Epoch 92/100
13/13 [==============================] - 0s 3ms/step - loss: 0.8271 - val_loss: 1.
4211
Epoch 93/100
13/13 [==============================] - 0s 4ms/step - loss: 0.7996 - val_loss: 1.
3989
Epoch 94/100
13/13 [==============================] - 0s 3ms/step - loss: 0.7729 - val_loss: 1.
3790
Epoch 95/100
13/13 [==============================] - 0s 3ms/step - loss: 0.7562 - val_loss: 1.
3617
Epoch 96/100
13/13 [==============================] - 0s 4ms/step - loss: 0.7391 - val_loss: 1.
3292
Epoch 97/100
13/13 [==============================] - 0s 4ms/step - loss: 0.7235 - val_loss: 1.
3122
Epoch 98/100
13/13 [==============================] - 0s 3ms/step - loss: 0.7109 - val_loss: 1.
3096
Epoch 99/100
13/13 [==============================] - 0s 4ms/step - loss: 0.6938 - val_loss: 1.
2771
Epoch 100/100
13/13 [==============================] - 0s 4ms/step - loss: 0.6795 - val_loss: 1.
2723
```

In [ ]:

```
# 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 2ms/step
R^2 score: 0.987125595222069
```

In [ ]:

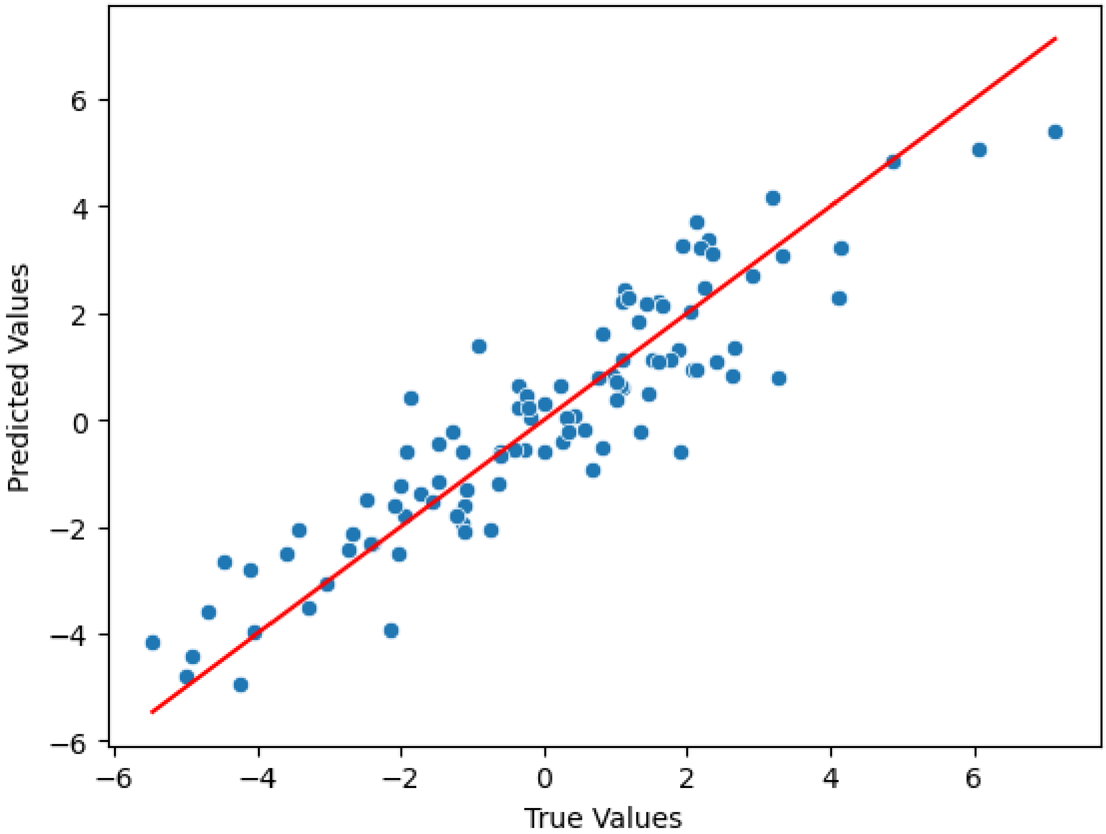
```
import numpy as np
import seaborn as sns

# 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:\Users\D_COMP_RSL-14\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: 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 wit
hout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
C:\Users\D_COMP_RSL-14\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futu
reWarning: 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 wit
hout an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[ ]:

```
<AxesSubplot:xlabel='True Values', ylabel='Predicted Values'>
```



In [ ]: