Assignment 1

In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
data = pd.read_csv("/Users/devarshimahajan/Desktop/college/HousingData.csv")
```

In [3]:

```
data.head()
```

Out[3]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90

In [4]:

```
data.isnull().sum()
```

Out[4]:

```
CRIM       20
ZN         20
INDUS      20
CHAS       20
NOX         0
RM          0
AGE        20
DIS         0
RAD         0
TAX         0
PTRATIO     0
B           0
LSTAT      20
MEDV        0
dtype: int64
```

In [5]:

```
data['CRIM'] = data['CRIM'].fillna(data['CRIM'].mean())
data['ZN'] = data['ZN'].fillna(data['ZN'].mean())
data['INDUS'] = data['INDUS'].fillna(data['INDUS'].mean())
data['CHAS'] = data['CHAS'].fillna(data['CHAS'].mean())
data['AGE'] = data['AGE'].fillna(data['AGE'].mean())
data['LSTAT'] = data['LSTAT'].fillna(data['LSTAT'].mean())
```

In [6]:

```
data.describe()
```

Out[6]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	A
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000
mean	3.611874	11.211934	11.083992	0.069959	0.554695	6.284634	68.518
std	8.545770	22.921051	6.699165	0.250233	0.115878	0.702617	27.439
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900
25%	0.083235	0.000000	5.190000	0.000000	0.449000	5.885500	45.925
50%	0.290250	0.000000	9.900000	0.000000	0.538000	6.208500	74.450
75%	3.611874	11.211934	18.100000	0.000000	0.624000	6.623500	93.575
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000

In [7]:

```
data.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    float64
 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(12), int64(2)
memory usage: 55.5 KB
```

In [8]:

```
X = data.iloc[:,:-1]
y = data.MEDV
```

In [9]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.2, ra
```

In [10]:

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
```

In [11]:

```
regressor.fit(X_train,y_train)
```

Out[11]:

```
▾ LinearRegression
LinearRegression()
```

In [12]:

```
y_pred = regressor.predict(X_test)
```

In [13]:

```
from sklearn.metrics import mean_squared_error
rmse = (np.sqrt(mean_squared_error(y_test, y_pred)))
print(rmse)
```

```
5.001766890194151
```

In [14]:

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

In [15]:

```
import keras
from keras.layers import Dense, Activation,Dropout
from keras.models import Sequential
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))
model.compile(optimizer = 'adam',loss = 'mean_squared_error')
```

In [16]:

```
model.fit(X_train, y_train, epochs = 100)
```

```
Epoch 1/100
13/13 [==============================] - 0s 1ms/step - loss: 590.2822
Epoch 2/100
13/13 [==============================] - 0s 1ms/step - loss: 517.5150
Epoch 3/100
13/13 [==============================] - 0s 1ms/step - loss: 346.3068
Epoch 4/100
13/13 [==============================] - 0s 806us/step - loss: 123.0947
Epoch 5/100
13/13 [==============================] - 0s 752us/step - loss: 60.7781
Epoch 6/100
13/13 [==============================] - 0s 947us/step - loss: 33.5860
Epoch 7/100
13/13 [==============================] - 0s 759us/step - loss: 26.8214
Epoch 8/100
13/13 [==============================] - 0s 975us/step - loss: 23.9563
Epoch 9/100
13/13 [==============================] - 0s 868us/step - loss: 21.4681
Epoch 10/100
13/13 [==============================] - 0s 866us/step - loss: 20.0007
Epoch 11/100
13/13 [==============================] - 0s 908us/step - loss: 18.8111
Epoch 12/100
13/13 [==============================] - 0s 836us/step - loss: 17.7034
Epoch 13/100
13/13 [==============================] - 0s 958us/step - loss: 16.7886
Epoch 14/100
13/13 [==============================] - 0s 920us/step - loss: 16.0094
Epoch 15/100
13/13 [==============================] - 0s 905us/step - loss: 14.9987
Epoch 16/100
13/13 [==============================] - 0s 875us/step - loss: 14.3688
Epoch 17/100
13/13 [==============================] - 0s 876us/step - loss: 13.8798
Epoch 18/100
13/13 [==============================] - 0s 985us/step - loss: 13.5313
Epoch 19/100
13/13 [==============================] - 0s 895us/step - loss: 12.9561
Epoch 20/100
13/13 [==============================] - 0s 898us/step - loss: 12.5246
Epoch 21/100
13/13 [==============================] - 0s 867us/step - loss: 12.1538
Epoch 22/100
13/13 [==============================] - 0s 848us/step - loss: 11.9327
Epoch 23/100
13/13 [==============================] - 0s 951us/step - loss: 11.8974
Epoch 24/100
13/13 [==============================] - 0s 872us/step - loss: 11.5242
Epoch 25/100
13/13 [==============================] - 0s 870us/step - loss: 11.2393
Epoch 26/100
13/13 [==============================] - 0s 876us/step - loss: 11.0092
Epoch 27/100
13/13 [==============================] - 0s 839us/step - loss: 10.6930
Epoch 28/100
13/13 [==============================] - 0s 1ms/step - loss: 10.4720
Epoch 29/100
13/13 [==============================] - 0s 890us/step - loss: 10.4307
Epoch 30/100
13/13 [==============================] - 0s 863us/step - loss: 9.8594
Epoch 31/100
13/13 [==============================] - 0s 872us/step - loss: 9.8081
Epoch 32/100
13/13 [==============================] - 0s 841us/step - loss: 9.5584
```

```
Epoch 33/100
13/13 [==============================] - 0s 928us/step - loss: 9.3505
Epoch 34/100
13/13 [==============================] - 0s 871us/step - loss: 9.6004
Epoch 35/100
13/13 [==============================] - 0s 867us/step - loss: 9.1988
Epoch 36/100
13/13 [==============================] - 0s 947us/step - loss: 9.1919
Epoch 37/100
13/13 [==============================] - 0s 874us/step - loss: 8.9034
Epoch 38/100
13/13 [==============================] - 0s 940us/step - loss: 8.5484
Epoch 39/100
13/13 [==============================] - 0s 846us/step - loss: 8.3108
Epoch 40/100
13/13 [==============================] - 0s 874us/step - loss: 8.1770
Epoch 41/100
13/13 [==============================] - 0s 970us/step - loss: 8.3204
Epoch 42/100
13/13 [==============================] - 0s 850us/step - loss: 8.0053
Epoch 43/100
13/13 [==============================] - 0s 1ms/step - loss: 7.8735
Epoch 44/100
13/13 [==============================] - 0s 884us/step - loss: 7.9175
Epoch 45/100
13/13 [==============================] - 0s 908us/step - loss: 7.5655
Epoch 46/100
13/13 [==============================] - 0s 2ms/step - loss: 7.3229
Epoch 47/100
13/13 [==============================] - 0s 956us/step - loss: 7.3337
Epoch 48/100
13/13 [==============================] - 0s 851us/step - loss: 7.2201
Epoch 49/100
13/13 [==============================] - 0s 872us/step - loss: 7.2482
Epoch 50/100
13/13 [==============================] - 0s 945us/step - loss: 6.9275
Epoch 51/100
13/13 [==============================] - 0s 844us/step - loss: 6.8826
Epoch 52/100
13/13 [==============================] - 0s 1ms/step - loss: 6.7524
Epoch 53/100
13/13 [==============================] - 0s 871us/step - loss: 6.5843
Epoch 54/100
13/13 [==============================] - 0s 859us/step - loss: 6.3108
Epoch 55/100
13/13 [==============================] - 0s 913us/step - loss: 6.5669
Epoch 56/100
13/13 [==============================] - 0s 838us/step - loss: 6.3247
Epoch 57/100
13/13 [==============================] - 0s 938us/step - loss: 6.0976
Epoch 58/100
13/13 [==============================] - 0s 855us/step - loss: 5.9578
Epoch 59/100
13/13 [==============================] - 0s 861us/step - loss: 6.0718
Epoch 60/100
13/13 [==============================] - 0s 862us/step - loss: 5.6460
Epoch 61/100
13/13 [==============================] - 0s 831us/step - loss: 5.5016
Epoch 62/100
13/13 [==============================] - 0s 894us/step - loss: 5.6344
Epoch 63/100
13/13 [==============================] - 0s 861us/step - loss: 5.4458
Epoch 64/100
13/13 [==============================] - 0s 837us/step - loss: 5.3982
```

```
Epoch 65/100
13/13 [==============================] - 0s 974us/step - loss: 5.2253
Epoch 66/100
13/13 [==============================] - 0s 830us/step - loss: 5.1140
Epoch 67/100
13/13 [==============================] - 0s 886us/step - loss: 5.0990
Epoch 68/100
13/13 [==============================] - 0s 938us/step - loss: 5.3324
Epoch 69/100
13/13 [==============================] - 0s 1ms/step - loss: 4.9519
Epoch 70/100
13/13 [==============================] - 0s 898us/step - loss: 4.8477
Epoch 71/100
13/13 [==============================] - 0s 991us/step - loss: 4.9727
Epoch 72/100
13/13 [==============================] - 0s 1ms/step - loss: 4.8309
Epoch 73/100
13/13 [==============================] - 0s 938us/step - loss: 5.1362
Epoch 74/100
13/13 [==============================] - 0s 1ms/step - loss: 4.9917
Epoch 75/100
13/13 [==============================] - 0s 864us/step - loss: 4.6516
Epoch 76/100
13/13 [==============================] - 0s 936us/step - loss: 4.4377
Epoch 77/100
13/13 [==============================] - 0s 997us/step - loss: 4.2724
Epoch 78/100
13/13 [==============================] - 0s 836us/step - loss: 4.3135
Epoch 79/100
13/13 [==============================] - 0s 921us/step - loss: 4.1679
Epoch 80/100
13/13 [==============================] - 0s 901us/step - loss: 4.1143
Epoch 81/100
13/13 [==============================] - 0s 887us/step - loss: 4.0359
Epoch 82/100
13/13 [==============================] - 0s 846us/step - loss: 4.1357
Epoch 83/100
13/13 [==============================] - 0s 804us/step - loss: 4.2562
Epoch 84/100
13/13 [==============================] - 0s 831us/step - loss: 3.8435
Epoch 85/100
13/13 [==============================] - 0s 830us/step - loss: 3.8222
Epoch 86/100
13/13 [==============================] - 0s 804us/step - loss: 4.0099
Epoch 87/100
13/13 [==============================] - 0s 931us/step - loss: 3.8272
Epoch 88/100
13/13 [==============================] - 0s 832us/step - loss: 3.6474
Epoch 89/100
13/13 [==============================] - 0s 771us/step - loss: 3.5514
Epoch 90/100
13/13 [==============================] - 0s 1ms/step - loss: 3.4363
Epoch 91/100
13/13 [==============================] - 0s 866us/step - loss: 3.4455
Epoch 92/100
13/13 [==============================] - 0s 964us/step - loss: 3.8596
Epoch 93/100
13/13 [==============================] - 0s 850us/step - loss: 3.6106
Epoch 94/100
13/13 [==============================] - 0s 831us/step - loss: 3.6345
Epoch 95/100
13/13 [==============================] - 0s 917us/step - loss: 3.5166
Epoch 96/100
13/13 [==============================] - 0s 818us/step - loss: 3.2813
```

```
Epoch 97/100
13/13 [==============================] - 0s 867us/step - loss: 3.3265
Epoch 98/100
13/13 [==============================] - 0s 841us/step - loss: 3.2840
Epoch 99/100
13/13 [==============================] - 0s 817us/step - loss: 3.3830
Epoch 100/100
13/13 [==============================] - 0s 936us/step - loss: 3.1371
```

Out[16]:

```
<keras.src.callbacks.History at 0x178d1b760>
```

In [17]:

```python
y_pred = model.predict(X_test)
```

```
4/4 [==============================] - 0s 974us/step
```

In [18]:

```python
from sklearn.metrics import mean_squared_error
rmse = (np.sqrt(mean_squared_error(y_test, y_pred)))
print(rmse)
```

```
3.534544173568633
```