Data Analytics I Create a Linear Regression Model using Python/R to predict home prices using Boston Housing Dataset (https://www.kaggle.com/c/boston-housing). The Boston Housing dataset contains information about various houses in Boston through different parameters. There are 506 samples and 14 feature variables in this dataset.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df = pd.read_csv('housing.csv')
df
```

	RM	LSTAT	PTRATIO	MEDV
0	6.575	4.98	15.3	504000.0
1	6.421	9.14	17.8	453600.0
2	7.185	4.03	17.8	728700.0
3	6.998	2.94	18.7	701400.0
4	7.147	5.33	18.7	760200.0
...	...	...	...	...
484	6.593	9.67	21.0	470400.0
485	6.120	9.08	21.0	432600.0
486	6.976	5.64	21.0	501900.0
487	6.794	6.48	21.0	462000.0
488	6.030	7.88	21.0	249900.0

Saved successfully!

```
df.isna().sum()
```

```
RM         0
LSTAT      0
PTRATIO    0
MEDV       0
dtype: int64
```

```
target = "MEDV"
y = df[target]
x = df.drop(target, axis=1)
```

```
y.head()
```

```
0    504000.0
1    453600.0
2    728700.0
3    701400.0
4    760200.0
Name: MEDV, dtype: float64
```

```
x.head()
```

	RM	LSTAT	PTRATIO
0	6.575	4.98	15.3
1	6.421	9.14	17.8
2	7.185	4.03	17.8
3	6.998	2.94	18.7
4	7.147	5.33	18.7

```
#Spliting data for training and testing
#Here, 20% data used for testing and 80% data used for training
```

```
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)
```

```
from sklearn.linear_model import LinearRegression
regression = LinearRegression()
regression.fit(x_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
train_score = round(regression.score(x_train, y_train)*100,2)
print('Train score of linear regression',train_score)
y_pred = regression.predict(x_test)
```

```
Train score of linear regression 70.67
```

```
from sklearn.metrics import r2_score
score=round(r2_score(y_test,y_pred)*100,2)
print('r_2 score',score)
```

```
r_2 score 76.05
```

```
round(regression.score(x_test,y_test)*100,2)
```

```
76.05
```

```
from sklearn import metrics
print("Mean absolute error on test data of linear regression",metrics.mean_absolute_error(y_test,y_pred))
print("Mean squared error on test data of linear regression",metrics.mean_squared_error(y_test,y_pred))
print("Root mean squared error on test data of linear regression",np.sqrt(metrics.mean_squared_error(y_test,y_pred)))
```

```
Mean absolute error on test data of linear regression 62702.27365826064
                                 of linear regression 6244287741.571801
                                 data of linear regression 79020.80575121846
```

Saved successfully! ×

```
df1=pd.DataFrame({'Actual':y_test,'Predicted':y_pred,'Variance':y_test-y_pred})
df1.head()
```

	Actual	Predicted	Variance
285	569100.0	586182.279279	-17082.279279
202	420000.0	260336.446319	159663.553681
176	835800.0	658274.838376	177525.161624
354	459900.0	297776.731147	162123.268853
8	346500.0	272172.683356	74327.316644

Saved successfully!