

```
In [1]: 1 ASSIGNMENT NO:4
        2
        3 AIM:
        4 1. Linear Regression : Univariate and Multivariate
        5 2. Least Square Method for Linear Regression
        6 3. Measuring Performance of Linear Regression
        7 4. Example of Linear Regression
        8 5. Training data set and Testing data set
```

```
In [ ]: 1 import numpy as np
        2 import pandas as pd
        3 import matplotlib.pyplot as plt
```

```
In [5]: 1 from sklearn.datasets import fetch_california_housing
        2
        3 california = fetch_california_housing()
        4 x = california.data
        5 y = california.target
```

In [6]: 1 california

```
Out[6]: {'data': array([[ 8.3252, 41., 6.98412698, ..., 2.55555556,
        37.88, -122.23],
       [ 8.3014, 21., 6.23813708, ..., 2.10984183,
        37.86, -122.22],
       [ 7.2574, 52., 8.28813559, ..., 2.80225989,
        37.85, -122.24],
       ...,
       [ 1.7, 17., 5.20554273, ..., 2.3256351,
        39.43, -121.22],
       [ 1.8672, 18., 5.32951289, ..., 2.12320917,
        39.43, -121.32],
       [ 2.3886, 16., 5.25471698, ..., 2.61698113,
        39.37, -121.24]]),
'target': array([4.526, 3.585, 3.521, ..., 0.923, 0.847, 0.894]),
'frame': None,
'target_names': ['MedHouseVal'],
'feature_names': ['MedInc',
'HouseAge',
'AveRooms',
'AveBedrms',
'Population',
'AveOccup',
'Latitude',
'Longitude'],
'DESCR': '.. _california_housing_dataset:\n\nCalifornia Housing dataset\n-----
\n\n**Data Set Characteristics:**\n\n :Number of Instances: 20640\n\n
:Number of Attributes: 8 numeric, predictive attributes and the target\n\n
:Attribute Information:\n
- MedInc median income in block group\n
- HouseAge median house age in block group\n
- AveRooms average number of rooms per household\n
- AveBedrms average number of bedrooms per household\n
- Population block group population\n
- AveOccup average number of household members\n
- Latitude block group latitude\n
- Longitude block group longitude\n\n
:Missing Attribute Values: None\n\nThis dataset was obtained from the StatLib repository.\nhttps://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html\n\nThe target variable is the median house value for California districts,\nexpressed in hundreds of thousands of dollars ($100,000).\n\nThis dataset was derived from the 1990 U.S. census, using one row per census\nblock group. A block group is the smallest geographical unit for which the U.S.\nCensus Bureau publishes sample data (a block group typically has a population\nof 600 to 3,000 people).\n\nA household is a group of people residing within a home. Since the average\nnumber of rooms and bedrooms in this dataset are provided per household, these\ncolumns may take surprisingly large values for block groups with few households\nand many empty houses, such as vacation resorts.\n\nIt can be downloaded/loaded using the\nfunc:`sklearn.datasets.fetch_california_housing` function.\n\n.. topic:: References\n\n - Pace, R. Kelley and Ronald Barry, Sparse Spatial Autoregressions,\nStatistics and Probability Letters, 33 (1997) 291-297\n'}
```

In [8]: 1 data = pd.DataFrame(california.data)

In [10]: 1 data.columns = california.feature_names
2 data.head()

```
Out[10]:
```

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25

```
In [11]: 1 data['PRICE'] = california.target
```

```
In [12]: 1 data.isnull().sum()
```

```
Out[12]: MedInc      0
HouseAge    0
AveRooms    0
AveBedrms   0
Population  0
AveOccup    0
Latitude    0
Longitude   0
PRICE       0
dtype: int64
```

```
In [13]: 1 x = data.drop(['PRICE'], axis = 1)
2 y = data['PRICE']
```

```
In [15]: 1 from sklearn.model_selection import train_test_split
2 xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2, random_state=0)
```

```
In [16]: 1 import sklearn
2 from sklearn.linear_model import LinearRegression
3 lm = LinearRegression()
4 model=lm.fit(xtrain, ytrain)
```

```
In [17]: 1 ytrain_pred = lm.predict(xtrain)
2 ytest_pred = lm.predict(xtest)
```

```
In [18]: 1 df=pd.DataFrame(ytrain_pred,ytrain)
2 df=pd.DataFrame(ytest_pred,ytest)
3
```

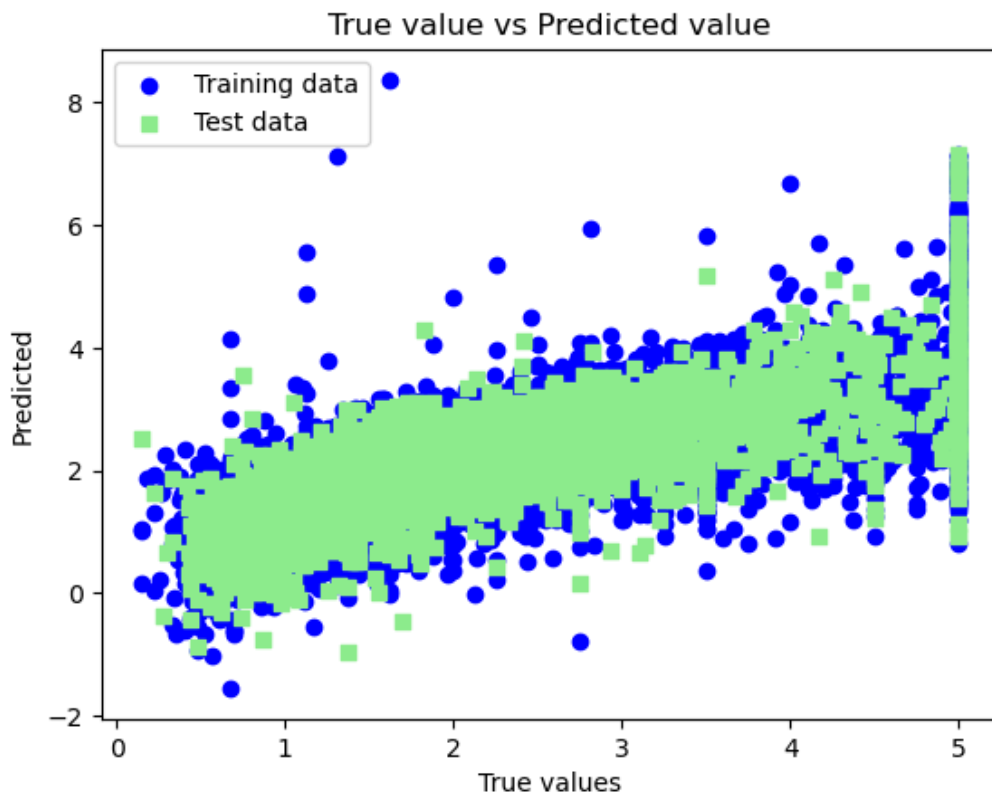
```
In [20]: 1 from sklearn.metrics import mean_squared_error, r2_score
2 mse = mean_squared_error(ytest, ytest_pred)
3 print(mse)
4 mse = mean_squared_error(ytrain_pred,ytrain)
5 print(mse)
```

```
0.5289841670367221
0.5234413607125449
```

```
In [21]: 1 mse = mean_squared_error(ytest, ytest_pred)
2 print(mse)
```

```
0.5289841670367221
```

```
In [23]: 1 plt.scatter(ytrain ,ytrain_pred,c='blue',marker='o',label='Training data')
2 plt.scatter(ytest,ytest_pred ,c='lightgreen',marker='s',label='Test data')
3 plt.xlabel('True values')
4 plt.ylabel('Predicted')
5 plt.title("True value vs Predicted value")
6 plt.legend(loc= 'upper left')
7 plt.plot()
8 plt.show()
```



```
In [ ]: 1 Name:Saurav Ratilal Raysing
2 Rollno:13267
3 B3
```