ASSIGNMENT NO : 4

```
[1]: Practical 4
     Aim:
     1. Linear Regression : Univariate and Multivariate
     2. Least Square Method for Linear Regression
     3. Measuring Performance of Linear Regression
     4. Example of Linear Regression
     5. Training dataset and Testing dataset
```

```
[ ]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
```

```
[2]: x=np.array([95,85,85,70,60])
     y=np.array([85,90,70,64,70])
     model= np.polyfit(x,y,1)
     model
```

```
[2]: array([ 0.53766234, 33.32467532])
```

```
[3]: predict=np.poly1d(model)
     predict(65)
```

```
[3]: 68.27272727272727
```

```
[4]: y_pred=predict(x)
     y_pred
```

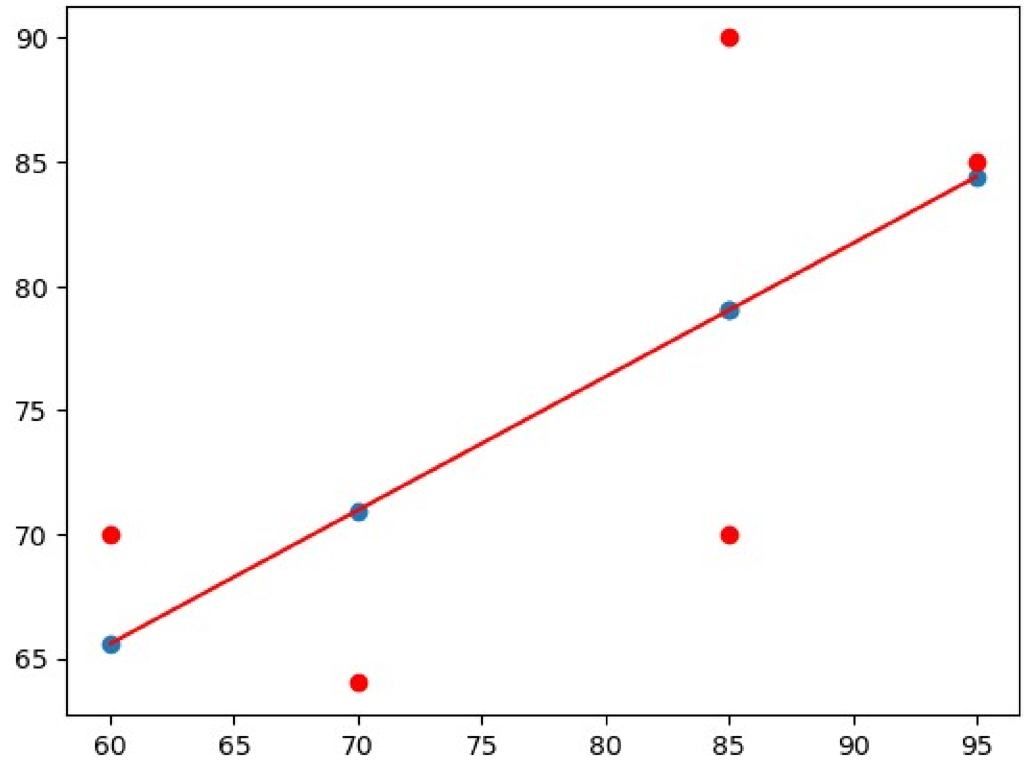
```
[4]: array([84.4025974 , 79.02597403, 79.02597403, 70.96103896,
65.58441558])
```

```
[5]: from sklearn.metrics import r2_score
     r2_score(y,y_pred)
```

```
[5]: 0.4516887333445776
```

```
[6]: y_line= model[1] + model[0]* x
     plt.plot(x, y_line,c='r')
     plt.scatter(x,y_pred)
```

```
[6]: <matplotlib.collections.PathCollection at 0x1cf91438450>
```



```
[26]: importssl
      from sklearn.datasetsimport fetch_california_housing
      ssl._create_default_https_context=ssl._create_unverified_context
      california = fetch_california_housing(download_if_missing=True)
      x = california.data
      y = california.target
```

```
[27]: df=pd.DataFrame(california.data,columns=california.feature_names)
      df
```

```
[27]:    MedInc HouseAge AveRooms AveBedrms Population AveOccup Latitude \
     0     8.3252     41.0 6.984127    1.023810    322.0 2.555556   37.88
     1     8.3014     21.0 6.238137    0.971880    2401.0 2.109842  37.86 2
      7.2574     52.0 8.288136    1.073446    496.0 2.802260   37.85 3
      5.6431     52.0 5.817352    1.073059    558.0 2.547945   37.85
      4     3.8462     52.0 6.281853  1.081081     565.0 2.181467    37.85
     …       …       …       …       …         …        …        …
     20635 1.5603     25.0 5.045455  1.133333     845.0 2.560606    39.48
      20636 2.5568     18.0 6.114035  1.315789     356.0 3.122807    39.49
      20637 1.7000     17.0 5.205543  1.120092    1007.0 2.325635    39.43
      20638 1.8672     18.0 5.329513  1.171920     741.0 2.123209    39.43
```

```
20639 2.3886     16.0 5.254717  1.162264     1387.0 2.616981   39.37

       Longitude
0        -122.23
1        -122.22
2        -122.24
3        -122.25
4        -122.25
…          …
20635    -121.09
20636    -121.21
20637    -121.22
20638    -121.32
20639    -121.24

[20640 rows x 8 columns]
```

```
[11]: data['PRICE'] = california.target
      data.isnull().sum()
```

```
[11]: 0        0
      1        0
      2        0
      3        0
      4        0
      5        0
      6        0
      7        0
      PRICE    0
      dtype: int64
```

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

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

```
[39]: import sklearn
```

```
[40]: from sklearn.linear_model import LinearRegression
```

```
[41]: lm = LinearRegression()
```

```
[42]: model= lm.fit(xtrain,ytrain)
```

```
[43]: ytrain_pred= lm.predict(xtrain)
```

```
[44]: ytest_pred= lm.predict(xtest)
```

```
[46]: df =pd.DataFrame(ytrain_pred,ytrain)
```

```
[48]: df =pd.DataFrame(ytest_pred,ytest)
```

```
[51]: from sklearnmetricsimportmean_squared_error , r2_score
      mse = mean_squared_error(ytest,ytest_pred)
      print(mse)
      mse = mean_squared_error(ytrain_pred,ytrain)
      print(mse)
```

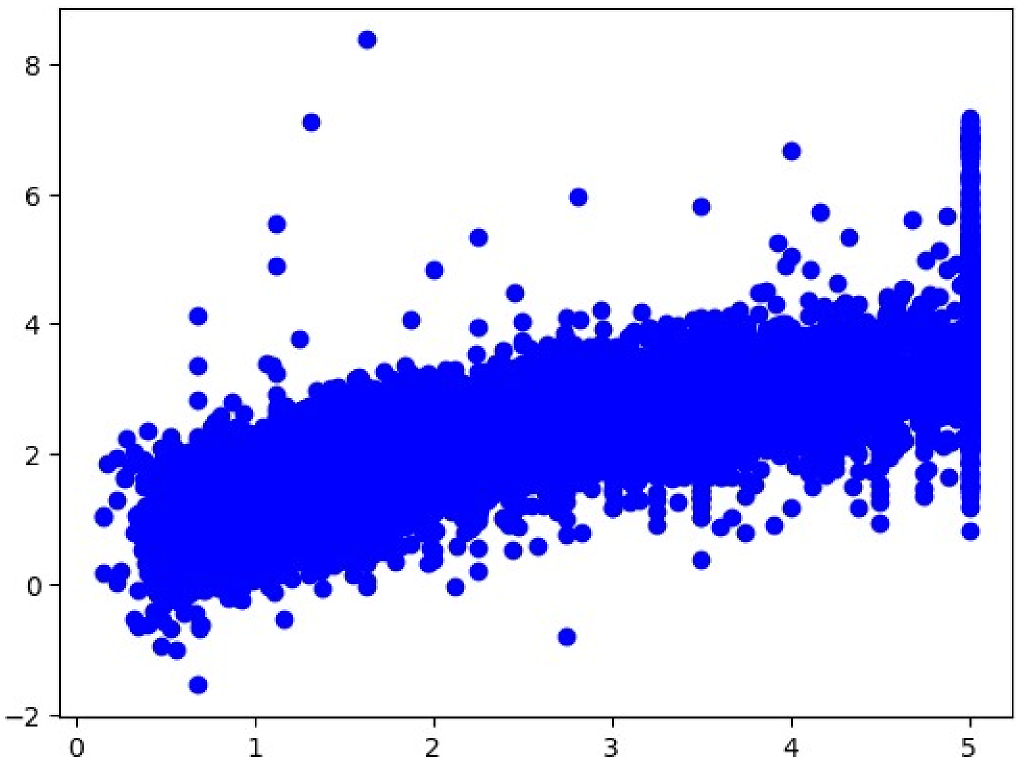
```
0.5289841670367192
 0.5234413607125448
```

```
[52]: mse = mean_squared_error(ytest, ytest_pred)
      print(mse)
```

```
0.5289841670367192
```

```
[57]:  plt.scatter(ytrain,ytrain_pred,c='blue',marker='o',label='Training
data')
```

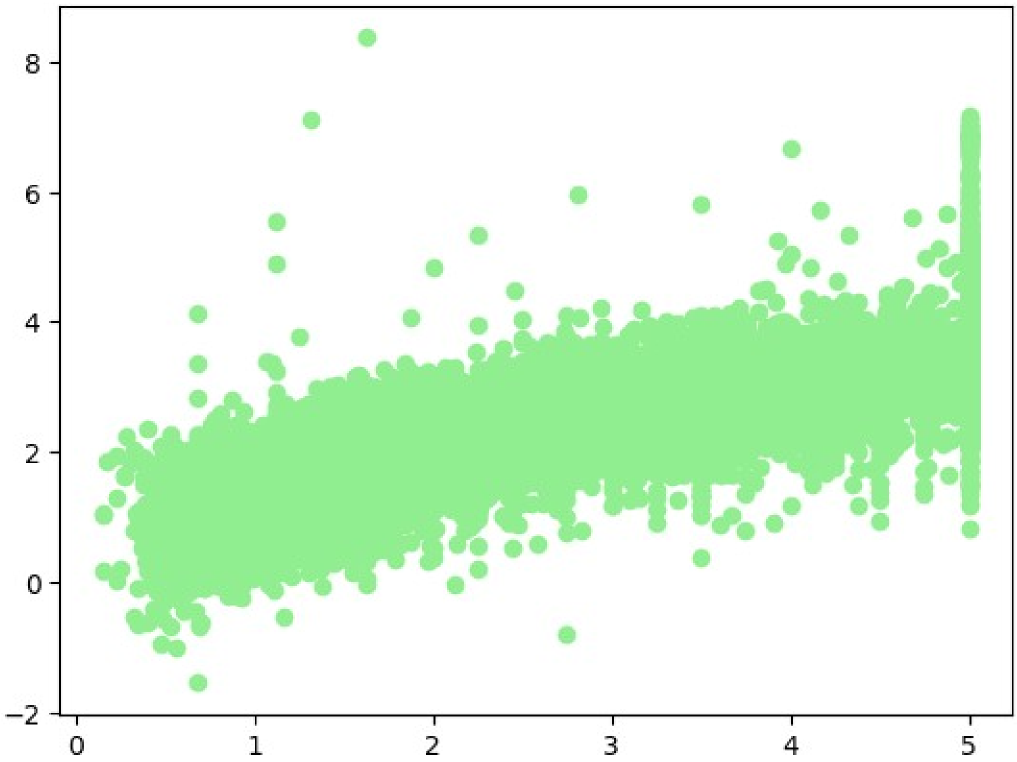
```
[57]: <matplotlib.collections.PathCollection at 0x1cf930e8a10>
```



[59]:

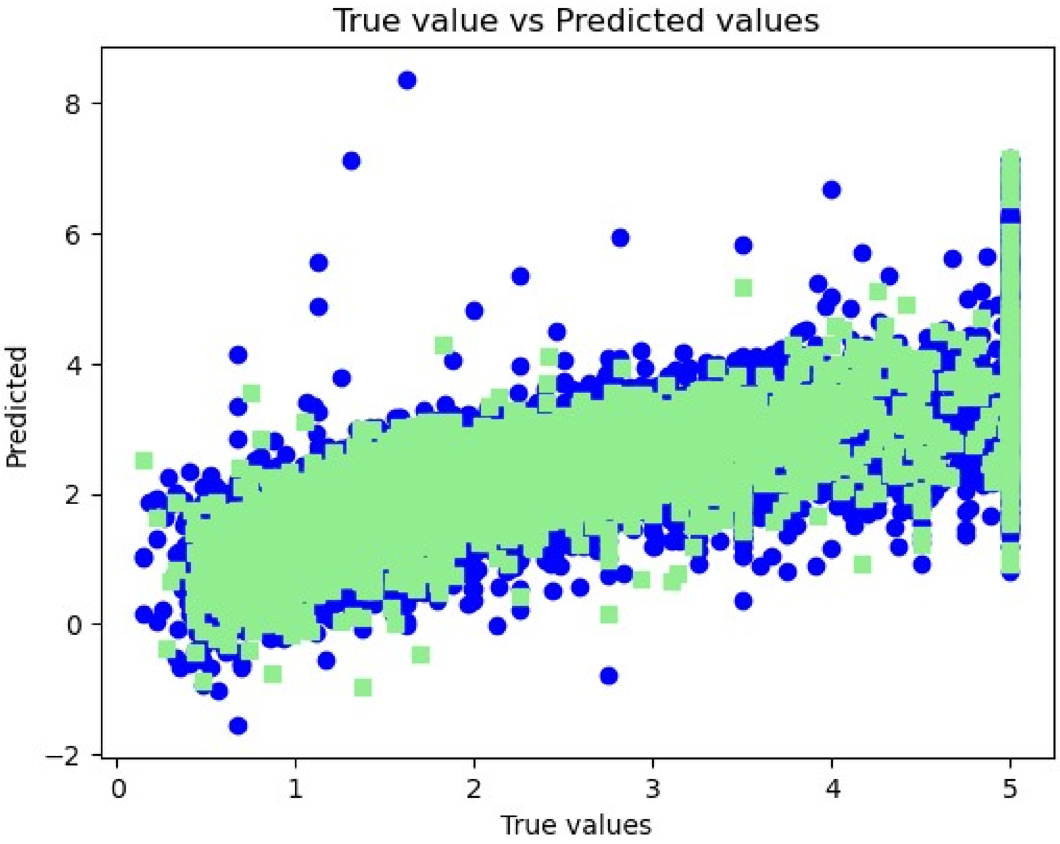
```
plt.scatter(ytrain,ytrain_pred,c='lightgreen',marker='o',label='Test
data')
```

```
[59]: <matplotlib.collections.PathCollection at 0x1cf9731e310>
```



```
[56]: plt.scatter(ytrain,ytrain_pred,c='blue',marker='o',
      label='Training data')
      plt.scatte
      r(ytest,ytest_pred,c
      ='lightgreen',marker='s' ,label='Test
      data') plt.xlabel('True values') plt.ylabel('Predicted')
      plt.title("True value vs Predicted values") plt.plot()
```

```
[56]: []
```



```
[ ]: Name: Ashwini Bharat Mohite
     Roll_no:13232
```

```
[ ]:
```