UniVariate Data

jupyter Assignment2 Last Checkpoint: 7 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

Prediction Function

In [24]:
```
def predict(predictiondata):
    predictiondata = vectorization(predictiondata)
    predictions = matrixMulti(a,predictiondata, predictiondata.shape[1] - 1)
    return predictions
y_predict=predict(X_test)
print(y_test)
y_predict
```

```
[17.929    0.55657  1.9869   3.6518   0.20421  7.5435  17.054    5.3854
  3.2522  17.592    5.9966   0.29678  1.844   13.501    6.7318   6.8233
  4.263    3.1551   2.8214   3.3411   1.0179   6.6799   1.0173   3.5129
  5.4974   1.8495   3.8166   0.47953 15.505    4.8852 ]
```

Out[24]:
```
array([21.94410875,  3.01826045,  2.89723213,  4.23772511,  2.75772165,
        8.21925875, 19.17774715, 10.53167473,  3.38051073,  3.85675419,
        4.42827019,  4.3787857 ,  3.48866116, 11.86715964,  5.90123851,
        3.55829716,  3.1174679 ,  3.40829359,  2.60390241,  5.18723104,
        3.17887636,  8.65722626,  3.12855719,  2.98546952,  4.14686444,
        3.59645387,  2.5981789 ,  3.40257008, 13.46020236,  5.17804958])
```

Calculating Accuracy

In [25]:
```
def EvaluatePerformance(prediction,actual):
#Evaluating preformance by using R^2 method
    sst = np.sum((actual-actual.mean())**2)
    ssr = np.sum((prediction-actual)**2)
    r2 = 1-(ssr/sst)
    return(r2)

EvaluatePerformance(y_predict,y_test)
```

Out[25]:
```
0.608541750529632
```

UniVariate Data

```
    return(r2)

EvaluatePerformance(y_predict,y_test)
```

Out[25]: 0.608541750529632

Comparing between this method and the original model from sklearn:

In [26]:
```
from sklearn.linear_model import LinearRegression
clf = LinearRegression()
reg=clf.fit(X_train, y_train)

pred=reg.predict(X_test)
print('Implementation of Code',y_predict)
print('Library Model Results',pred)
EvaluatePerformance(pred,y_test)
```

```
Implementation of Code [21.94410875  3.01826045  2.89723213  4.23772511  2.75772165  8.21925875
 19.17774715 10.53167473  3.38051073  3.85675419  4.42827019  4.3787857
  3.48866116 11.86715964  5.90123851  3.55829716  3.1174679   3.40829359
  2.60390241  5.18723104  3.17887636  8.65722626  3.12855719  2.98546952
  4.14686444  3.59645387  2.5981789   3.40257008 13.46020236  5.17804958]
Library Model Results [21.94510456  3.01839742  2.8973636   4.23791742  2.75784679  8.21963173
 19.17861742 10.53215265  3.38066414  3.85692921  4.42847114  4.37898441
  3.48881947 11.86769816  5.9015063   3.55845863  3.11760937  3.40844825
  2.60402057  5.18746643  3.17902061  8.65761912  3.12869917  2.985605
  4.14705262  3.59661708  2.5982968   3.40272449 13.46081317  5.17828455]
```

Out[26]: 0.6085249061106279

In [ ]:
```

```

MultiVariate Data

```
y_predict=predict(X_test)
print(y_test)
y_predict
```

```
[469000. 287000. 464500. 299000. 573900. 229900. 329999. 599000. 539900.
 249900. 212000. 232000. 314900. 368500. 579900.]
```

Out[10]:
```
array([376976.75720562, 332696.40167842, 252218.84560074, 222080.69841529,
       536311.76790698, 226636.4648503 , 302682.71995777, 434208.22301235,
       430586.97276914, 210399.24601783, 243107.31273072, 249072.50439426,
       261096.74942281, 278042.50633996, 448350.4313541 ])
```

Calculating Accuracy

In [11]:
```
def EvaluatePerformance(prediction,actual):
#Evaluating preformance by using R^2 method
    sst = np.sum((actual-actual.mean())**2)
    ssr = np.sum((prediction-actual)**2)
    r2 = 1-(ssr/sst)
    return(r2)

EvaluatePerformance(y_predict,y_test)
```

Out[11]:
```
0.5173270013165043
```

MultiVariate Data

jupyter Assignment2 Last Checkpoint: 7 hours ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
    r2 = 1-(ssr/sst)
    return(r2)

EvaluatePerformance(y_predict,y_test)
```

Out[11]: 0.5173270013165043

Comparing between this method and the original model from sklearn:

In [12]:
```
from sklearn.linear_model import LinearRegression
clf = LinearRegression()
reg=clf.fit(X_train, y_train)

pred=reg.predict(X_test)
print('Implementation of Code',y_predict)
print('Library Model Results',pred)
EvaluatePerformance(pred,y_test)
```

```
Implementation of Code [376976.75720562 332696.40167842 252218.84560074 222080.69841529
 536311.76790698 226636.4648503  302682.71995777 434208.22301235
 430586.97276914 210399.24601783 243107.31273072 249072.50439426
 261096.74942281 278042.50633996 448350.4313541 ]
Library Model Results [377894.73136494 330773.98374507 251478.67244215 220939.96157878
 539347.45081315 225556.2783372  302613.25807384 433634.9904903
 429965.61050284 209103.25194181 242246.03892532 250542.37893594
 260474.57176624 279897.41883561 450217.0272468 ]
```

Out[12]: 0.5186915143443924