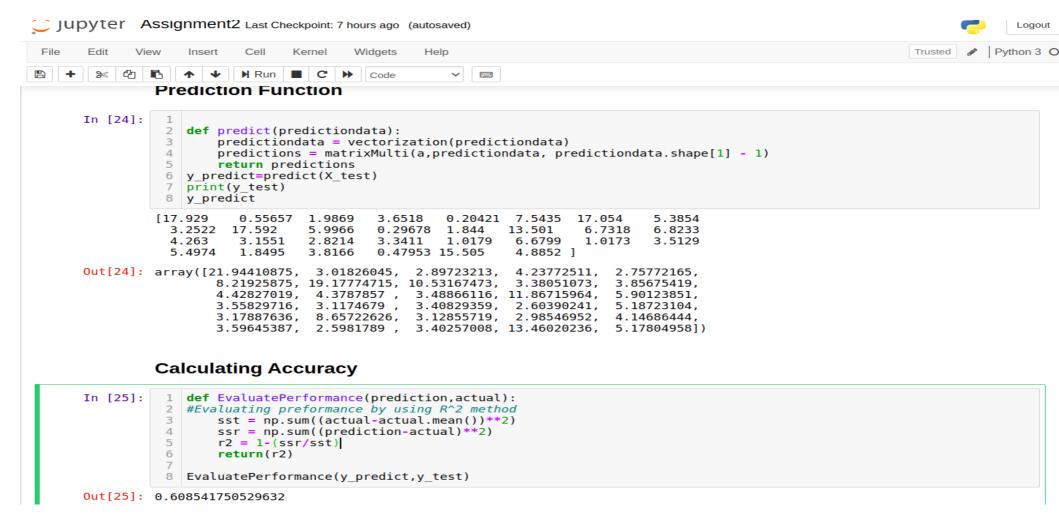
# **UniVariate Data**



# **UniVariate Data**

return(r2)

```
8 EvaluatePerformance(y predict,y test)
Out[25]: 0.608541750529632
        Comparing between this method and the original model from sklearn:
In [26]: 1 from sklearn.linear model import LinearRegression
          2 clf = LinearRegression()
          3 reg=clf.fit(X train, y train)
          5 pred=reg.predict(X test)
          6 print('Implementation of Code', y predict)
          7 print('Library Model Results', pred)
          8 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 [ ]: 1
```

# **MultiVariate Data**

#### **Calculating Accuracy**

# **MultiVariate Data**

