In [4]:

```
#Description: This program predicts the price of Bitcoin for the next 30 days
```

In [5]:

```
#Import the libraries
import pandas as pd
import numpy as np
```

In [6]:

```
#Store the data into a variable
df = pd.read_csv('data/BTC-USD.csv')
```

In [7]:

```
#Show the first 7 rows of data
df.head(7)
```

Out[7]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	14/6/2015	232.442001	234.858002	232.003998	233.542999	233.542999	12165900
1	15/6/2015	233.421997	237.835999	233.421997	236.822998	236.822998	19912100
2	16/6/2015	236.764999	251.742004	236.121994	250.895004	250.895004	41612000
3	17/6/2015	250.822998	256.852997	246.475998	249.283997	249.283997	43858400
4	18/6/2015	249.427994	252.108002	244.126999	249.007004	249.007004	30980200
5	19/6/2015	249.042999	250.977005	243.787003	244.606003	244.606003	23965300
6	20/6/2015	244.529999	245.828003	240.626999	245.212006	245.212006	20608100

In [8]:

```
#Remove the date column
df.drop(['Date'],1,inplace=True)
df.drop(['Open'],1,inplace=True)
df.drop(['High'],1,inplace=True)
df.drop(['Low'],1,inplace=True)
df.drop(['Close'],1,inplace=True)
df.drop(['Volume'],1,inplace=True)
```

In [9]:

```
#show the first 7 rows of the new data set
df.head(7)
```

Out[9]:

	Adj Close
0	233.542999
1	236.822998
2	250.895004
3	249.283997
4	249.007004
5	244.606003
6	245.212006

In [10]:

In [10]:

```
#A variable for predicting 'n' days out into the future
prediction_days = 30 #n = prediction_days = 30
```

In [11]:

```
#Create another column shifted 'n' units up
df['Prediction'] = df[['Adj Close']].shift(-prediction_days)
```

In [12]:

```
#Show the first 7 rows of the new data set
df.head(7)
```

Out[12]:

	Adj Close	Prediction
0	233.542999	287.463989
1	236.822998	285.829010
2	250.895004	278.088989
3	249.283997	279.471985
4	249.007004	274.901001
5	244.606003	273.614014
6	245.212006	278.980988

In [13]:

```
#Show the last 7 rows of the new data set
df.tail(7)
```

Out[13]:

	Adj Close	Prediction
1818	9665.533203	NaN
1819	9653.679688	NaN
1820	9758.852539	NaN
1821	9771.489258	NaN
1822	9795.700195	NaN
1823	9870.094727	NaN
1824	9426.325195	NaN

In [14]:

```
#Create the independent data set

#Convert the dataframe to a numpy array and drop the prediction column
x = np.array(df.drop(['Prediction'], 1))
```

In [15]:

```
#Remove the last 'n' rows where 'n' is the prediction_days
x = x[:len(df)-prediction_days]
print(x)
```

```
[[ 233.542999]
 [ 236.822998]
 [ 250.895004]
 ...
 [8756.430664]
 [8601.795898]
 [8804.477539]]
```

In [16]:

```
#Create the dependent data set

#Convert teh dataframe to a numpy array
y = np.array(df['Prediction'])
```

In [17]:

```
#Get all of the values except the last 'n' rows
y = y[:-prediction_days]
print(y)
```

```
[ 287.463989  285.82901   278.088989 ... 9795.700195 9870.094727
 9426.325195]
```

In [18]:

```
#Split the data into 80% training and 20% testing
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)
```

In [19]:

```
#Set the prediction_days_array equal to the last 30 rows from the original data set
prediction_days_array = np.array(df.drop(['Prediction'],1))[-prediction_days:]
print(prediction_days_array)
```

```
[[ 9269.987305]
 [ 9733.72168 ]
 [ 9328.197266]
 [ 9377.013672]
 [ 9670.739258]
 [ 9726.575195]
 [ 9729.038086]
 [ 9522.981445]
 [ 9081.761719]
 [ 9182.577148]
 [ 9209.287109]
 [ 8790.368164]
 [ 8906.93457 ]
 [ 8835.052734]
 [ 9181.017578]
 [ 9525.750977]
 [ 9439.124023]
 [ 9700.414063]
 [ 9461.058594]
 [10167.26856 ]
 [ 9529.803711]
 [ 9656.717773]
 [ 9800.636719]
 [ 9665.533203]
 [ 9653.679688]
 [ 9758.852539]
 [ 9771.489258]
 [ 9795.700195]
 [ 9870.094727]
 [ 9426.325195]]
```

In [20]:

```
from sklearn.svm import SVR
```

In [21]:

```
#Create and train the Support Vector Machine (Regression) using radio basis function
svr_rbf = SVR(kernel = 'rbf', C=1e3, gamma=0.00001)
svr_rbf.fit(x_train, y_train)
```

Out[21]:

```
SVR(C=1000.0, cache_size=200, coef0=0.0, degree=3, epsilon=0.1, gamma=1e-05,
    kernel='rbf', max_iter=-1, shrinking=True, tol=0.001, verbose=False)
```

In [22]:

```
#Test the model
svr_rbf_confidence = svr_rbf.score(x_test, y_test)
print("svr_rbf accuracy: ", svr_rbf_confidence)
```

```
svr_rbf accuracy:  0.8223776853564652
```

In [23]:

```
#Print the predicted values
svm_prediction = svr_rbf.predict(x_test)
print(svm_prediction)

print()

#Print the actual values
print(y_test)
```

```
[10002.92967984   597.95505661  8547.59925133  8732.80230515
  3931.86298915  2056.65976813   872.96526796  5334.87863056
  8310.68201258   620.75199235  2728.63474106  6973.50449349
   318.43673826  8310.48425737  3993.62100683  7539.72252822
  4719.99143127  7170.80357     9402.38768998  4433.05744873
  2800.19557423  6252.2774308   7374.36143434  6477.70479899
  6490.74643015  8402.92620244  6548.65751665   691.08859007
  7556.22282019   294.98314262   609.48918193  2560.35659079
  5020.03217448   495.38703358   451.48303435  7494.29219658
  3907.1273553   8025.01045591  1246.19531864  2731.45106597
   353.06554702   701.30196631  1877.07789156   601.03060848
  2735.08531542  8669.25297994  8788.90678721  8258.99394907
  4158.51332289  7007.53684072  7687.84687823  6893.70189388
   841.53611651  3919.57841835  7712.33389093  1061.21319813
  9109.0939603   6980.08894907  6525.66622689  6414.51159662
   266.21014623  7350.20042737  7966.21347377  7897.18401938
  8292.10270071   452.77824633  2692.71284656  8082.7984722
  1509.14465268  8990.01667042  6646.46974302  1575.09220131
  8259.72116535  2729.23907348  8355.21164454  4350.00244842
  7896.48802473  2730.04481215  6255.35879725  8325.97338841
  9495.91511631  1151.13099804   768.41286676  6314.39733193
  4015.40953871  6991.72623704  4279.34009309  7969.66384816
  7449.94429017  8307.74416518  9059.52929699  7988.9139657
 10046.93931424  1123.17315851  8265.23843311   653.24340913
   383.50404462  4437.36115658  5022.67870141  2859.48050496
  8266.100606    2777.39088097  2617.57897859  3922.58551958
   454.6657999   7714.00027947  6188.75371675  1283.03947366
   636.62085109  9202.41819358  8116.28610015   400.91298183
  7363.54913091  8667.36646371   451.8265527   8194.36019069
  7552.63770314  3874.40030463  8552.3975287   8598.45473453
   645.25008171   642.01833517  6531.70853611  8069.83816038
   717.87848297  7472.14240859  8286.08919981   434.42691067
   468.94881363  8197.04369396   757.23983907  7975.53555172
  2805.48383717  6743.60293259  9163.68600523  6708.4440618
  8911.98822362  6240.71307482   610.95896529  4363.38244247
   869.79073328  4227.8426481    291.16546694  9178.01836689
   641.03742207  8291.02034137   480.2968499    263.69179821
 10020.97044827   479.65524215   299.43416735  1074.95419088
  8297.36055639  7970.48721979  3916.29956413  5658.99718898
  7913.53558447  7698.34530133   494.71726089  1233.29742056
   444.5212981    270.83287682  7025.44724037   380.23722789
   270.38080061  1146.61074982  1166.27513695  6193.75474541
   711.65726902  8330.26818469   304.51690762  1132.99650561
   758.27530031  4146.64636313   444.34707141  4656.62338571
  9142.64192658  4224.82659951  8184.23483995  1480.99691362
  7732.16893748  9177.38002531   624.49964329  7976.1497848
  2565.59107429  3903.44292117   278.79867741  8249.56370836
   295.57490037  8288.39853847  7561.65788381  8536.14171935
  2828.95566813   262.37969933  6220.82054864   451.77211729
  1490.67091444  8628.28590197  6814.54299736  8148.00192793
  2805.68189441  3899.39206325  9736.74844407  7956.25343376
  2842.34208089  5074.27145099   866.98777534 10261.92485068
  1329.54303095   866.39778609  5103.59499596   474.63083891
  2760.89944656   695.79589198   633.71083135  6834.75423828
   443.15768441  8795.65431144  3879.5685091   8785.65633521
   605.87371749   482.16504894  8529.88252091 10253.8720121
   467.31287592 10301.28203479   971.97098434  1349.24287622
  6606.26703521  6275.55509099  8342.34220905  8961.39148958
  7528.93523049  7746.12588797  8460.11432559  9357.46174066
  3297.8723609    443.03725117  9068.0902556    670.83139945
   264.90557698  8681.33059214  7324.87478162  9143.93645295
  7350.44536271  8781.99987772  6209.38238034   437.37507436
  1744.3843434   8354.3938699   8555.74283733  8408.88765675
   642.59252229   321.66441988  1146.31762338   272.7800155
  1159.85891227   280.20696791  1468.14168016  2758.22029382
  7536.27083364  9093.34741428  7312.06733584  6366.73770767
   269.49149428  3346.7917247   2864.27939337  8275.07900126
  1147.97221002   286.05999962  8349.58420661  7234.32810586
   703.45755831  9116.78537322  7929.6587128   3899.95407719
  7537.12576052  7105.25936317  1084.75492113  9207.17289516
   466.07012248   263.350396     498.70428565  8639.3224873
  8812.99051822  4899.16249391  2725.97190213  4456.51042051
   574.18730166  5085.07319762  4502.31104181  8558.41227927
  7549.0549702  10186.71782215  7695.41753525   261.76645723
  8408.79266902   647.20280316  7944.00293761  8289.65983637
  6209.66650979  3924.17173202  6274.25449368   611.30524315
  7111.55950958  1507.05566473   292.85798221  7568.40440337
   753.42317803  8247.87590151  1636.82375354  1060.91885463
   492.7020223   7328.82683777  6453.46022841  8242.88404821
   264.46873004  8645.46453944  8026.42532534  8488.01299368
  1439.68523974   638.19085293   263.7658597   8879.47168488
   453.29535787  3971.85394228  4351.70521911   264.49081437
  7795.07836061  6199.19155012   701.44731794   265.84256716
  8588.56230585  8953.58637217  8910.86972809   292.71028521
   272.05856033  8513.95334358  8231.49241825  4144.7792772
   272.87363635  8508.52233705   278.80731723  2885.44053696
  7668.36672675  9640.24340481  8164.51239665   280.81921192
  5109.86357463   269.09738264  1375.67756933  7251.65272315
  8812.65486358  8265.72865741   924.19373285  3876.54320387
   691.37034052  9036.89413104  9136.66468568]

[15201.         575.536987  9795.700195 10231.74414   3999.820557
  2255.610107   902.828003  6618.140137  9934.433594   622.861023
  3213.939941  7708.990234   463.615997 10347.71289   4382.879883
  9293.521484  5289.770996  6517.310059  9174.910156  5059.817383
  2659.629883  6489.189941  7987.371582  6398.540039  6467.069824
  9700.414063  8864.766602   751.346985  6218.299805   262.868988
   600.825989  4325.129883  4826.47998    672.783997   446.721985
  9358.589844  3960.911133  7472.589844  1421.599976  2671.780029
   422.278992   581.697021  2202.419922   608.04303   2718.26001
 10701.69141  11786.29981   8253.549805  6468.399902  6972.371582
  8672.455078  6322.689941   673.106018  4226.060059  7624.919922
  1054.420044  7902.089844  6480.379883  6332.629883  6321.200195
   328.205994  6487.160156  8251.845703  9842.666016  6191.192871
   410.444      2552.449951  5238.438477  1175.949951  7895.959961
  6334.27002   2320.419922  8071.259766  2318.879883  7988.560547
  3882.590088  6529.589844  2564.060059  3486.950195  7988.155762
  6811.470215  1175.829956   778.088013  6238.049805  3470.450439
  7881.84668   4087.066162 10159.96094   7711.109863  9729.324219
  8807.010742 11474.90039   6853.839844   919.495972  9690.142578
   577.502991   414.321014  4565.299805  5251.937988  2805.620117
 10583.13477   2515.350098  4181.930176  3954.118164   455.670013
  6580.629883  3424.588135  1452.819946   610.203979 10326.05469
  9316.629883   400.184998  6162.47998   8912.654297   421.563995
 14606.5      11916.7002    5526.640137  7269.68457   8942.808594
   711.521973   700.971985  6351.799805  9395.009766   575.04303
  6773.879883  8813.582031   427.398987   368.766998  7879.071289
   654.468018  7271.208008  2895.889893  9328.197266  6884.640137
  9654.799805  6763.189941  4214.671875   608.312012  2608.560059
   911.198975  4014.182617   357.381012  7047.160156   703.41803
  8913.469727   737.226013   239.839996 11403.7002     763.781006
   232.975006  1061.349976  8804.880859  6985.470215  4048.72583
  4403.740234  9758.852539  9693.802734   526.232971  1347.890015
   449.010986   221.608994  3545.864746   388.78299    266.376007
  1274.98999   1255.150024  4451.870117   664.551025  8393.041992
   436.571991  1250.150024   780.086975  4106.660156   450.282013
  5204.958496  9235.354492  4236.310059 10360.54688   1039.969971
  6842.427734  8441.490234   650.619019  5982.45752   2407.879883
  3864.415039   242.968994  9757.970703   236.686996 10181.6416
  7531.663574 10801.67773   4073.26001     257.976013  6388.439941
   420.872986  1734.449951 19497.40039   7204.771484  9613.423828
  2744.909912  3678.924561  9607.423828 10131.05566   2686.810059
  3843.52002    1021.75     10441.27637   1555.449951   908.585022
  4781.990234   756.22699   2529.449951   567.23999    714.479004
  8205.167969   458.048004 17429.5       3857.717529  8510.379883
   610.892029   473.463989  9578.629883  9813.070313   369.949005
 11354.02441    895.026001  1537.670044 15455.40039   3419.937256
  5200.366211 10106.29981   6305.799805 17527.        9281.509766
 16477.59961   4892.009766   447.610992  9670.739258   702.031006
   228.121002  8838.375     7193.25      8693.833008  5903.439941
  9348.480469  6474.75       363.183014  2443.639893 10895.83008
  9439.124023 11208.55078    617.120972   415.479004  1221.380005
   311.084015  1187.810059   361.188995  1723.349976  2757.179932
  7569.629883 11392.37891   6083.689941  6543.200195   294.427002
  4703.390137  3650.620117  6906.919922  1249.609985   281.653992
 12952.2002    7790.149902   665.012024  9856.611328  7343.895508
  4599.879883  9180.962891  8038.77002    1222.5       6625.560059
   426.765015   235.018997   585.536987  9650.174805  9055.526367
  4023.968262  4371.600098  4087.476318   703.702026  3728.568359
  3583.96582   6793.624512  7146.133789 11523.5791    7707.770996
   281.881989  8367.847656   723.27301   5014.47998   9630.664063
  6184.709961  3947.094482  7567.149902   609.22699   6635.75
  1755.359985   254.320007  7424.29248    735.604004  8104.185547
  1182.680054   921.012024   665.122986  7292.995117  6385.620117
  8192.494141   230.056      9508.993164  9653.679688 10141.99609
   972.778992   688.700012   284.649994 17899.69922    458.536011
  4228.75      3601.013672   323.04599   6506.069824  4602.169922
   654.096985   332.906006  8000.32959  10343.10645   8247.179688
   227.085007   386.354004  6582.359863  9268.761719  4069.107178
   278.980988 11182.80664    246.063004  2732.159912 13819.79981
 10399.66895   6681.062988   403.416992  4130.810059   338.152008
  1045.77002   6475.740234  7621.299805  6834.759766  1004.450012
  3671.203613   739.247986 13831.79981   8845.740234]
```

In [24]:

```
#Print the model predictions for the next 'n=30' days
svm_prediction = svr_rbf.predict(prediction_days_array)
print(svm_prediction)

print()

#Print the actual price for Bitcoin for the last 30 days
print(df.tail(prediction_days))
```

```
[7582.23018614 7998.77010702 7721.28275456 7881.46753359 8123.03434859
 8011.35773458 8006.95050401 8260.11905623 7765.30482585 7547.37173834
 7533.7814144  8558.24075643 8377.77097254 8528.23967448 7548.81137732
 8262.53727188 8088.10094687 8061.77307833 8149.49164991 8313.52638374
 8265.66414465 8151.59004806 7919.82687416 8133.74079006 8157.63076097
 7960.10088663 7944.50562998 7922.84887999 7927.16267291 8048.43932047]

        Adj Close  Prediction
1795  9269.987305         NaN
1796  9733.721680         NaN
1797  9328.197266         NaN
1798  9377.013672         NaN
1799  9670.739258         NaN
1800  9726.575195         NaN
1801  9729.038086         NaN
1802  9522.981445         NaN
1803  9081.761719         NaN
1804  9182.577148         NaN
1805  9209.287109         NaN
1806  8790.368164         NaN
1807  8906.934570         NaN
1808  8835.052734         NaN
1809  9181.017578         NaN
1810  9525.750977         NaN
1811  9439.124023         NaN
1812  9700.414063         NaN
```

```
1813   9461.058594         NaN
1814  10167.268560         NaN
1815   9529.803711         NaN
1816   9656.717773         NaN
1817   9800.636719         NaN
1818   9665.533203         NaN
1819   9653.679688         NaN
1820   9758.852539         NaN
1821   9771.489258         NaN
1822   9795.700195         NaN
1823   9870.094727         NaN
1824   9426.325195         NaN
```