Bitcoin prediction and forecasting using machine learning

Prasad J Shimpi
07/01/2022

Introduction:

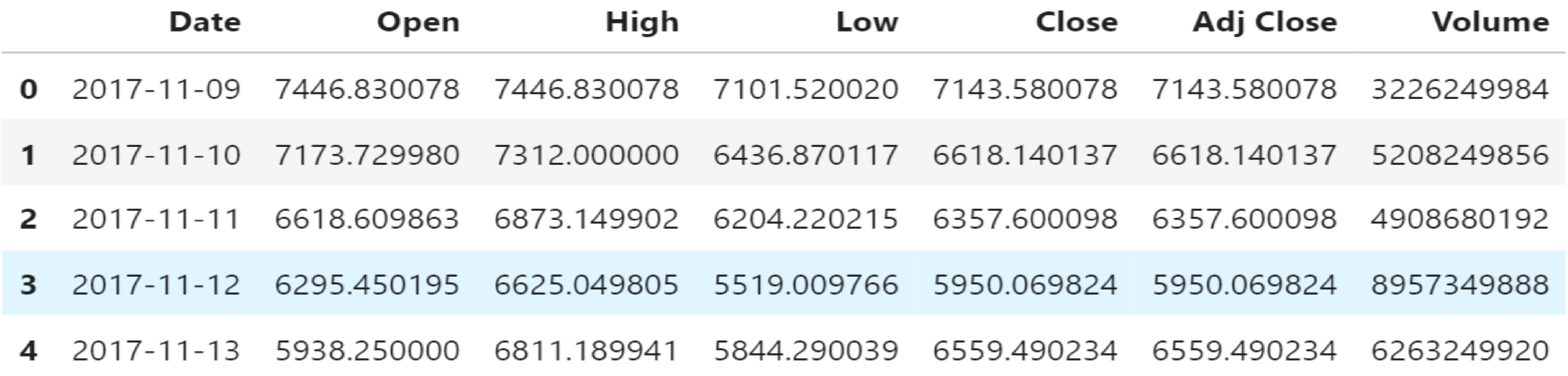
- Bitcoin nowadays is the talk of the town we can see bitcoin is moving in such amazing way ,that some of the investors have said that, Bitcoin is the new Gold.
- The way people used to think about gold as an investment, so now large number of people are showing interest in Bitcoin.
- But I saw still people think that since the bitcoin is giving high returns ; so with high returns comes high risk.
- The way bitcoin has increased it has been seen the most volatile cryptocurrency. So, some of the investors still don’t prefer to invest in it . or end up loosing money in it .
- So here in this presentation I have tried to predict the movement of the bitcoin by using machine learning and deep learning algorithms.

Data Collection:

1. Data sources:

- The source of the data is yahoo finance.

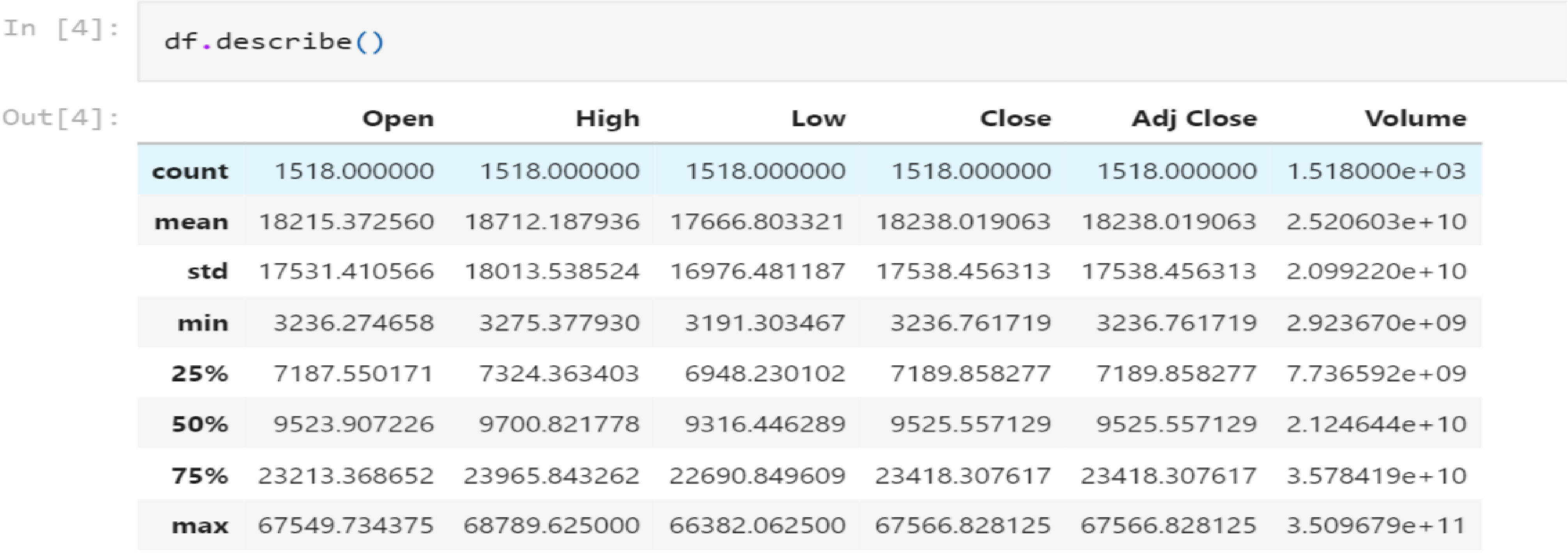
The data is consists of open , high , low , close , adjusted close and volume.



	Date	Open	High	Low	Close	Adj Close	Volume
0	2017-11-09	7446.830078	7446.830078	7101.520020	7143.580078	7143.580078	3226249984
1	2017-11-10	7173.729980	7312.000000	6436.870117	6618.140137	6618.140137	5208249856
2	2017-11-11	6618.609863	6873.149902	6204.220215	6357.600098	6357.600098	4908680192
3	2017-11-12	6295.450195	6625.049805	5519.009766	5950.069824	5950.069824	8957349888
4	2017-11-13	5938.250000	6811.189941	5844.290039	6559.490234	6559.490234	6263249920

Data Visualization:

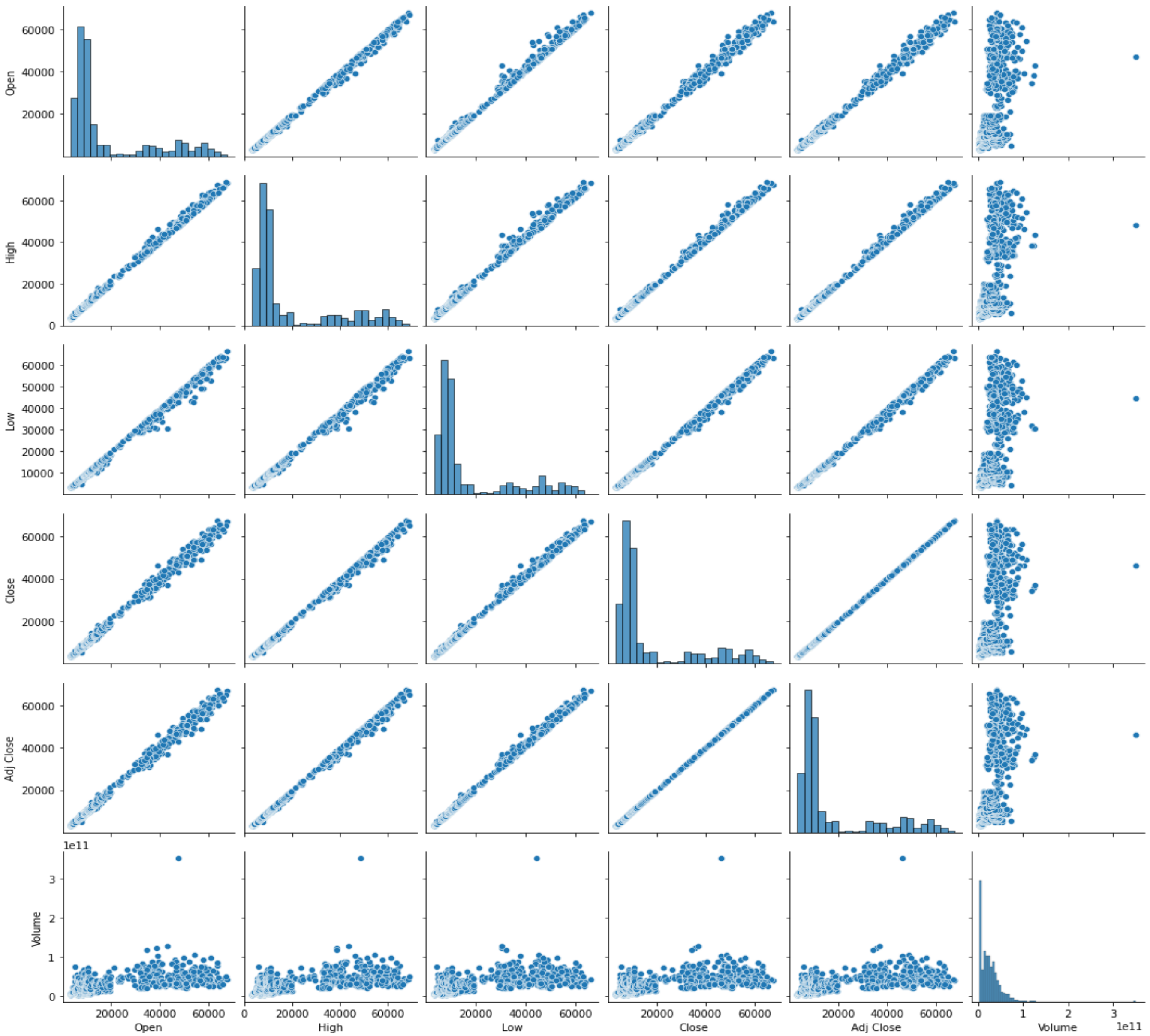
- Data visualization is so important to get the complete idea about the data .
- So I have plotted various graphs just to get the complete understanding of the correlation between various attributes.

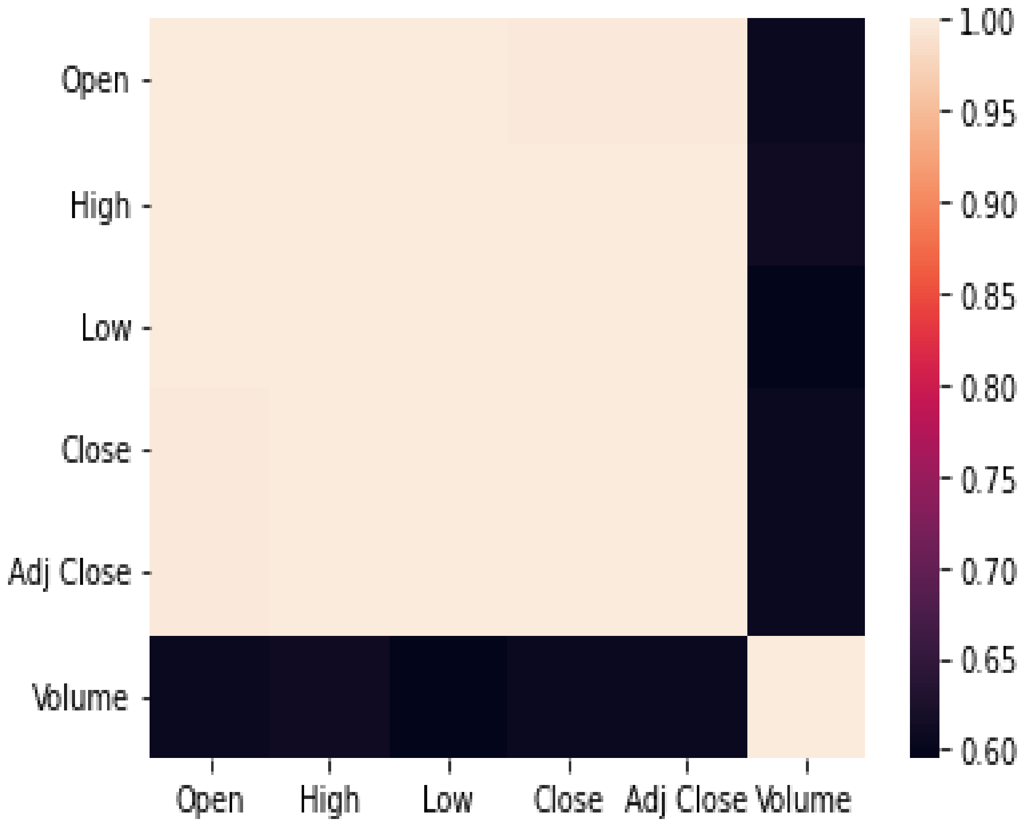


```
In [4]: df.describe()
```

Out[4]:

	Open	High	Low	Close	Adj Close	Volume
count	1518.000000	1518.000000	1518.000000	1518.000000	1518.000000	1.518000e+03
mean	18215.372560	18712.187936	17666.803321	18238.019063	18238.019063	2.520603e+10
std	17531.410566	18013.538524	16976.481187	17538.456313	17538.456313	2.099220e+10
min	3236.274658	3275.377930	3191.303467	3236.761719	3236.761719	2.923670e+09
25%	7187.550171	7324.363403	6948.230102	7189.858277	7189.858277	7.736592e+09
50%	9523.907226	9700.821778	9316.446289	9525.557129	9525.557129	2.124644e+10
75%	23213.368652	23965.843262	22690.849609	23418.307617	23418.307617	3.578419e+10
max	67549.734375	68789.625000	66382.062500	67566.828125	67566.828125	3.509679e+11

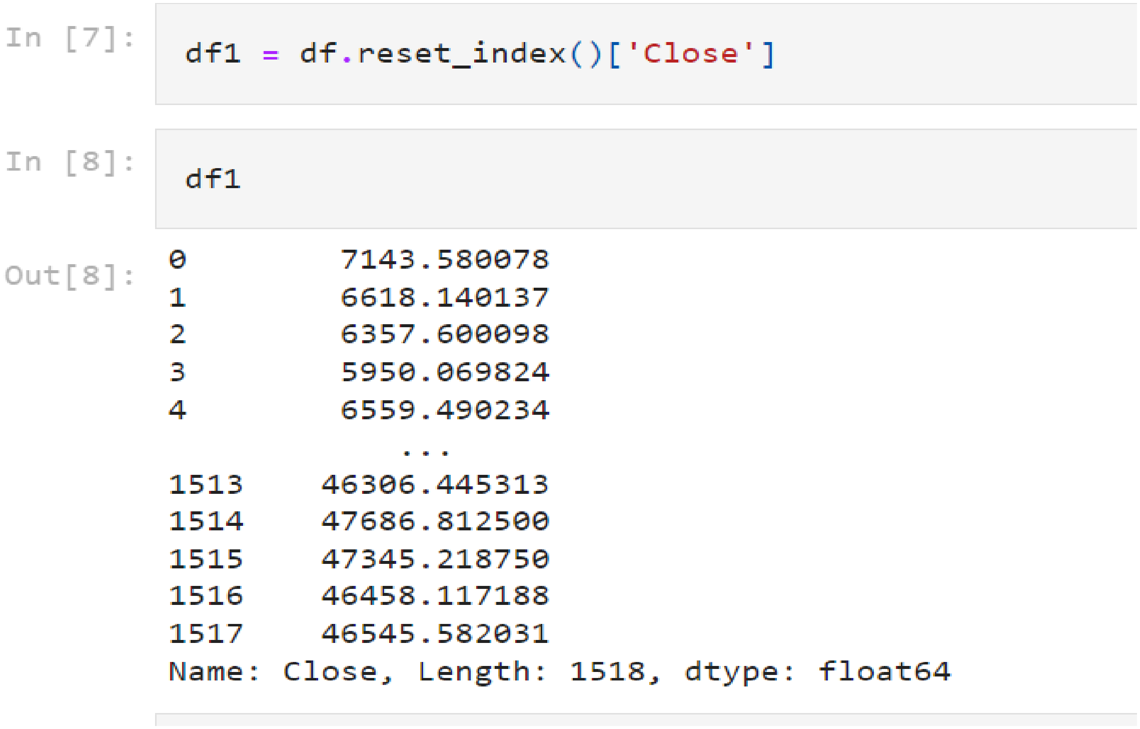

Open
High
Low
Close
Adj Close
Volume
60000
40000
20000
0
10000
20000
30000
40000
50000
60000
1e11
3
2
1
0
20000
40000
60000
Open
High
Low
Close
Adj Close
0
1
2
3
Volume
1e11


Open
High
Low
Close
Adj Close
Volume
Open
High
Low
Close
Adj Close
Volume
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60

Exploratory Data Analysis:

In our data set there are various attributes but we want to perform our operations mainly on the closing price of the bitcoin .

So for this purpose we have to make a new data frame with the desired value i.e. the column “Close”.

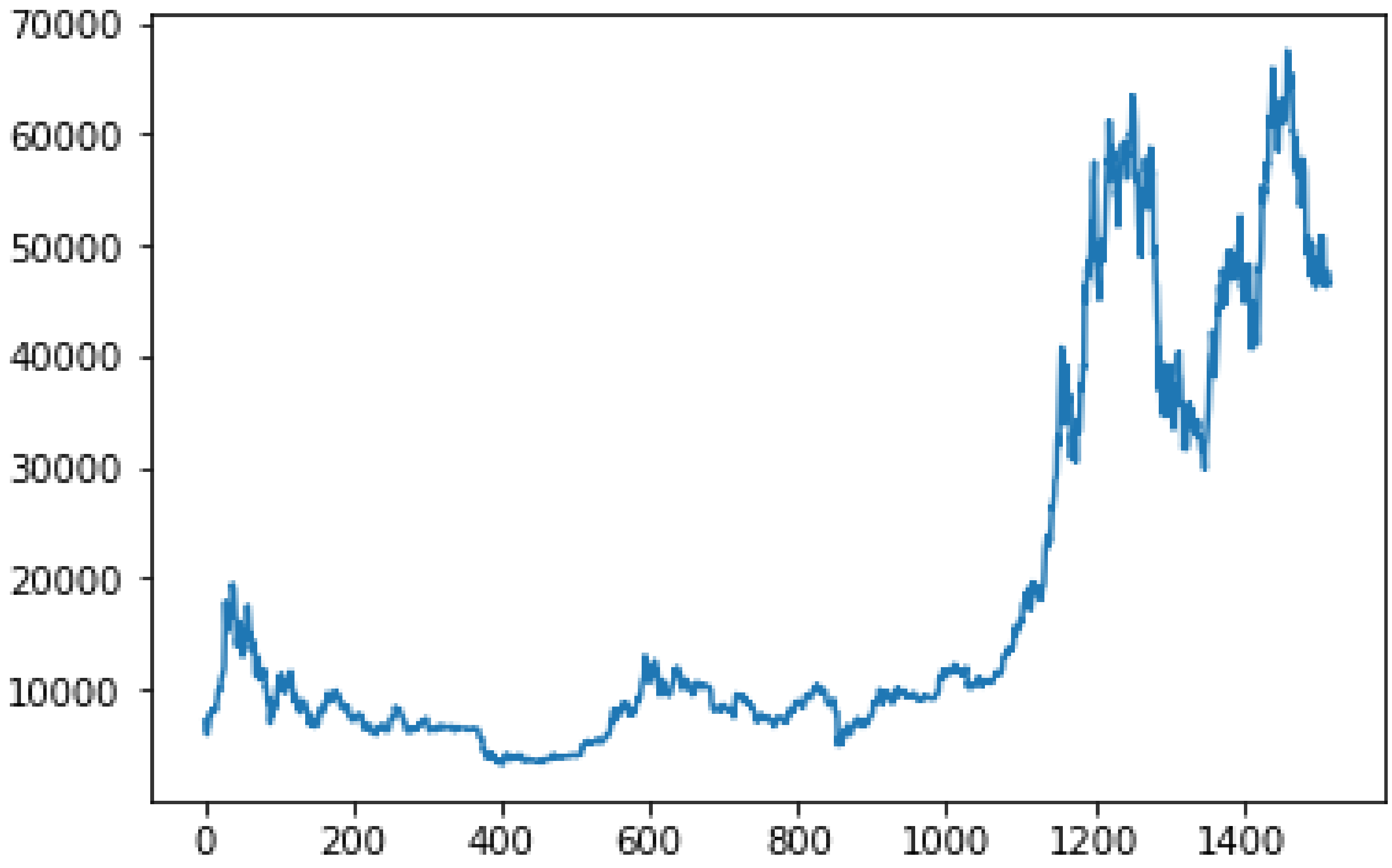


```
In [7]: df1 = df.reset_index()['Close']
```

```
In [8]: df1
```

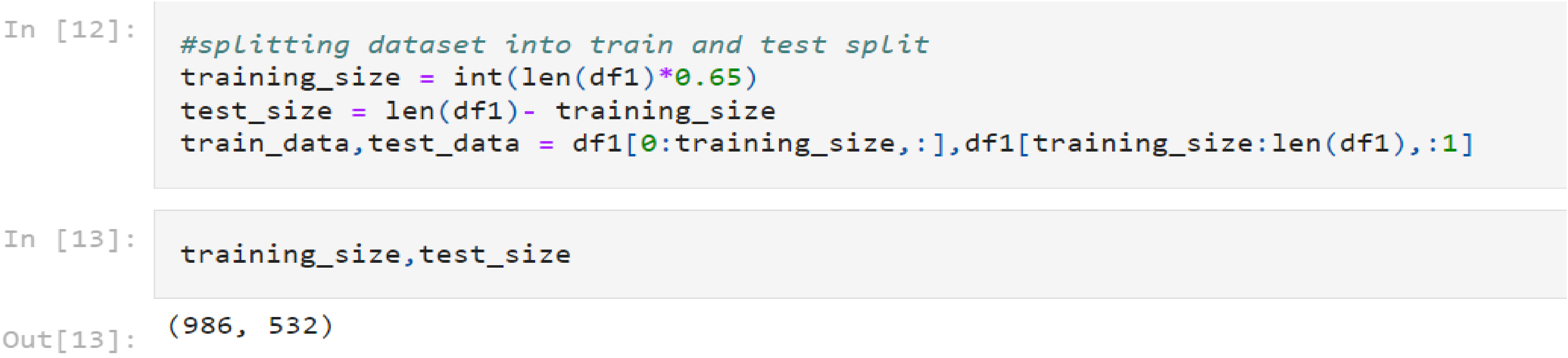
```
Out[8]: 0         7143.580078
        1         6618.140137
        2         6357.600098
        3         5950.069824
        4         6559.490234
                     ...
        1513     46306.445313
        1514     47686.812500
        1515     47345.218750
        1516     46458.117188
        1517     46545.582031
        Name: Close, Length: 1518, dtype: float64
```

We have plotted the data we have in our new dataframe so, just take a look at it:



•Machine Learning

1. We have to split our dataset into train and test split .
2. Here I have split the dataset as the 65% training and the rest is testing data.
3. This is the very important part in machine learning that we splitting data so that our model will perform better.



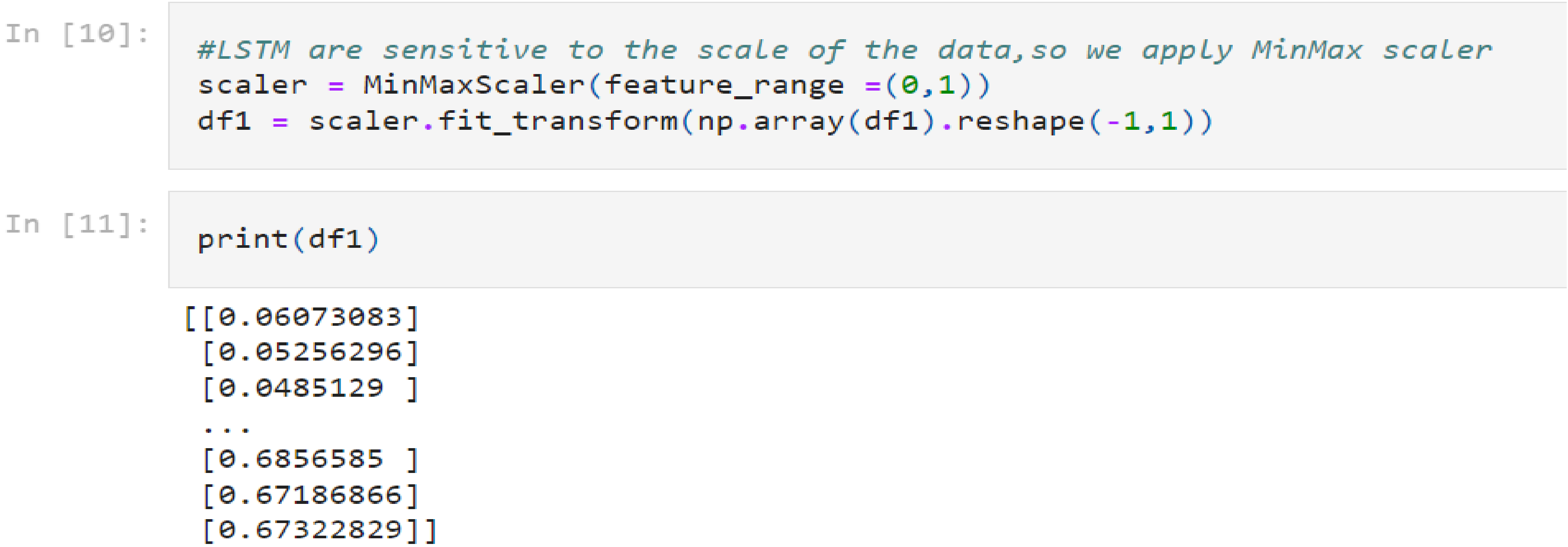
```
In [12]:  #splitting dataset into train and test split
          training_size = int(len(df1)*0.65)
          test_size = len(df1)- training_size
          train_data,test_data = df1[0:training_size,:],df1[training_size:len(df1),:1]

In [13]:  training_size,test_size

Out[13]:  (986, 532)
```

Data Normalization

- Since we are going to use LSTM for the prediction ,we have to normalize the data .
- LSTM are sensitive to the scale of the data.
- So we will apply MinMax Scaler just to get more accurate predictions.
- So we have applied the minmax normalization as follows:



In [10]:

```
#LSTM are sensitive to the scale of the data,so we apply MinMax scaler
scaler = MinMaxScaler(feature_range =(0,1))
df1 = scaler.fit_transform(np.array(df1).reshape(-1,1))
```

In [11]:

```
print(df1)
```

```
[[0.06073083]
 [0.05256296]
 [0.0485129 ]
 ...
 [0.6856585 ]
 [0.67186866]
 [0.67322829]]
```

- # Methodology

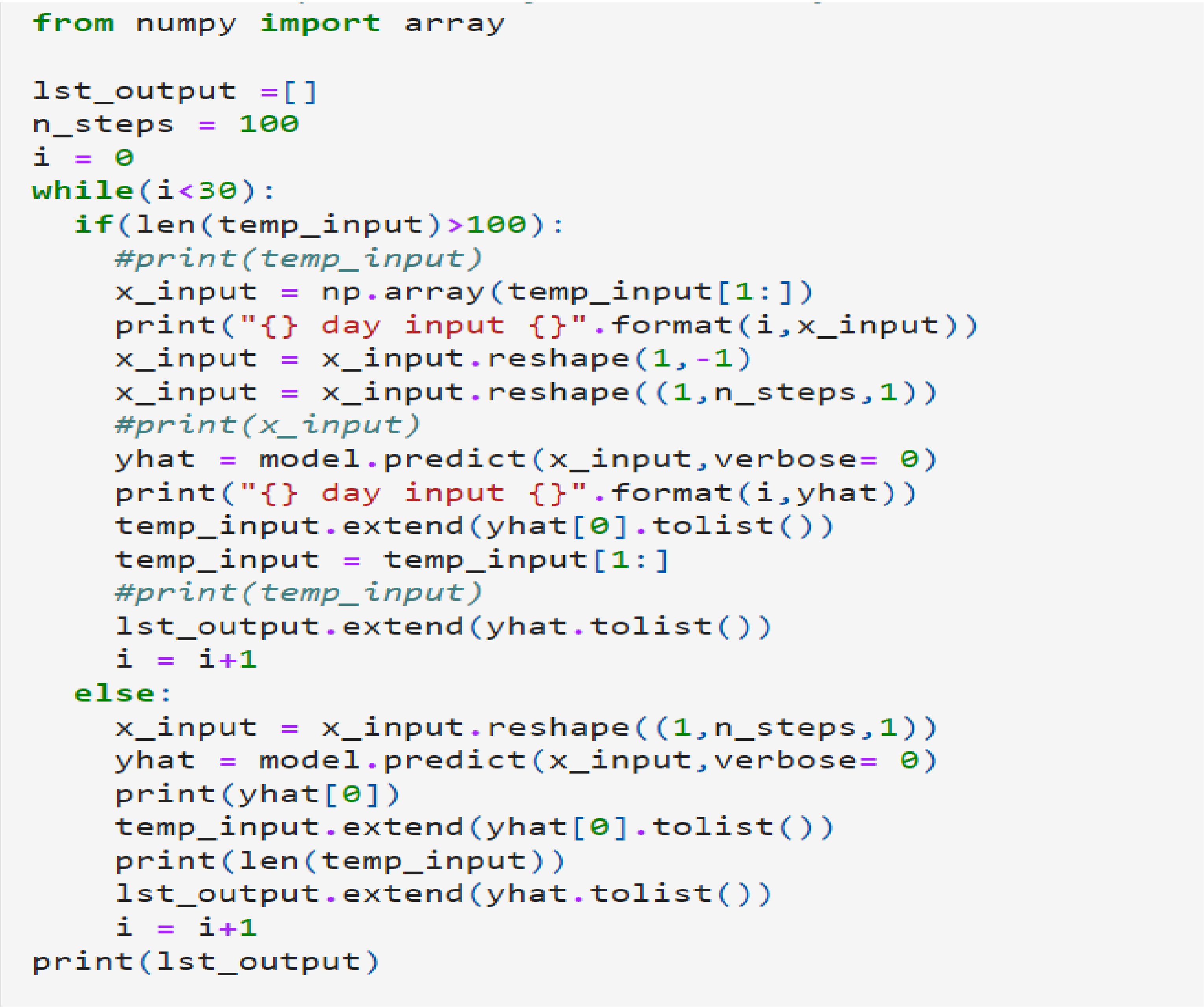
  ## We are predicting the movement of bitcoin using stacked LSTM

- Long Short Term Memory (LSTM) is an artificial recurrent neural network(RNN)
- How this model works?

1. Cryptocurrency market exhibits similar patterns at some extent.
2. LSTM is the most appropriate RNN for time series analysis as it processes data as it propagates forward.
3. So we can apply LSTM to predict the closing price of bitcoin from previous price data we have.

- # Discussion:

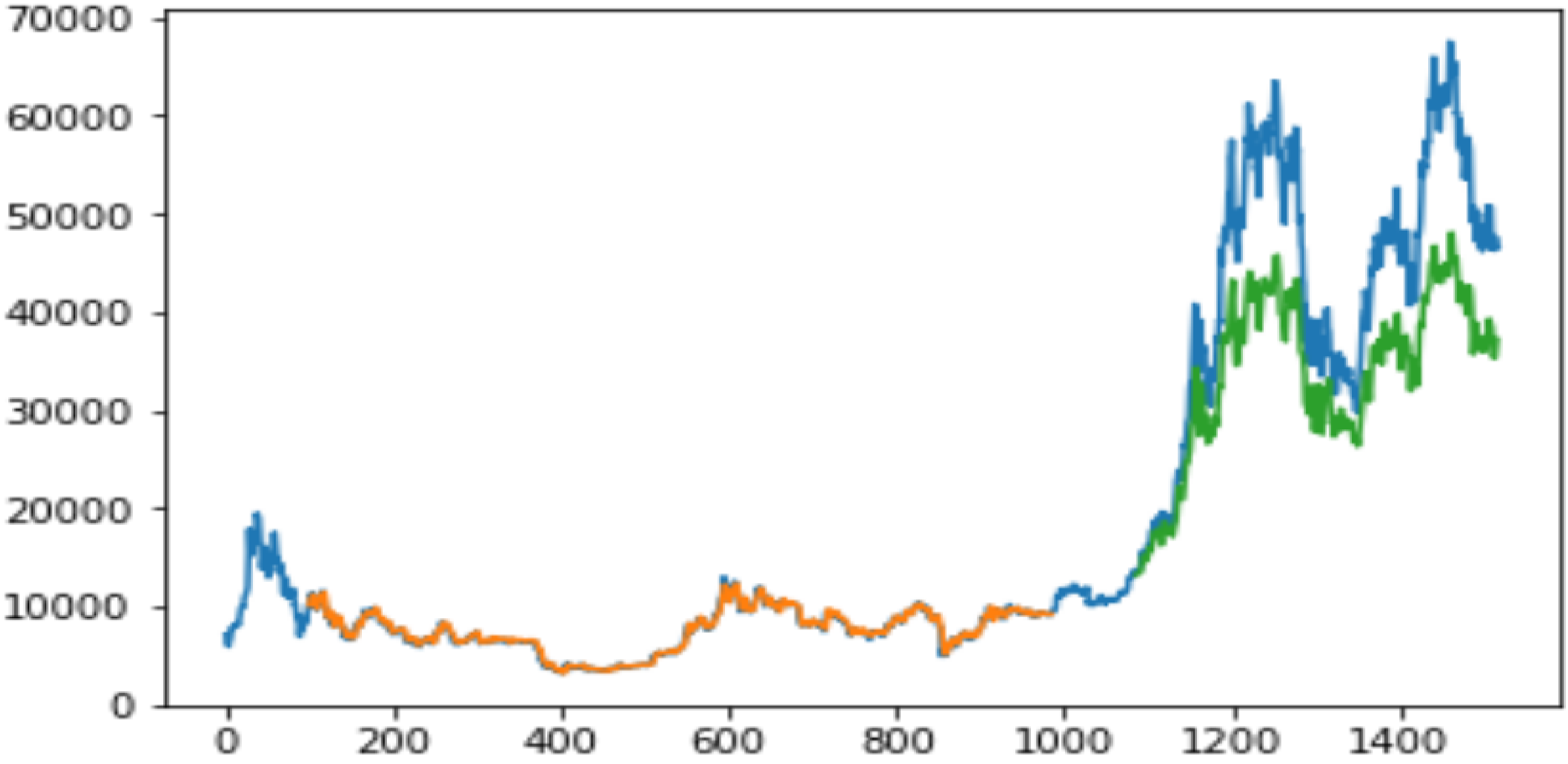
1. Here we are going to apply the logic in a way that we will get the data
2. In the way that we will collect first 100 records and we will relate the 101st record with that. And we will do this to the whole dataset and we will apply the same approach to predict the future values of bitcoin .
3. Since this is the time series data we have to handle this data in this way .
4. Because in time series dataset new occurrences are corelated with previous occurrences and we are applying the algorithm to get this corelation help us to predict future data.
5. The program we have used to predict the future values is this:



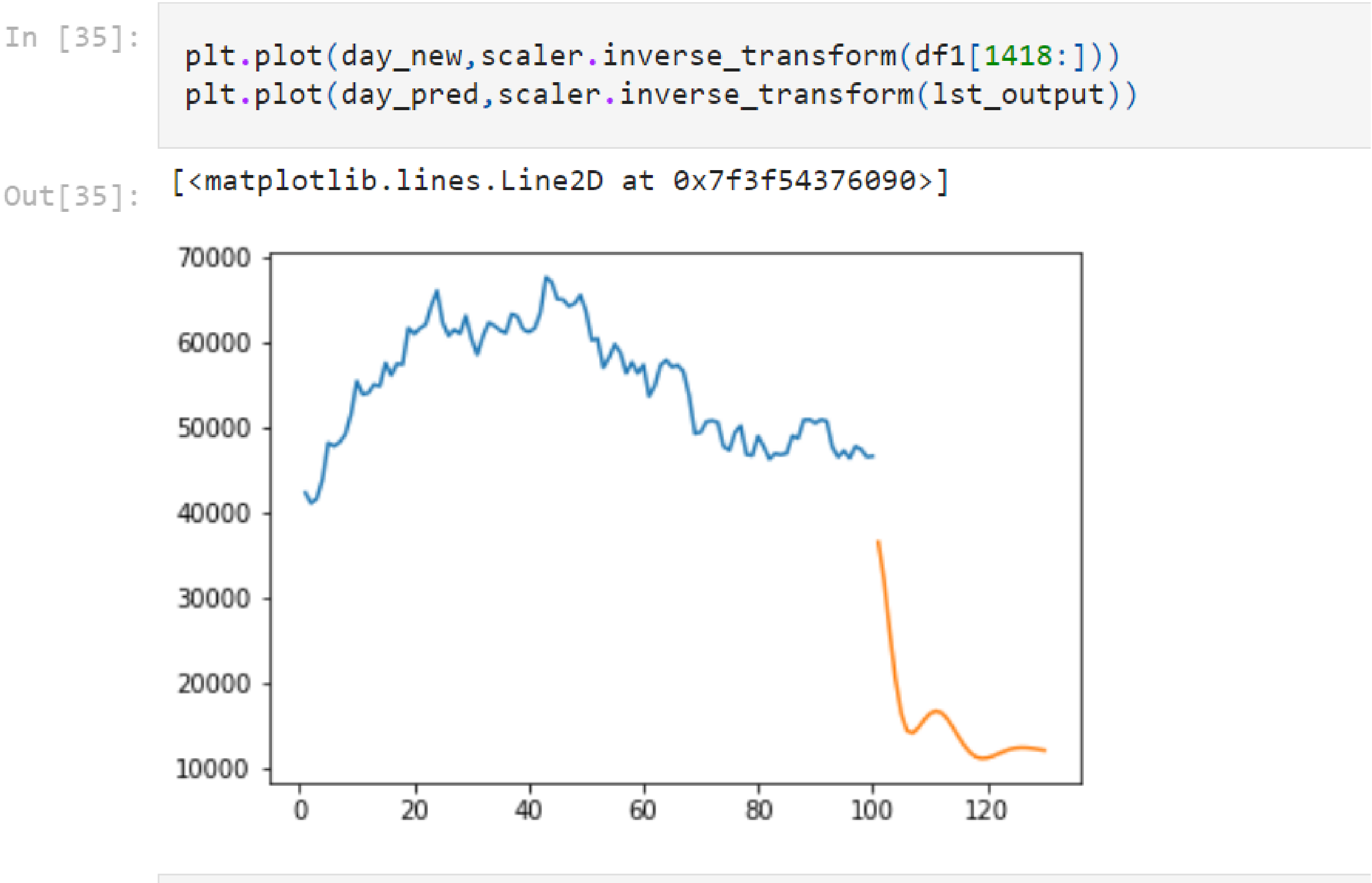
```
from numpy import array

lst_output =[]
n_steps = 100
i = 0
while(i<30):
  if(len(temp_input)>100):
    #print(temp_input)
    x_input = np.array(temp_input[1:])
    print("{} day input {}".format(i,x_input))
    x_input = x_input.reshape(1,-1)
    x_input = x_input.reshape((1,n_steps,1))
    #print(x_input)
    yhat = model.predict(x_input,verbose= 0)
    print("{} day input {}".format(i,yhat))
    temp_input.extend(yhat[0].tolist())
    temp_input = temp_input[1:]
    #print(temp_input)
    lst_output.extend(yhat.tolist())
    i = i+1
  else:
    x_input = x_input.reshape((1,n_steps,1))
    yhat = model.predict(x_input,verbose= 0)
    print(yhat[0])
    temp_input.extend(yhat[0].tolist())
    print(len(temp_input))
    lst_output.extend(yhat.tolist())
    i = i+1
print(lst_output)
```

This is what we got when i have plotted the training data



This is the final output we got after applying the Stacked LSTM to the data:



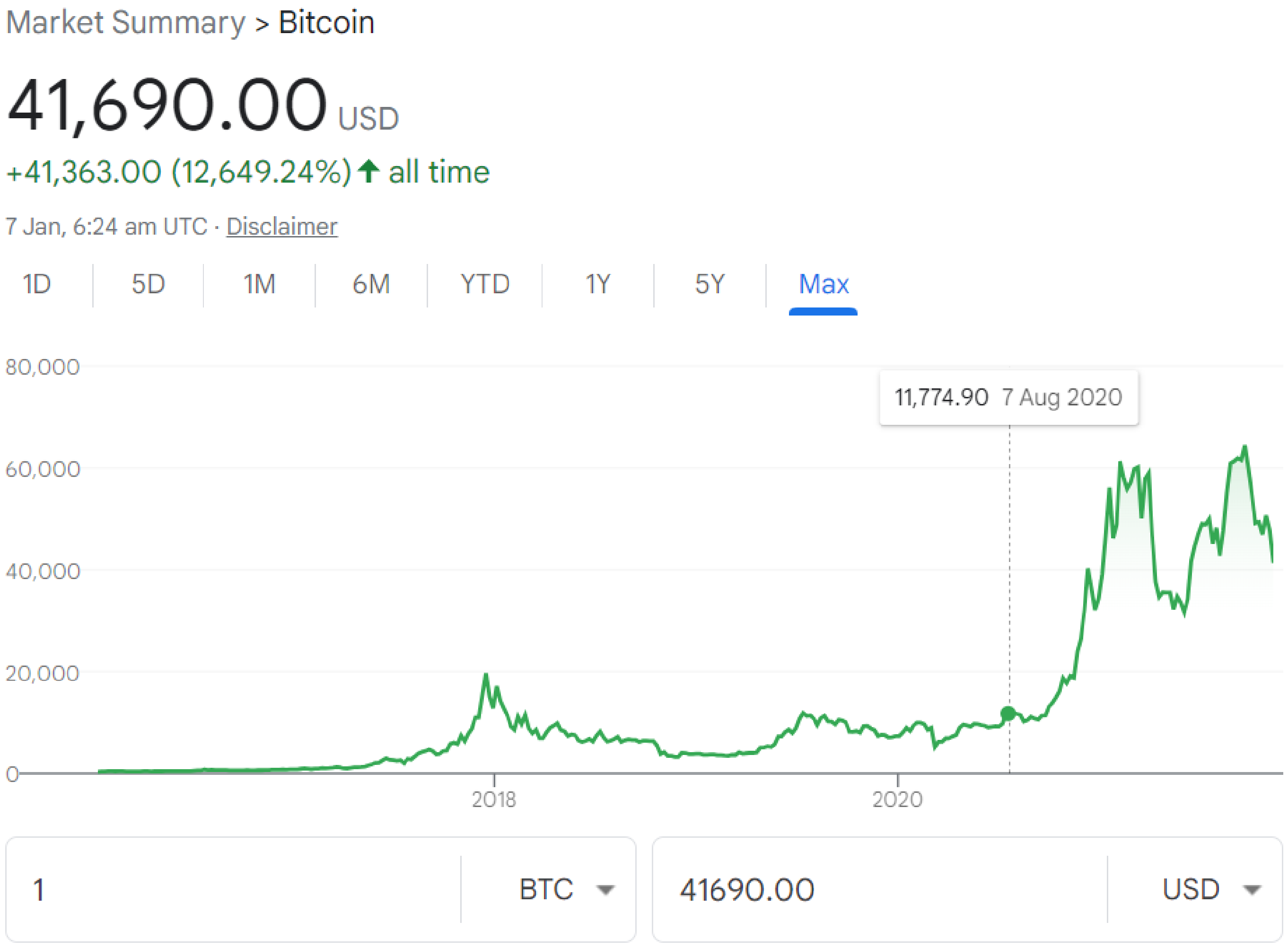
In [35]:

```
plt.plot(day_new,scaler.inverse_transform(df1[1418:]))
plt.plot(day_pred,scaler.inverse_transform(lst_output))
```

Out[35]:

```
[<matplotlib.lines.Line2D at 0x7f3f54376090>]
```

This is how bitcoin is increasing over the years



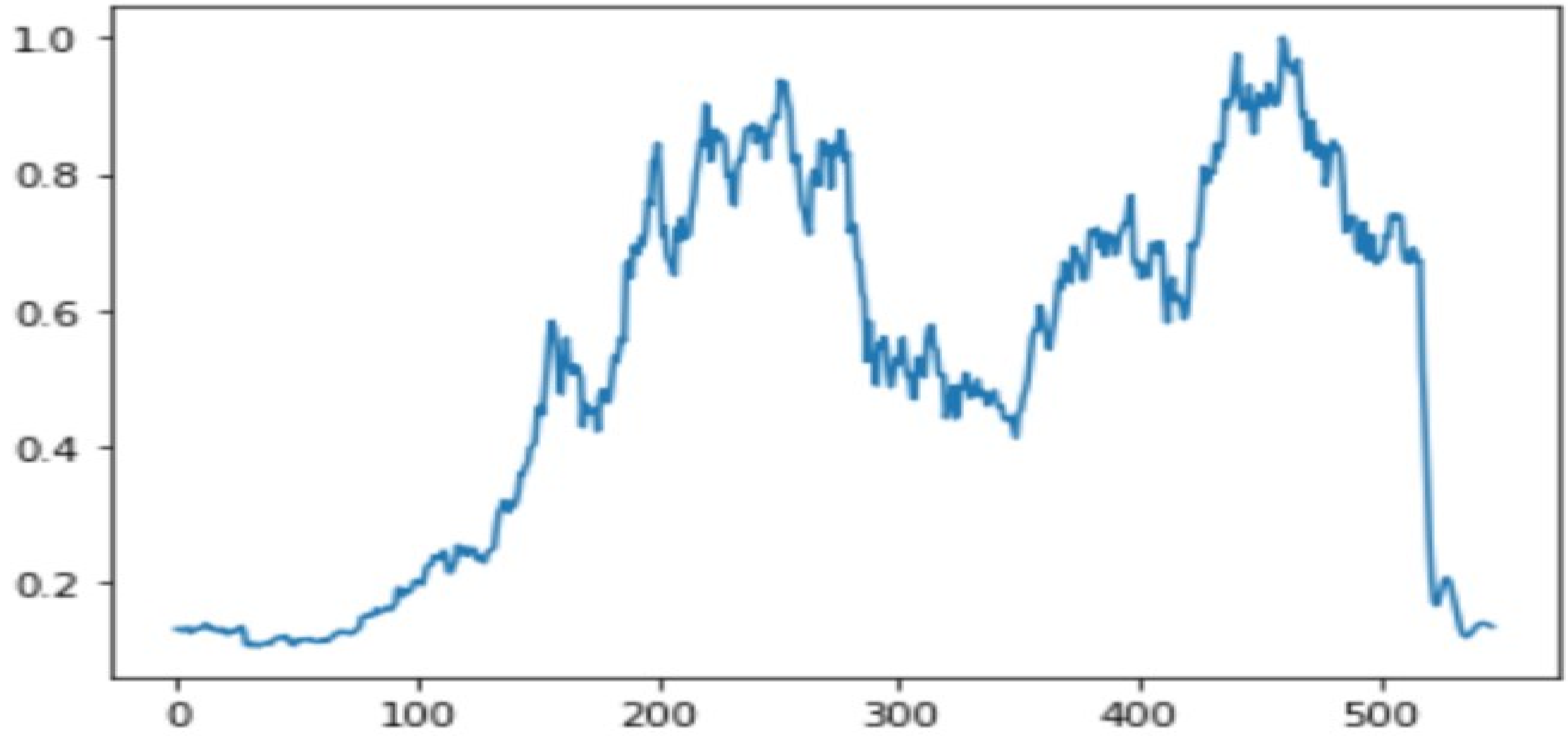
Market Summary > Bitcoin

41,690.00 USD

+41,363.00 (12,649.24%) ↑ all time

7 Jan, 6:24 am UTC · Disclaimer

This is what we came up with on the large scale



Conclusion:

- We can predict Bitcoin using the previous data in our dataset.
- Using LSTM we were able to achieve but this accuracy is varies as per the nature of the investors.
- LSTM is not great at predicting short term price fluctuation because it is mainly based on the overall s sentiments of the market.
- That’s why we can see a sharp drop in the predicted values which is surely based on the last couple of negative weeks.

Outcome:

I will come up with the new strategies by adding more features into consideration to achieve the highest accuracy in the future model.