NAME: Yashika Nihalani
UBITNAME: yashikav
UBIT: 50425015

NAME: Ashutosh Shailesh Bhawsar
UBITNAME: abhawsar
UBIT: 50416025

PROF: ERIC MIKIDA

DIC Phase 1

1) **Title:** *Cryptocurrency data analysis and investing strategies*

Problem Statement:

- ➔ *Analyze the bitcoin data and realize trends in price, volume, and weighted average using time series analysis.*
- ➔ *Design suitable prediction algorithms/models to determine the price of bitcoin.*
- ➔ *Develop strategies to help an investor with trading decisions.*

a) **Motivation and problem:**

- The use of cryptocurrencies has grown significantly in recent years. One of the digital assets, Bitcoin's value has surged by over 200%, reaching new heights before tumbling back down.
- The market's volatility is a significant disadvantage of cryptocurrency
- Trading takes place 24/7, and tracking it is a challenging task.
- It is tough / nearly impossible for humans or companies to manually analyze the trends in cryptocurrency daily.
- Companies need some form of computational data to design their strategies for investments.

b) **Solution and impact:**

- Analyze the data and engineer features to realize underlying patterns through visualizations to understand the dataset properly.
- Build different prediction models and determine future market movements.
- Devise suitable investment/trading strategies to maximize profit for investments.
- Expose the visualizations and build interactive UI for users.

2) **Data Sources:**
    a) **Kaggle** - https://www.kaggle.com/datasets/mczielinski/bitcoin-historical-data

3) **Data cleaning / Preprocessing:**
    a) **Conversion of Unix time to Datetime - Timestamp column**

The timestamp column is provided in UNIX time, which means the number of seconds that have elapsed since January 1, 1970. We have converted this column into DateTime format, which is more readable and helps visualize the data according to the date and time in minutes.

```
          Timestamp      Open      High       Low     Close  Volume_(BTC)  \
0        1325317920      4.39      4.39      4.39      4.39      0.455581
1        1325317980       NaN       NaN       NaN       NaN           NaN
2        1325318040       NaN       NaN       NaN       NaN           NaN
3        1325318100       NaN       NaN       NaN       NaN           NaN
4        1325318160       NaN       NaN       NaN       NaN           NaN
...             ...       ...       ...       ...       ...           ...
4857372  1617148560  58714.31  58714.31  58686.00  58686.00      1.384487
4857373  1617148620  58683.97  58693.43  58683.97  58685.81      7.294848
4857374  1617148680  58693.43  58723.84  58693.43  58723.84      1.705682
4857375  1617148740  58742.18  58770.38  58742.18  58760.59      0.720415
4857376  1617148800  58767.75  58778.18  58755.97  58778.18      2.712831

         Volume_(Currency)  Weighted_Price
0                 2.000000        4.390000
1                      NaN             NaN
2                      NaN             NaN
3                      NaN             NaN
4                      NaN             NaN
...                    ...             ...
4857372       81259.372187    58692.753339
4857373      428158.146640    58693.226508
4857374      100117.070370    58696.198496
4857375       42332.958633    58761.866202
4857376      159417.751000    58764.349363

[4857377 rows x 8 columns]
```

    b) **Dropping all NAN or null values**

A few columns contained NAN values of no meaningful use for analysis. Hence the rows with such values are deleted. The pandas function dropna() is used, which determines the NAN values and drops them from the dataset.

	Timestamp	Open	High	Low	Close	Volume_BTC	Volume_Currency	Weighted_Price
0	2011-12-31 07:52:00	4.39	4.39	4.39	4.39	0.455581	2.000000	4.390000
478	2011-12-31 15:50:00	4.39	4.39	4.39	4.39	48.000000	210.720000	4.390000
547	2011-12-31 16:59:00	4.50	4.57	4.50	4.57	37.862297	171.380338	4.526411
548	2011-12-31 17:00:00	4.58	4.58	4.58	4.58	9.000000	41.220000	4.580000
1224	2012-01-01 04:16:00	4.58	4.58	4.58	4.58	1.502000	6.879160	4.580000
...	...	...	...	...	...	...	...	...
4857372	2021-03-30 23:56:00	58714.31	58714.31	58686.00	58686.00	1.384487	81259.372187	58692.753339
4857373	2021-03-30 23:57:00	58683.97	58693.43	58683.97	58685.81	7.294848	428158.146640	58693.226508
4857374	2021-03-30 23:58:00	58693.43	58723.84	58693.43	58723.84	1.705682	100117.070370	58696.198496
4857375	2021-03-30 23:59:00	58742.18	58770.38	58742.18	58760.59	0.720415	42332.958633	58761.866202
4857376	2021-03-31 00:00:00	58767.75	58778.18	58755.97	58778.18	2.712831	159417.751000	58764.349363

3613769 rows × 8 columns

c) Grouping open column

The data is for every minute; we have grouped it according to day by providing the parameter frequency= “D.” Thus, all the data of a particular date are grouped together. The first available data in the dataset for that particular day is considered for the open column.

d) Grouping High column

The data for the same day are grouped together, and the max value for the high column that is the highest of the day is taken.

e) Grouping Low column

The minimum value available in the dataset for each day is taken for the low column.

f) Grouping Close column

The last available data in the dataset for that particular day is considered for the close column.

g) Grouping Volume_BTC, Volume_Currency, and Weighted_price

The mean data for volume_BTC, volume currency, and weighted price for each day are taken in respective columns.

Timestamp	Open	Close	High	Low	Volume_BTC	Volume_Currency	Weighted_Price
2011-12-31	4.39	4.58	4.58	4.39	23.829470	106.330084	4.471603
2012-01-01	4.58	5.00	5.00	4.58	7.200667	35.259720	4.806667
2012-01-02	5.00	5.00	5.00	5.00	19.048000	95.240000	5.000000
2012-01-03	5.32	5.29	5.32	5.14	11.004660	58.100651	5.252500
2012-01-04	4.93	5.57	5.57	4.93	11.914807	63.119577	5.208159
...	...	...	...	...	...	...	...
2021-03-27	55081.26	55839.42	56686.15	53948.35	1.823877	100884.732367	55193.357260
2021-03-28	55817.85	55790.92	56573.04	54677.51	1.447939	80632.115263	55832.958824
2021-03-29	55790.28	57600.10	58402.68	54892.42	3.732887	213754.555988	56913.993819
2021-03-30	57623.66	58760.59	59388.66	57011.00	2.363999	138231.241926	58346.912268
2021-03-31	58767.75	58778.18	58778.18	58755.97	2.712831	159417.751000	58764.349363

3379 rows × 7 columns

h) Compute and add a column for the change percentage daily

We computed the change percentage for each day by comparing the current day's closing price with the previous day's closing price. This column will contribute to analyzing the change and dividing the day into a particular category.

Timestamp	Open	Close	High	Low	Volume_BTC	Volume_Currency	Weighted_Price	Change_percentage
2011-12-31	4.39	4.58	4.58	4.39	23.829470	106.330084	4.471603	NaN
2012-01-01	4.58	5.00	5.00	4.58	7.200667	35.259720	4.806667	9.170306
2012-01-02	5.00	5.00	5.00	5.00	19.048000	95.240000	5.000000	0.000000
2012-01-03	5.32	5.29	5.32	5.14	11.004660	58.100651	5.252500	5.800000
2012-01-04	4.93	5.57	5.57	4.93	11.914807	63.119577	5.208159	5.293006
...	...	...	...	...	...	...	...	...
2021-03-27	55081.26	55839.42	56686.15	53948.35	1.823877	100884.732367	55193.357260	1.376237
2021-03-28	55817.85	55790.92	56573.04	54677.51	1.447939	80632.115263	55832.958824	-0.086856
2021-03-29	55790.28	57600.10	58402.68	54892.42	3.732887	213754.555988	56913.993819	3.242786
2021-03-30	57623.66	58760.59	59388.66	57011.00	2.363999	138231.241926	58346.912268	2.014736
2021-03-31	58767.75	58778.18	58778.18	58755.97	2.712831	159417.751000	58764.349363	0.029935

3379 rows × 8 columns

i) Round all integer data in DataFrame to 2 decimal places

Having a scale of 6 digits is unnecessary for us since those are the least significant. Hence we round up all integers in our dataset to the scale of 2.

j) Convert data frame to NumPy

Now that the data is cleaned, to make it ready for preprocessing and apply machine learning models, we convert the DataFrame from pandas to NumPy.

k) Divide the data columns into test and train data

The entire data is divided into train data and test data. After processing the model with train data, our model will be tested using the test data and hence make predictions.

4) Exploratory Data Analysis (EDA):

a) Describe data to get mathematical insights-

To look into mathematical insights of data, we have described the data using the describe() method. It can be inferred that the mean of weighted price is the most and the mean of our added column change percentage is the least. Many other insights, such as the min and max values in each column, can be viewed.

	Open	Close	High	Low	Volume_BTC	Volume_Currency	Weighted_Price	Change_percentage
count	3376.000000	3376.000000	3376.000000	3376.000000	3376.000000	3376.000000	3376.000000	3378.000000
mean	4602.417399	4619.687260	4750.700598	4442.507965	10.355643	31790.810193	4605.576496	0.386519
std	8193.870228	8245.987435	8497.261901	7874.336609	8.897324	62753.976370	8207.031563	4.564684
min	3.800000	4.230000	4.380000	1.500000	0.250000	1.230000	4.330000	-48.520000
25%	244.792500	244.940000	249.777500	239.952500	4.670000	1916.185000	244.952500	-1.210000
50%	696.020000	697.120000	716.465000	668.265000	7.620000	6832.005000	697.945000	0.215000
75%	7249.760000	7257.850000	7430.267500	7058.395000	13.112500	36074.510000	7242.892500	1.970000
max	61177.030000	61165.190000	61781.830000	58959.570000	119.520000	950995.600000	60455.840000	40.140000

b) Get detailed information on data for coding insights-

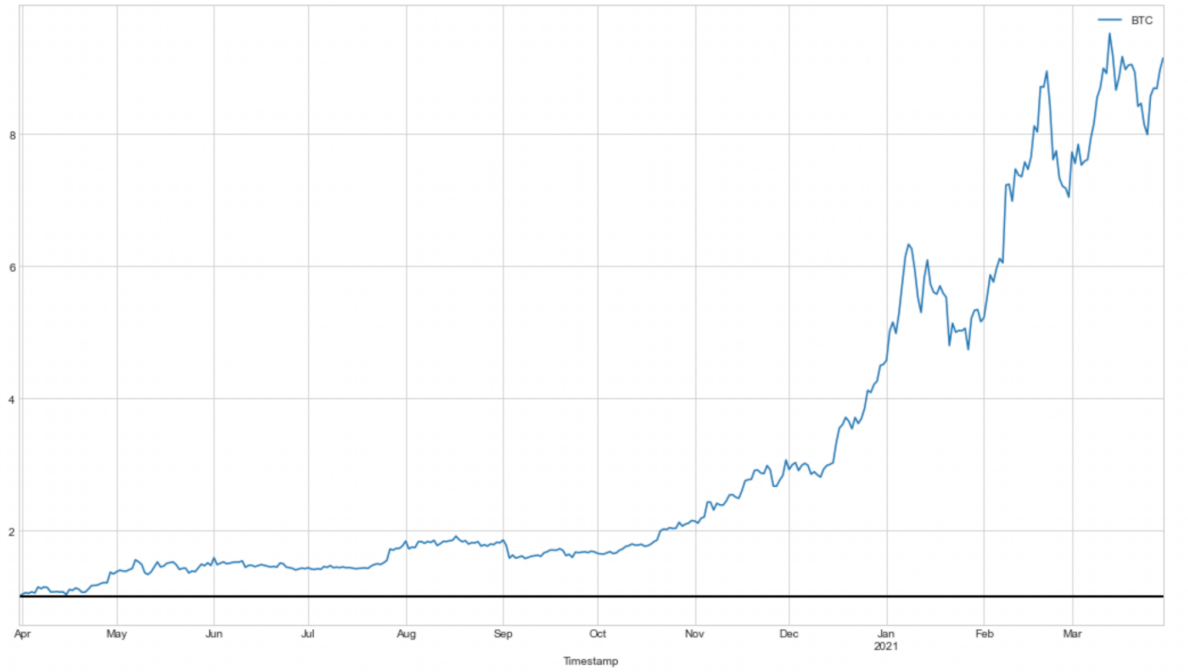
Using the info method, we can get information about the details, such as the count of all rows in the column, the number of the not null values, and the data type of each column. It is seen that all the columns have data types as float.

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 3379 entries, 2011-12-31 to 2021-03-31
Freq: D
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Open               3376 non-null   float64
 1   Close              3376 non-null   float64
 2   High               3376 non-null   float64
 3   Low                3376 non-null   float64
 4   Volume_BTC         3376 non-null   float64
 5   Volume_Currency    3376 non-null   float64
 6   Weighted_Price     3376 non-null   float64
 7   Change_percentage  3378 non-null   float64
dtypes: float64(8)
memory usage: 237.6 KB
```

c) Cumulative returns-

We took the last closing price precisely one year ago. The cumulative daily returns were calculated by dividing the price by the above-calculated closing price. From the graph, it inferred that starting from April, if $1 was invested, in around Jan 2021, it increased to $6, and in March 2021, it became 8 times - $8

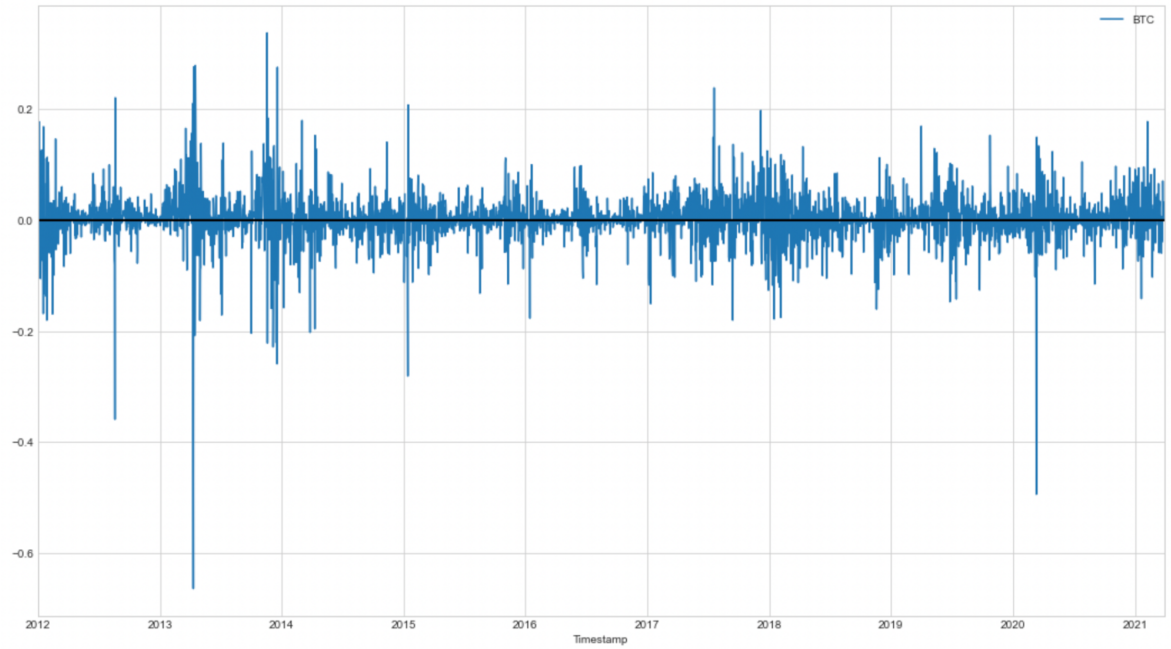
```
6421.14
Index(['BTC'], dtype='object')

<matplotlib.lines.Line2D at 0x7fca1a285e50>
```



d) Daily returns-

Using the formula, the daily log returns are calculated. Logarithmic returns are valid for mathematical finance. One of the advantages is that the logarithmic returns are symmetric. While ordinary returns are not, logarithmic returns of equal magnitude but opposite signs will cancel each other out.
Considering the baseline, we can see from the plot that in around the 4th month of 2013, it's the highest loss which is more than 65%.
In the first two months of 2013 and the last 2 months of 2017, it would have been a good time to invest as the stock price was as high as 25%.

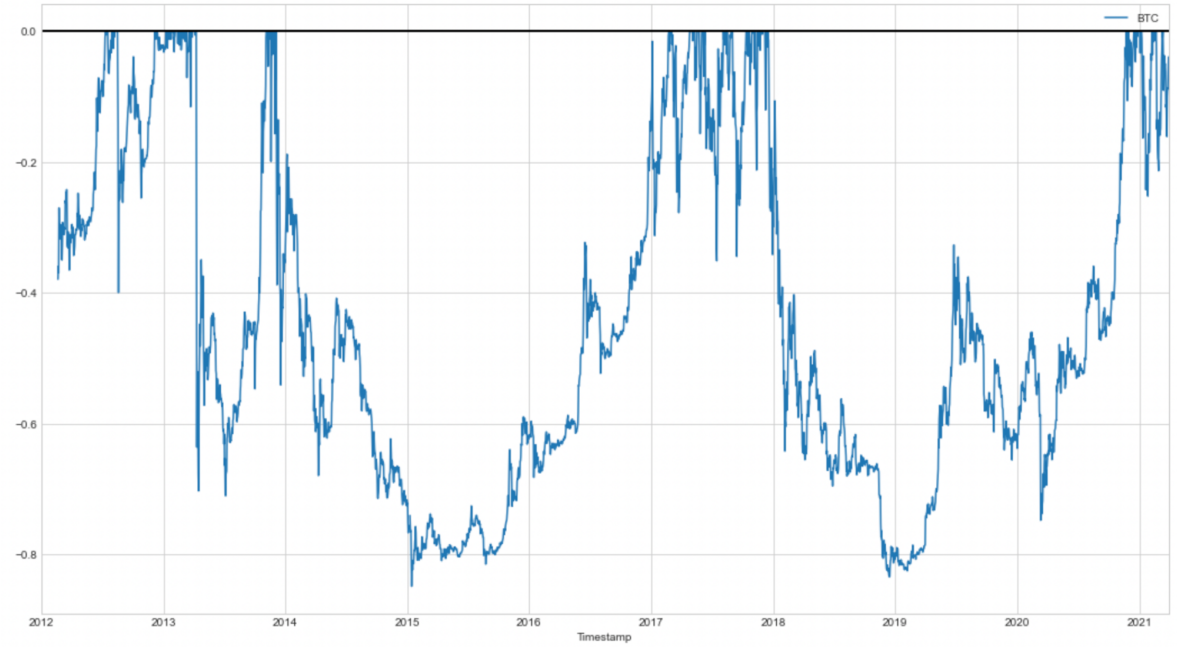
```
Index(['BTC'], dtype='object')

<matplotlib.lines.Line2D at 0x7fc9d8bb37f0>
```



e) Drawdown-

Drawdowns measure the downside volatility of a stock/crypto price. We compare each price point to the previous running peak, which is further used to estimate the historical risks associated with an investment in cryptocurrencies. Drawdowns and losses can be confused to be the same thing, but the key difference is that drawdown is measured from the previous local peak and is a peak-to-trough metric. We infer that the maximum drawdown for BTC is 84.85%, meaning the investor's portfolio could've been 84.85% down in the window of buying and selling.

```
BTC   -0.848579
dtype: float64
```

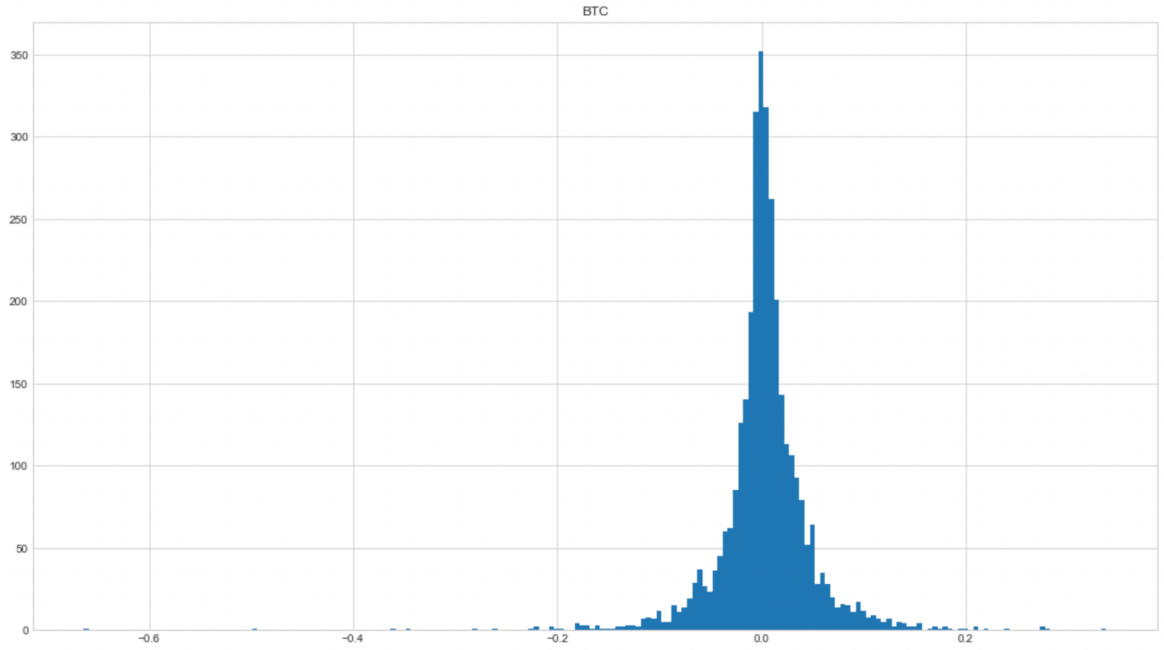


f) Skewness and kurtosis-

Skewness measures the asymmetry of the probability distribution about its mean. Skewness differentiates extreme values in one versus the other tail. Similarly, kurtosis measures extreme values in either tail of the distribution. The base of kurtosis is used as 3. After applying these to our data, we infer that skewness is negative (-1.41), meaning the direction of outliers is on the left(loss) side of the distribution. A high kurtosis value (23.41) implies that the investor will experience occasional extreme returns in either profit or loss.

```
Skewness:  BTC   -1.415538
dtype: float64
Kurtosis:  BTC    23.413106
dtype: float64
```

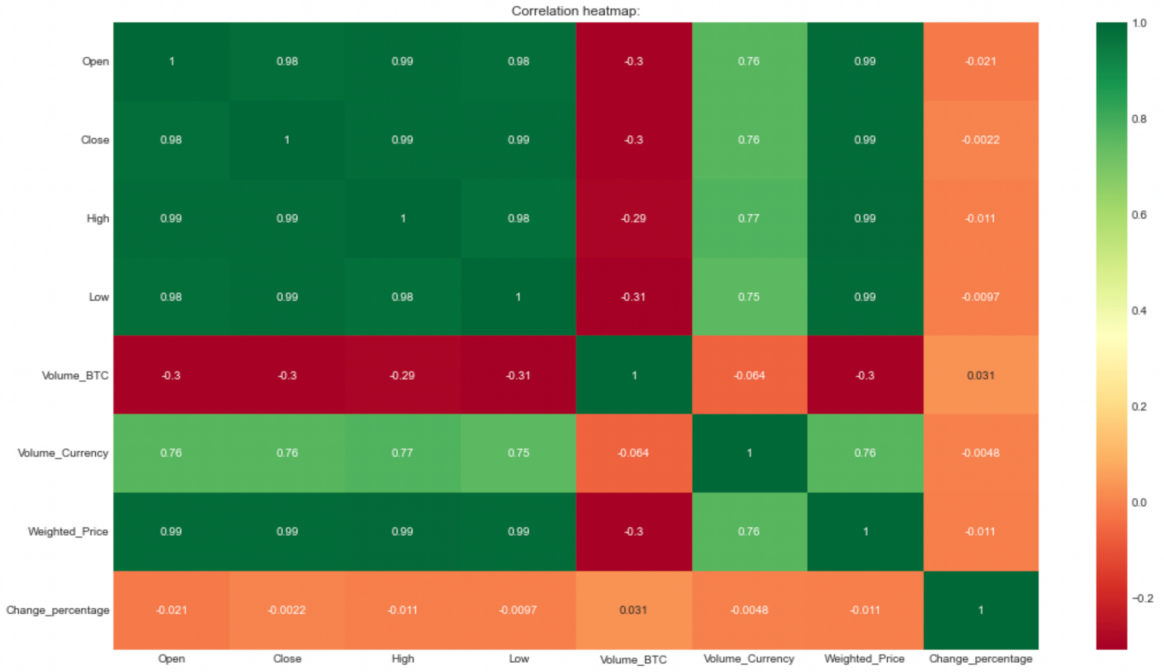
```
array([[<AxesSubplot:title={'center':'BTC'}>]], dtype=object)
```



g) Correlation analysis-

From the Correlation Analysis graph, it is evident that the following features- High, Weighted Price, Close, Open, Low are strongly correlated. The remaining features - Volume_BTC and Volume_Currency are closely related.

```
Text(0.5, 1.0, 'Correlation heatmap:')
```

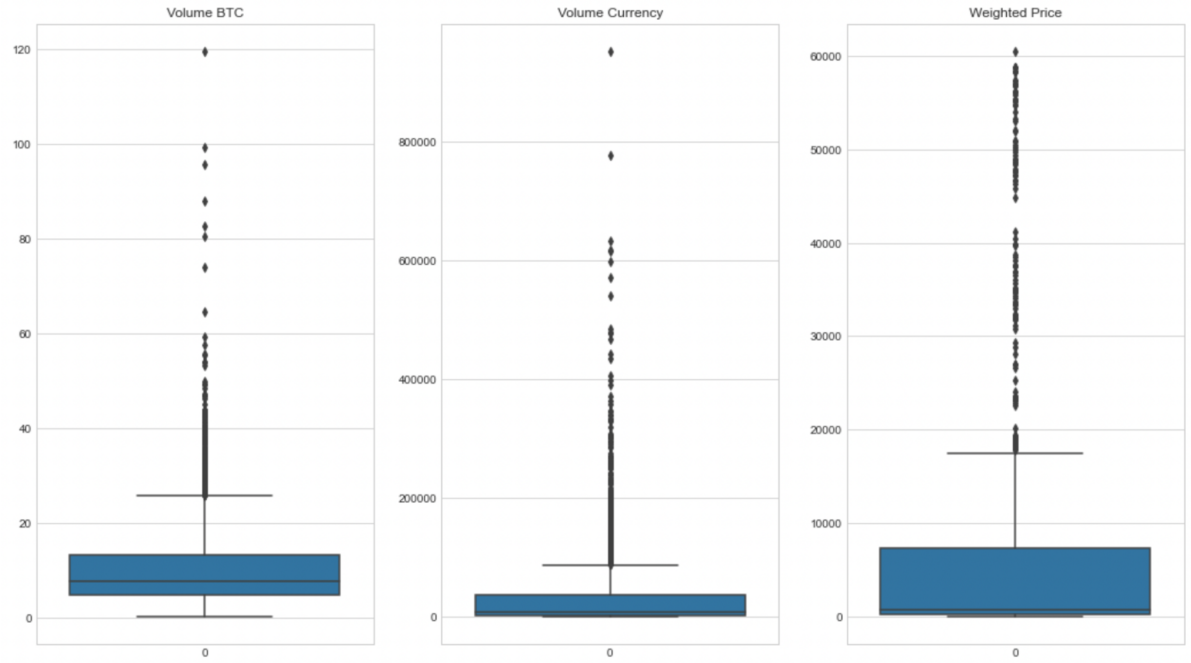


h) Outliers detection using boxplot-

Volume BTC- The frequency of the days when 40 to 60 bitcoins were traded is more than the days more than 100 bitcoins were traded. It can be seen that the number of days 120 bitcoins were traded is an outlier and can be removed in further processing.

Volume Currency- The number of days when USD in a range of 200000 to 600000 were traded is more—the number of days when 800000 USD or more were traded is the least and is an outlier.

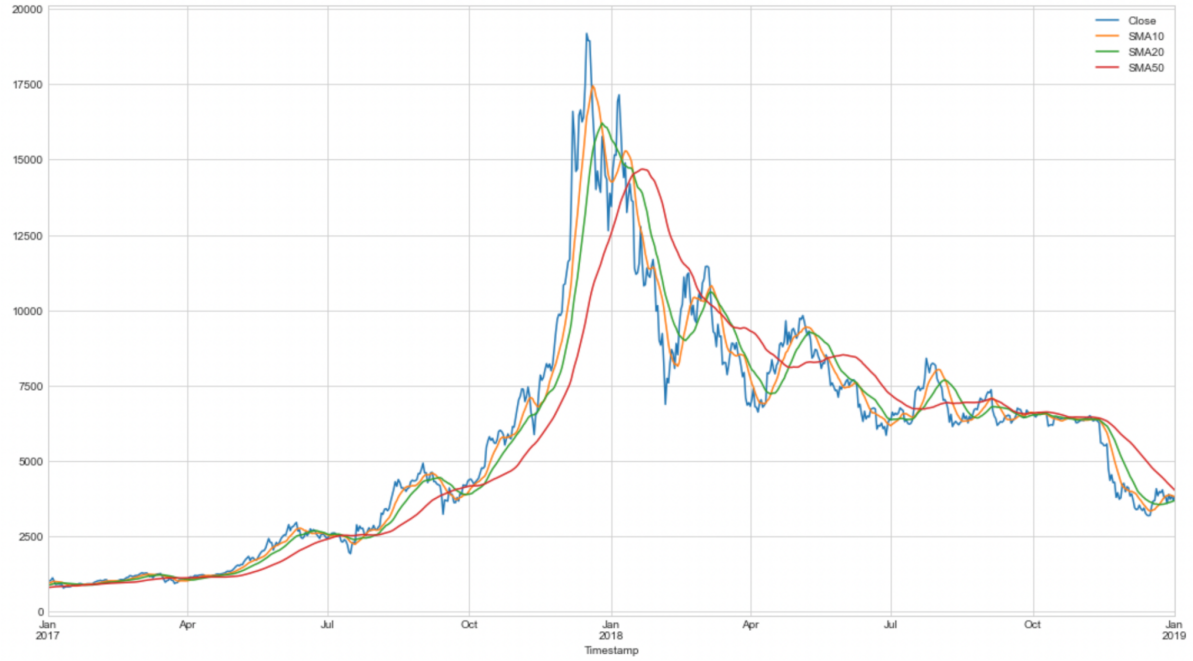
Weighted Price- The distribution for the weighted price is almost equal among different ranges, and few outliers can be detected.



i) SMAs to detect golden and death cross-

A golden cross is a strong indicator of a bull market. A death cross is a strong indicator of a bear market. This is further used for analysis in the next point.

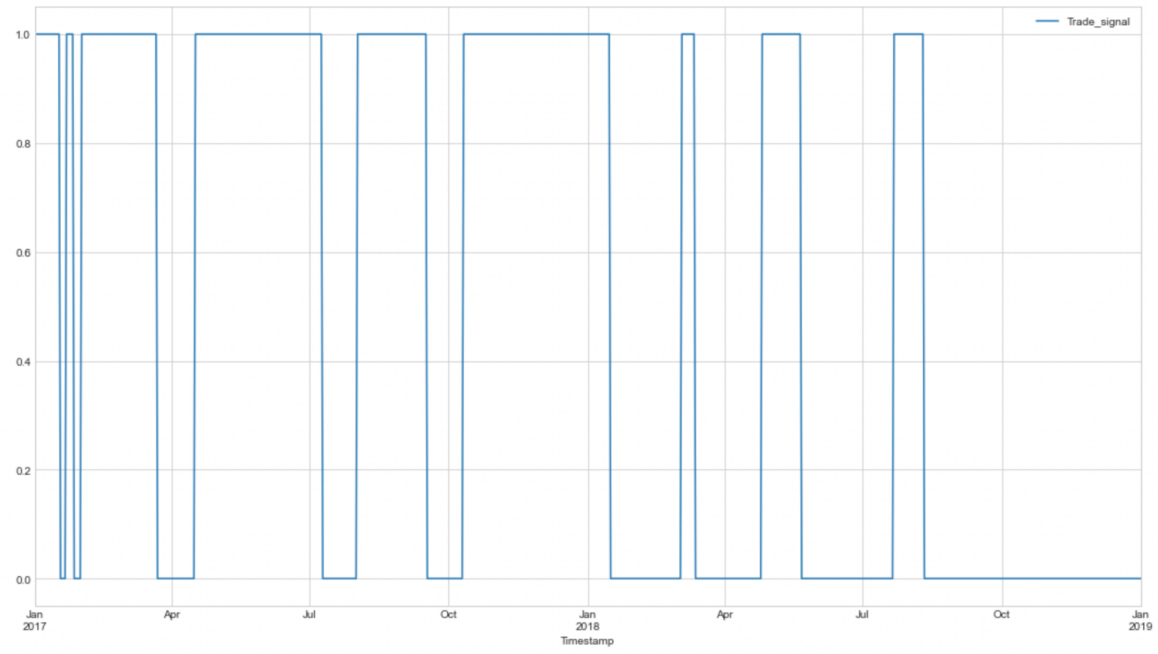
```
<AxesSubplot:xlabel='Timestamp'>
```



j) Adding true labels to dataset based on SMAs-

A new label is assigned to data based on the simple moving average. If the SMA of 10 days and SMA of 20 days are greater than the SMA of 50 days, then accordingly, we are assigning 1 as the label, which signifies a Golden cross. And if the above is invalid, it is assigned as 0, which means death cross.

```
<AxesSubplot:xlabel='Timestamp'>
```



5) **REFERENCES:**

a) Drawdown Definition
b) Golden Cross Pattern Explained With Examples and Charts
c) Positively and Negatively Skewed Defined with Formula
d) Kurtosis Definition, Types, and Importance
e) https://www.kaggle.com/code/urayukitaka/time-series-analysis-of-bitcoin/notebook - Reference for SMA idea
f) https://www.kaggle.com/code/shtrausslearning/building-an-asset-trading-strategy - Literature survey and motivation reference