**Index**

1. Introduction………………………………………………………….. 2
2. Analysis……………………………………………………………… 2
3. Resampling Dataset on Daily Basis……………………………...….. 3
4. Splitting of Dataset…………………………………………………… 4
5. Seasonal Decompose…………………………………………………. 4
6. Long Short-Term Memory (LSTM)………………………………….. 5
7. Reshaping Data……………………………………………………….. 7
8. LSTM Model Process………………………………………………… 8
9. Validation Loss……………………………………………………….. 10
10. Predictions……………………………………………………………. 11
11. Visualization of Predictions…………………………………………… 13
12. Evaluation……………………………………………………………… 14
13. Evaluation2……………………………………………………………. 14
14. References……………………………………………………………. 15

**Final Report**

**Introduction**

The bitcoin dataset is defining time series data as the value of bitcoin changes with span of time. The EDA was exploring basic trends and patterns of dataset. In this part, we are working on models to predict some interesting facts about the changes in price of bitcoin. In EDA, time resampling was done on daily basis and also considered for modelling too. This type of resampling is more consistent with time series **LSTM (LONG SHORT-TERM MEMORY)** model.

The analysis consists seasonal decompose or seasonality patterns. Further, it involves spitting of dataset to validate the model. We have worked with **Vanilla LSTM** model and forecasted results. The predictions are made and a detailed analysis will be given in next section.

**Analysis**

The structure of dataset with datetime index is shown below:

**Graphical user interface, text

Description automatically generated**

The starting and end time of dataset specify a period of 4 years which is shown below:

**Text

Description automatically generated**

The datatset contains missing values which are processed. The statistics are shown below:

**Text

Description automatically generated A picture containing text

Description automatically generated**

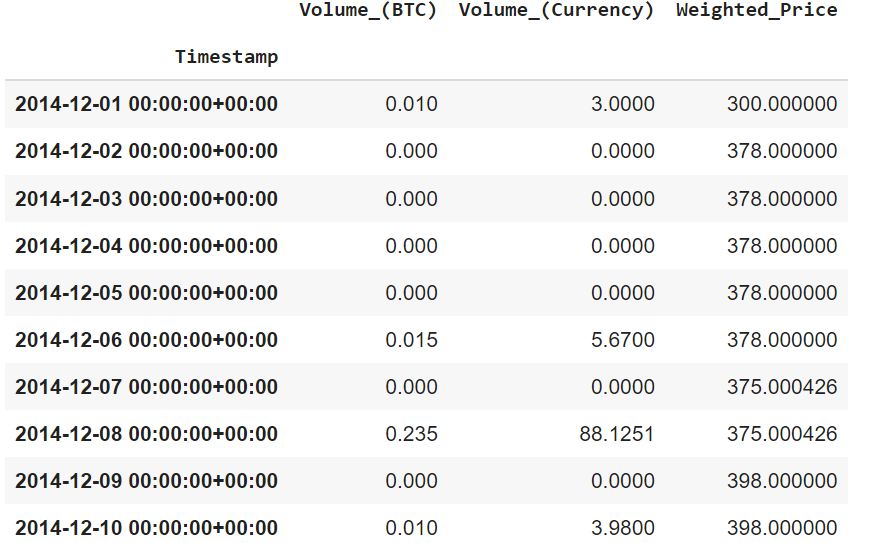
The first glimpse of dataset showing all the variables is given in the table below: **A picture containing table

Description automatically generated**

**Resampling Dataset on Daily Basis**

In EDA, the analysis was done with daily dataset to observe general trends of variables. And in this part, the **modelling is also done on dailly data.**

The processing of Daily resampling is done by taking the weighted price and replacing the zero value with null values that can also be said as missing values. Then these null values are taken to exchange it with the last values of the rows using backward fill. Also, the null values of volume variables are replaced with zero.



**Daily Resampled**

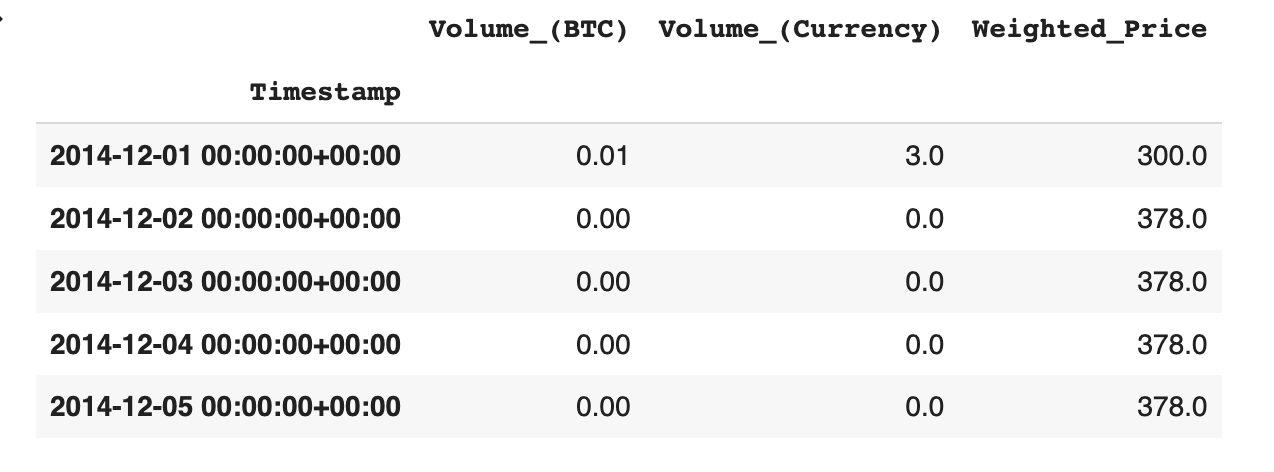
**Splitting of dataset**

Further, we split the data into two parts: Train dataset and Test dataset. In case of time series, the traditional approach of splitting the data into a ratio like 80:20 or 70:30 is not followed.

The time series problem, most of the data (historical) is used for training our model and recent data is used for testing the model. Therefore, a big portion of data is kept for training processing and the rest of data is used for testing process.

We have taken **3.5 years** for training dataset with period before **June 2018** out of the total span of almost 4 years. However, testing data involve records of **6 months after June 2018**.

The head of test data is shown as:



**Seasonal Decompose**

In this part, the seasonal trends of data are observed. The **pattern is observe**d for 1 year defining period from January 2015 to January 2016 which is a full year of data. This will clarify that how Bitcoin fluctuate over the different months in year.

**Chart, histogram

Description automatically generated**

The target variable: weighted price will help to check the variations of the price based on seasonal pattern which confirms that our dataset has weekly seasonality. As there are 4 peaks in second plot from January to February, this implies data has weekly seasonal patterns (m = 7).

The stability of data has been checked to generate the patterns. It is inspected that whether the data is stationary or not and the test called **Dickey Fuller test** is used in this context. This test is present in “statstools.adfuller” package.

For this, **null hypothesis claims** that the data is **not stationary** with **p-value>0.05** and **alternate hypothesis states** that it is **stationary with p<0.05.** The following picture specifies p-value as 0.731698 which is greater than 0.05. Therefore, our data is not stationary as we fail to reject null hypothesis.

The non- stationary behavior is visualized by peeks among different months of weighted seasonal graph. Sometimes, it goes up and in the next months, it drops down. The residuals follow similar fluctuating nature in the beginning and around the month of November.

**LSTM- Long Short-Term Memory model**

Now we are familiar with basic data analysis of time series, further we will contiue with modelling. Time series is basically processed with machine learning models. The most popular among those is LSTM- Long Short-Term Memory model.

It is an artificial neural network mostly processed in the area of artificial intelligence and deep learning. It has feedback connections which makes it capable to process sequences of data. It uses recurrent neural network. LSTM is basically used in connected handwriting recognization, speech recognization, machine translation, time series etc.

The terms neural network and recurrent neural network are defined as:

**Neural Network** is basically a inspired from biological neural networks that processes with layered structure of connected neurons. It is a compositions of many algorithms to process complex operations on data.

**Recuurent Neural Networks (RNN)** is a specific class of neural network to process temporal data. Basically the neuron have cell state memory. Here, the input is processed by this specific internal state called cell memory/state and the process is accomplished with the help of loops in neural network. The recurring module of hyperbolic tangent layers allow them to retain information for a short time.

**LSTM** is a special type of RNN which works for long duration.The structure of LSTM aims to facilitate a short term memory which can last thousands of timesteps and ultimately make it a long term memory.

Chart, box and whisker chart

Description automatically generated

The above picture defines the network used in LSTM. The four neural network layers are represented by yellow boxes, yellow circles receive input, blue circles are showing cell states and the green circles are representing operators.

The input gate accepts three dimensional input, and the cell memory gate terminals store the value of all feedback inputs. The forget gate is represent with different mathematical functions called activation functions. These keep the model activated in a long run and the output gate records output value.

**Reshaping Data**

The dataset is reshaped for the target variable. The reshaping is achieved to make the LSTM input layers three-dimensional. The input contains three dimensions as discussed below:

**Samples:** A group is compiled from one or more samples which are further combined to give a specific sequence.

**Time Steps:** It is defined as the time limit of reccurence events. It is one point of focus for each sample

**Features:** The feature is one variable of focus for specified time step.

In the following picture, the train and test copies are formed for LSTM processings. Here the reshaping is done by taking all observations across days versus the target variable (1282, 1) for train and (160, 1) for test data. Here the first index represents observational value and second index gives number of columns.

Before taking the final reshaped dataset, we normalized our variable or data for stable or better results. The X\_train and y\_train are derived to get relationship of one day with its previous days. In X\_train the index of days starts from 0 to one value less than maximum value whereas in y\_train, the index of day strats from 1 which goes to maximum value. This way the day recorded in X\_train is 0 and it is recorded with value 1 in y\_train. The modelling is done with X\_train variable and plotted with y\_train variable to see the visual patterns.

Now onwards, the reshaping is done based on above described method to make our variable three dimansional (1281, 1, 1). Here, the number of observations are reduced by 1 because we have taken 1 observation to define time step in second index and the last index represent number of variables.

**Graphical user interface, text, application

Description automatically generated**

The train and test dataset for target variable ‘weighted price’ is joined to give following plot or the following plot visualize data splitting. It specifies that the price was spiked for specific time around January 2018.

**Graphical user interface, application

Description automatically generated**

**LSTM model process**

In general, time series modelling, or the neural network is created by taking some hidden layers between input and output layer. We have used **four layers** as these optimized our result. We have used **tangent hyperbolic for activation** as the range of second order derivative for tanh is sufficient to keep the model working for a long span. **Recurrent activation is used to update or forget for data selection in** circuit to make RNN work for long time in Long Short-Term Memory model. **The return sequences** returned the hidden output state for input state in recurring iteration. **Dropout of 0.2** is regularization of 20%. This functions for input and recurrent connections are probabilistically excluded for activation and weight updates while network training in LSTM.

Text

Description automatically generated

After defining the layers, we have optimized with stochastic gradient descent (which optimize the model for efficient performance, or it reduces the coefficient of errors). We have used hyperparameter of this gradient descent that are called epochs. It basically controls counts of complete passes through training data.

**Table

Description automatically generated**

**Table

Description automatically generated**

The model is optimized with 38 epochs because of the callback function. Callback function ensure that if the validation loss start increasing from the best minimum value, it will optimize and stop at a certain mentioned patience epoch. (In our case 20). The final validated model will be the best minimum validation loss model.

**Validation Loss**

Further a plot is drawn to see that how well the model fits the training data and it is observed that with each epoch, the validation loss is reducing till epoch 18. After that the validation loss started increasing again and was stopped after 20 more epochs.

**Chart, histogram

Description automatically generated**

**Predictions**

Predictions are made to validate our LSTM model because usually this is best model for stationary data and our time series data is not stationary. To find the prediction trend, we have modified training and testing dataset. It is shown that initially, the testing dataset is reshaped and then prediction\_LSTM is defined with scaled transformation on testing dataset. Similar process is used to define prediction\_lstm\_train. These are now compared with initial training and testing dataset with three variables. Following is the initial and prediction modified shapes of testing and training dataset and prediction.

Graphical user interface, text, application, email

Description automatically generated

**Visualization of predictions**

The following plots are drawn to validate the predictions. The following graph visualized **predictions made on test dataset.** The plot has been drawn for almost six months and it shows that the original weighted price and predicted price almost overlap. This validates the predictions made on test dataset are almost valid.

**Chart, line chart, histogram

Description automatically generated**

**Test dataset prediction (from June 2018 to November 2018)**

The next plot shown prediction made on test dataset for a small time interval. This prediction is made for initial point of test dataset as this is the point of split for test and train dataset. This plot again fitted well our target variable with its predicted value.

**Chart, line chart

Description automatically generated**

**Test dataset prediction (from 9 June 2018 to 23 June 2018)**

This plot shows **predictions made on train dataset** and it is observed that from January 2015 to December 2017, the prediction fits target variable well. However, for the peak trend, prediction have sufficient gap between original and predicted value which is shown from January 2018 onwards. After the peak time, the predictions again fit well. Moreover it is a best practice to predict for small recent time burst.

**Chart, line chart

Description automatically generated**

**Train dataset predictions**

**Evaluation**

The evaluations of predicted values have been made by taking **Mean Squared Error Values.**

**MSE (Train Prediction) = 696.85**

**MSE (Test Prediction) = 94.93**

By comparing these values, we can say that error is more in train prediction which conclude that this model does not capture all random variations on train dataset whereas test predictions are comparably better with low values of MSE.

**Graphical user interface, text, application

Description automatically generated**

**Evaluation 2**

**MSE (Test Prediction: for 2 recent weeks) = 86.107**

**Test Mean = 6703.86**

By comparing these values, we can say that our prediction for 2 week is quite good with just 1.28% deviation from real test mean.

**Summary**

1. First of all, we started researching on RNN and LSTM and learned about its working. We explored various Kaggle reports on how LSTM work and forecast in general.
2. We also used the same backward filling method for missing data of weighted price and 0 for volume.
3. We then used the same preprocessed bitcoin data we used before for EDA and SARIMAX model. We considered the resampling of hourly data to daily data like before.
4. We split the data into 2 parts: 3.5 years for training and 6 months for testing.
5. We used “statstools.adfuller” package for seasonal decompose and analysed the trend and seasonality. In our case, it was a weekly seasonal data.
6. We then started preprocessing our data based on LSTM input requirement. We converted our 2D data frame into 3D shape. (Number of rows, timesteps, number of attributes)
7. Then we designed LSTM by defining 5 LSTM layers. The first is the input layer with 50 units and activation function of “tanh” and “sigmoid”. We then added three more hidden layers and one output layer. We also used gradient descent for our optimizer and mean squared error for evaluation. We split the train dataset into 90:10 train-validation set.
8. By using callback function, our model reached its best valuation at 38th epoch.
9. We predicted our LSTM model for recent 2 weeks and found that prediction is only 1.28% deviated from real test mean. We also calculated the RMSE of train and test period.

**References**

1. Mittal, A. (2019). *Understanding RNN and LSTM*. [online] Medium. Available at: <https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e>.
2. Wikipedia Contributors (2018). *Long short-term memory*. [online] Wikipedia. Available at: https://en.wikipedia.org/wiki/Long\_short-term\_memory.
3. www.pluralsight.com. (n.d.). *Introduction to LSTM Units in RNN | Pluralsight*. [online] Available at: https://www.pluralsight.com/guides/introduction-to-lstm-units-in-rnn.
4. baeldung (2022). *Training and Validation Loss in Deep Learning | Baeldung on Computer Science*. [online] www.baeldung.com. Available at: https://www.baeldung.com/cs/training-validation-loss-deep-learning.

‌