Project Title

"CRYPTOCURRENCY PAIR FORECASTING & ANALYSIS BY USING TIME SERIES MODELS & ML ALGORITHMS WITH INTRODUCTION TO BLOCKCHAIN TECHNOLOGY"

An Abstract of The

"CRYPTOCURRENCY PAIR FORECASTING & ANALYSIS BY USING TIME SERIES MODELS & ML ALGORITHMS WITH INTRODUCTION TO BLOCKCHAIN TECHNOLOGY"

Submitted to the **Statistics Department**

as partial fulfillment of **P-305**

of the requirements for the **MSc. Statistics PG Degree**

in ***Progressive Education Society's Modern College*** of Arts, Science and Commerce (Autonomous) Shivajinagar, Pune – 411005,

Affiliated by **Savitribai Phule Pune University (SPPU)**

The goal of this study is to predict prices for Cryptocurrencies using Time series analysis and machine learning techniques. The purpose of this project is to take a sneak peek into the future by **forecasting the next 30 days' average daily Realized Volatility (RV) of ETH-BTC** using 2 different approaches - the traditional econometric approach to volatility prediction of financial time series **GARCH** and state-of-the-art **LSTM Neural Networks**.

Quantitative research methodology was used in this study and the The dataset Consist the historical data values of any any crypto-pair such as Open/Close/High/Low prices of any interval such as 15-minutes, Hourly, 1-day interval weekly, monthly. Dataset were obtained using the Binance API .

CERTIFICATE

This is to certify that

Ms. Aishwarya Sachine Mehendale

Mr. Abhijeet Balasaheb Talole

Ms. Akshata Gangadhar Musale

The students of MSc. Statistics department of Modern College of Arts, Science and Commerce (Autonomous), Shivajinagar, Pune successfully completed the project named

"CRYPTOCURRENCY PAIR FORECASTING & ANALYSIS BY USING TIME SERIES MODELS & ML ALGORITHMS WITH INTRODUCTION TO BLOCKCHAIN TECHNOLOGY"

Guided by Assistant **Prof. Sarika Khirid** during academic year **2021-2022** as per the guidelines issued by Department of Statistics.

Mrs. Sarika Khirid
Project guide

Dr. P.G.Dixit
(HOD, Department of Statistics)

INDEX

ACKNOWLEDGEMENT

In developing and completing our project, **"CRYPTOCURRENCY PAIR FORECASTING & ANALYSIS BY USING TIME SERIES MODELS & ML ALGORITHMS WITH INTRODUCTION TO BLOCKCHAIN TECHNOLOGY"**, we take the opportunity to thank all those who were directly or indirectly involved in this project.

We would like to express gratitude to all the teachers of department of statistics, non-teaching staff of Statistics department of Modern College, Pune for their support and kind co-operation. Specially we would also like to acknowledge the guidance provided by our project guide Mrs. Sarika Khirid and the teaching staff. Along with that we would like to thank our friends for their support and encouragement.

An accomplishment and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project. All that we have done was possible only due to such supervision and assistance and we would not forget to thank them.

INTRODUCTION

In recent times, the universe of virtual market trading has encountered impressive expansion. Online trading platforms have changed the manner by which speculators inquire about and execute orders. The idea of algorithmic trading presents the next phase of this development, and one that is as of now affecting entities of high volume and liquidity.

In this virtual world, Cryptocurrencies can no longer be ignored. “Cryptocurrencies are digital, straightforward and simple to use in comparison with the traditional currencies”, The individual investor, the big cooperation and even governments (who seek ways to control their influence in their financial markets) are interested in them.

Bitcoin is one of the oldest and biggest cryptocurrencies being traded as of now, in terms of volume being traded it is so big that even now, with the advent of thousands of new cryptocurrencies (Altcoins), Bitcoin has market share of more than 45% as compared to other cryptocurrencies being followed by Ethereum at 8.75%. This says a lot about why Bitcoin and Etherium might be really interesting and important stock to predict. Also, Their prices fluctuate heavily. Over the past two

Cryptocurrency is one of the most volatile markets today and has gained a lot of attention from investors across the globe. Cryptocurrency, being a novel technique foe transaction system, has led to a lot of confusion among the investors.

In India Government imposed 30% flat tax on cryptocurrency and include it in the high risk financial/digital asset instead of completely banning it in India and also setup a government body CBDC (Central Bank Digital Currency) regulated by RBI, to regulate Cryptocurrencies. As a Indian crypto treader considering cryptocurrency related taxation structure we need to reduced loses and take better decisions while booking profit from Cypto-treading or investing. This is one of the most important reasons which attracted us to make statistical analysis of Cryptocurrencies.

KEY WORDS

Cryptocurrency:

What is cryptocurrency and how it works?

A cryptocurrency is **an encrypted data string that denotes a unit of currency**. It is monitored and organized by a peer-to-peer network called a blockchain,

Cryptocurrency is **a digital payment system that** doesn't rely on banks to verify transactions. ... When you transfer cryptocurrency funds, the transactions are recorded in a public ledger. Cryptocurrency is stored in digital wallets. Cryptocurrency received its name because it uses encryption to verify transactions.

Blockchain:

Blockchain is **a system of recording information in a way that makes it difficult or impossible to change, hack, or cheat the system**. ... Each block in the chain contains a number of transactions, and every time a new transaction occurs on the blockchain, a record of that transaction is added to every participant's ledger.

Bitcoin:

Bitcoin is **a digital currency which operates free of any central control or the oversight of banks or governments**. Instead it relies on peer-to-peer software and cryptography. A public ledger records all bitcoin transactions and copies are held on servers around the world.

Since Bitcoin's first appearance in 2009, it has changed the world's financial landscape substantially. The decentralized cryptocurrency has established itself as an asset class recognized by many asset managers, large investment banks and hedge funds. As the speed of mainstream adoption continues to soar, it is also leading investors to explore new ventures, such as crypto options and futures.

Bitcoin has been historically known to be more volatile than regulated stocks and commodities. Its most recent surge in late December 2020, early January 2021 has brought about a lot of questions and uncertainties about the future financial landscape. At the point of writing this report Bitcoin is traded at slightly below USD 39,000, which is no small feat considering it entered 2020 at around USD 7,200.

Altcoin: The term altcoin refers to **cryptocurrencies other than Bitcoin** (and sometimes also other than Ether). Such coins distinguish themselves from Bitcoin by extending their capabilities and plugging their shortcomings.

Etherium:

Ethereum is a **decentralized blockchain platform** that establishes a peer-to-peer network that securely executes and verifies application code, called smart contracts. ... A sender must sign transactions and spend Ether, Ethereum's native cryptocurrency, as a cost of processing transactions on the network.

Stablecoin:

Stablecoins aim to **eliminate price volatility**. A stablecoin, if successful, would be effective as a store of value and a medium of exchange, just like a fiat currency, while retaining the qualities of a cryptocurrency.

API: API is the **acronym for Application Programming Interface**, which is a software intermediary that allows two applications to talk to each other. Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you're using an API.

Binance API:

The Binance API is **a method that allows you to connect to the Binance servers via Python or several other programming languages**. With it, you can automate your trading. More specifically, Binance has a RESTful API that uses HTTP requests to send and receive data.

Ticker:

Tickers are a financial instrument used to track the price value of given asset or basket of assets

MOTIVATION

In India Government imposed 30% flat tax on cryptocurrency and include it in the high risk financial/digital asset instead of completely banning it in India and also setup a government body CBDC (Central Bank Digital Currency) regulated by RBI, to regulate Cryptocurrencies. As a Indian crypto treader considering cryptocurrency related taxation structure we need to reduced loses and take better decisions while booking profit from Cypto-treading or investing. This is one of the most important reasons which attracted us to make statistical analysis of Cryptocurrencies.

Volatility attempts to measure magnitude of price movements that a financial instrument experiences over a certain period of time. The more dramatic the price swings are in that instrument, the higher the level of volatility, and vice versa.

Volatility is generally accepted as the best measure of market risk and volatility forecasting is used in many different applications across the industry. **Realized Volatility Forecasting** models are typically utilized in risk management, market making, portfolio optimization, and option trading. Specifically, according to Sinclair (2020), a number of trading strategies evolve around identifying situations where this volatility mismatch occurs:

in which Vega is the measurement of an option's price sensitivity to changes in the volatility of the underlying asset, and σ is volatility. As Implied Volatility could be derived from Option Prices using models such as the Black Scholes Model, forecasting Realized Volatility would give us the key to the second part of the equation.

Although the forecasting and modeling of volatility has been the focus of many empirical studies and theoretical investigations in academia, forecasting volatility accurately remains a crucial challenge for scholars. On top of that, since crypto option trading is relatively new, there has not been as much research done on this Crypto volatility forecasting. In addition, cryptocurrencies carry certain nuances that differ themselves from traditional regulated stocks and commodities, which would also need to be accounted for.

OBJECTIVES:

- Finding best treading strategy for Cryptocurrency Treading.
- Find best Crypto-pair for Binance Spot treading
- To Find appropriate model for forecasting of cryptocurrency Volatility.
- The goal of this study is to find the best Time series model and predict prices for Cryptocurrencies using Time series analysis and machine learning techniques.
- The purpose of this project is to take a sneak peek into the future by **forecasting the next 7 days' average daily Realized Volatility (RV) of ETH-BTC** using 2 different approaches - the traditional econometric approach to volatility prediction of financial time series **GARCH** and state-of-the-art **LSTM Neural Networks**.

Additionally, there will be two secondary objectives:

- Spread awareness about new Fintech term Cryptocurrency
- Introduce the blockchain technology

DATASET:

The historical dataset of any cryptocurrency Open/Close/High/Low prices were obtained using the Binance API . we can generate this API keys only if we have investment in blockchain or we mine some cryptocurrencies, very easy to set up, but yet still contains a wide range of data and offerings.

I will be downloading ANY CRYPTO's prices using ticker ANY CRYPTO at any interval such as 15-minutes, Hourly, 1-day interval weekly, monthly.
API | Binance Support: https://www.binance.com/en-IN/support/faq/c-6

Binance API Documentation: https://binancedocs.github.io/apidocs/spot/en/#change-log

For the base currency: Bitcoin We use the data for the duration 1 JAN 2017 to Current day

repeat same for the common Cryptocurrencies

according to NATHAN REIFF in his article "The 10 Most Important Cryptocurrencies Other Than Bitcoin", Updated Jan 8, 2020(source: https://www.investopedia.com/tech/most-important-cryptocurrencies-other-than-bitcoin/) the top three are . Ethereum (ETH), . Ripple (XRP) and . Litecoin (LTC). for the sake of a more robust overview, we will be considering the top ten Cryptocurrencies as at 8 October 2019 noted by Yahoo Finance : Source: https://finance.yahoo.com/news/top-10-cryptocurrencies-market-capitalisation-160046487.html

1. Bitcoin (BTC)

2. Ethereum (ETH)

3. XRP (XRP)

4. Bitcoin Cash (BCH)

5. Tether (USDT)

6. Litecoin (LTC)

7. EOS (EOS)

8. Binance Coin (BNB)

9. Bitcoin SV (BSV) the BSV has no unique ticker in (and claims to be the original version of what Bitcoin was meant to be ; Source: https://bitcoinsv.com/en/learn)

10. Stellar (XLM)

these Eight (save Bitcoin SV (BSV)) would be examined to get the best pair for our base currency(Bitcoin)

STATISTICAL TOOLS USED FOR DATA ANALYSIS

Normalization

Exploratory Data Analysis

Time Series Analysis:

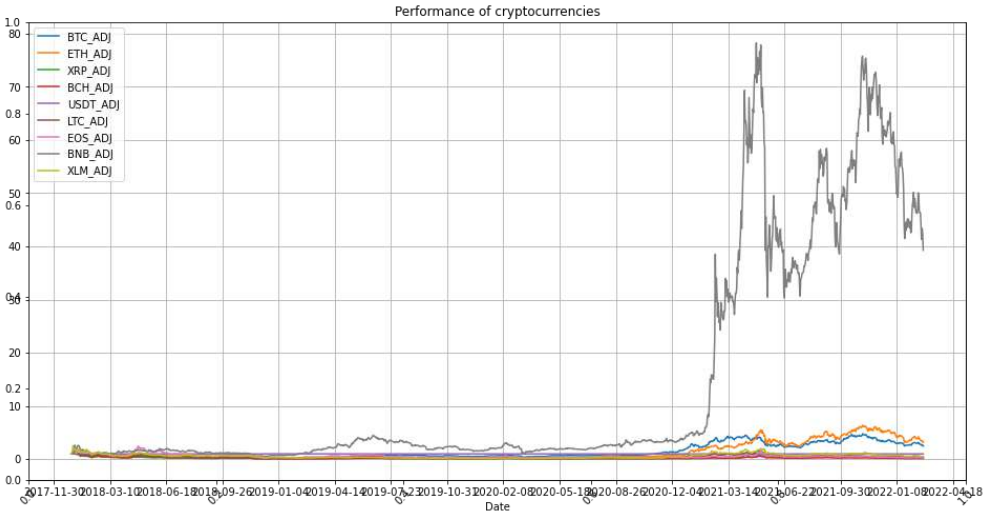
- Auto ARIMA
- Baseline Models
  - Mean Baseline Model
  - Random Walk Naïve Forecasting
- GARCH
  - GJR-GARCH
  - TARCH
    - Simulation
    - Bootstrap

Machine Learning

1. Neural Network
   - LSTM

SELECTION OF BEST CRYPTO PAIR

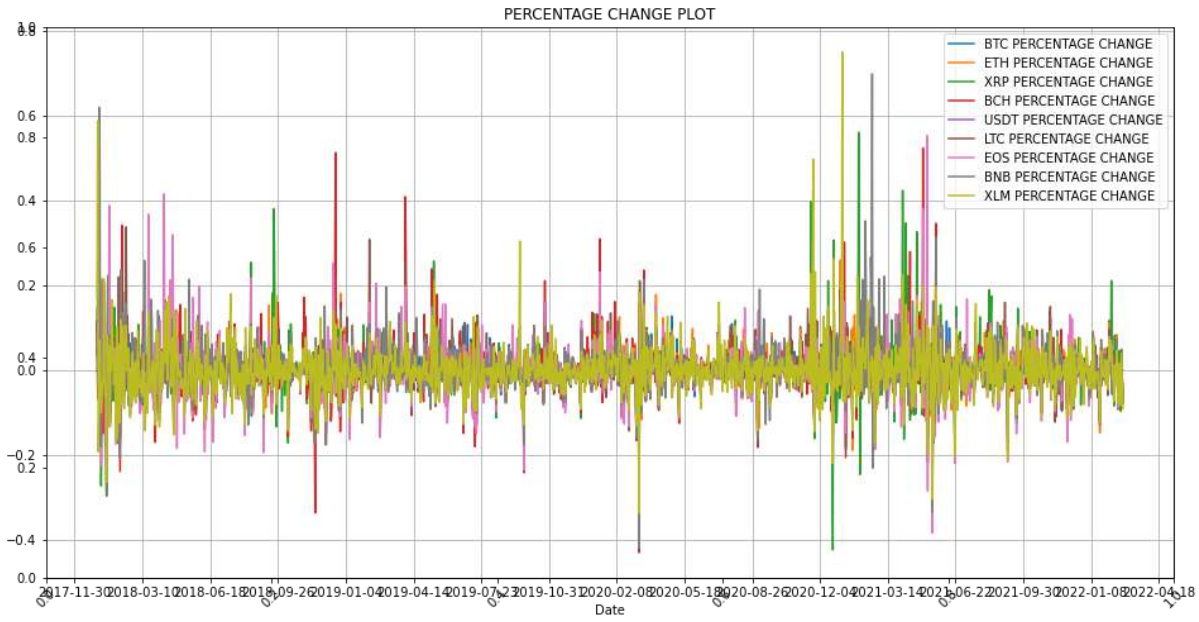
Performance of Cryptocurrencies:



Conclusion :

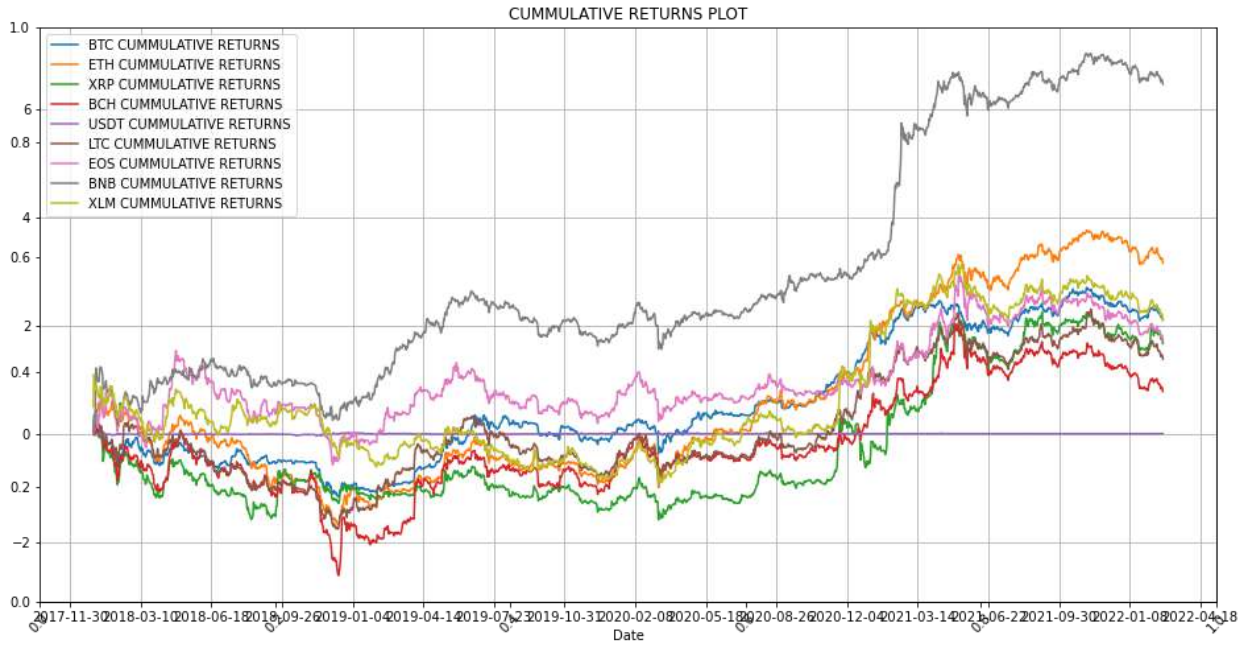
we observed that some of the currencies tend to out perform others within the reviewed perio d Then we then introduce the percentage change to our data so as to scale them

Percentage Change Plot:



Conclusion : Percentage change of all Cryptos are overlapping and randomness is Stationary.

Cumulative Returns Plot:



Two major criteria that would determine the desired pair are the correlation value and Cointegration. The pair with the highest correlation, is the desired pair. High positive correlation indicates that the returns of both currencies are trending in same direction. Additionally, the cointegration of the same pair should be significantly low. A value lower than 0.05 is desired

First we attempted to get the p-value of all possible pair in our selected (9- Crypto currency Universe)

Obtaining the P-value for every possible pair within our selected (9- Crypto currency Universe):

```
BTC_RET and ETH_RET: p-value = 9.580499756418874e-24
BTC_RET and XRP_RET: p-value = 1.0379453946237762e-12
BTC_RET and BCH_RET: p-value = 0.0
BTC_RET and USDT_RET: p-value = 0.0
BTC_RET and LTC_RET: p-value = 0.0
BTC_RET and EOS_RET: p-value = 2.363992104909598e-24
BTC_RET and BNB_RET: p-value = 1.4604139207330056e-24
```

```
BTC_RET and XLM_RET: p-value = 0.0
ETH_RET and BTC_RET: p-value = 6.317121790275272e-19
ETH_RET and XRP_RET: p-value = 3.862348570721113e-15
ETH_RET and BCH_RET: p-value = 0.0
ETH_RET and USDT_RET: p-value = 6.959068059089577e-20
ETH_RET and LTC_RET: p-value = 0.0
ETH_RET and EOS_RET: p-value = 1.0586422437213404e-25
ETH_RET and BNB_RET: p-value = 1.3701831083563046e-16
ETH_RET and XLM_RET: p-value = 1.4092472333155725e-25
XRP_RET and BTC_RET: p-value = 7.363929287366418e-28
XRP_RET and ETH_RET: p-value = 4.0349475618371465e-09
XRP_RET and BCH_RET: p-value = 2.781644897574791e-09
XRP_RET and USDT_RET: p-value = 0.0
XRP_RET and LTC_RET: p-value = 4.984689744363252e-28
XRP_RET and EOS_RET: p-value = 2.638846202621077e-10
XRP_RET and BNB_RET: p-value = 9.977447221798753e-28
XRP_RET and XLM_RET: p-value = 6.374734802497867e-29
BCH_RET and BTC_RET: p-value = 0.0
BCH_RET and ETH_RET: p-value = 0.0
BCH_RET and XRP_RET: p-value = 0.0
BCH_RET and USDT_RET: p-value = 4.412834541311187e-26
BCH_RET and LTC_RET: p-value = 0.0
BCH_RET and EOS_RET: p-value = 0.0
BCH_RET and BNB_RET: p-value = 0.0
BCH_RET and XLM_RET: p-value = 0.0
USDT_RET and BTC_RET: p-value = 1.328235952802207e-22
USDT_RET and ETH_RET: p-value = 2.6142780156456317e-22
USDT_RET and XRP_RET: p-value = 3.4209453126419756e-22
USDT_RET and BCH_RET: p-value = 1.160320989454982e-22
USDT_RET and LTC_RET: p-value = 1.5788570045733988e-22
USDT_RET and EOS_RET: p-value = 1.431743150071143e-22
USDT_RET and BNB_RET: p-value = 1.2099201277491802e-22
USDT_RET and XLM_RET: p-value = 1.3131029750743377e-22
LTC_RET and BTC_RET: p-value = 0.0
LTC_RET and ETH_RET: p-value = 0.0
LTC_RET and XRP_RET: p-value = 4.1743983085475546e-20
LTC_RET and BCH_RET: p-value = 0.0
LTC_RET and USDT_RET: p-value = 1.101645528394078e-19
LTC_RET and EOS_RET: p-value = 1.8947809066467715e-11
LTC_RET and BNB_RET: p-value = 0.0
LTC_RET and XLM_RET: p-value = 0.0
EOS_RET and BTC_RET: p-value = 3.5851448776334765e-10
EOS_RET and ETH_RET: p-value = 5.7487270119543615e-21
EOS_RET and XRP_RET: p-value = 2.0720999782124922e-24
EOS_RET and BCH_RET: p-value = 2.8313952877158206e-23
EOS_RET and USDT_RET: p-value = 1.0373939850750631e-24
EOS_RET and LTC_RET: p-value = 6.459009527457782e-13
EOS_RET and BNB_RET: p-value = 3.1785746267886466e-18
EOS_RET and XLM_RET: p-value = 3.0120943697847417e-21
BNB_RET and BTC_RET: p-value = 2.9560697286257155e-20
BNB_RET and ETH_RET: p-value = 3.796215994277962e-11
BNB_RET and XRP_RET: p-value = 9.667299533122745e-11
BNB_RET and BCH_RET: p-value = 1.397774396384114e-10
BNB_RET and USDT_RET: p-value = 7.090537834096173e-16
```

```
BNB_RET and LTC_RET: p-value = 3.204440374791345e-09
BNB_RET and EOS_RET: p-value = 6.755267722430436e-15
BNB_RET and XLM_RET: p-value = 7.640061787437978e-16
XLM_RET and BTC_RET: p-value = 0.0
XLM_RET and ETH_RET: p-value = 2.9830854265107574e-17
XLM_RET and XRP_RET: p-value = 0.0
XLM_RET and BCH_RET: p-value = 0.0
XLM_RET and USDT_RET: p-value = 0.0
XLM_RET and LTC_RET: p-value = 0.0
XLM_RET and EOS_RET: p-value = 7.290954038229479e-24
XLM_RET and BNB_RET: p-value = 0.0
```

Conclusion:

From the results obtained we saw some very low Cointegration for our Desired Bitcoin Pair.

BTC_RET and ETH_RET: p-value = 0.0

BTC_RET and XRP_RET: p-value = 2.470976042943848e-29

BTC_RET and BCH_RET: p-value = 1.9958557257613114e-24

BTC_RET and USDT_RET: p-value = 1.8460483551336274e-23

BTC_RET and LTC_RET: p-value = 7.855462705098844e-28

BTC_RET and EOS_RET: p-value = 1.0541411510509593e-22

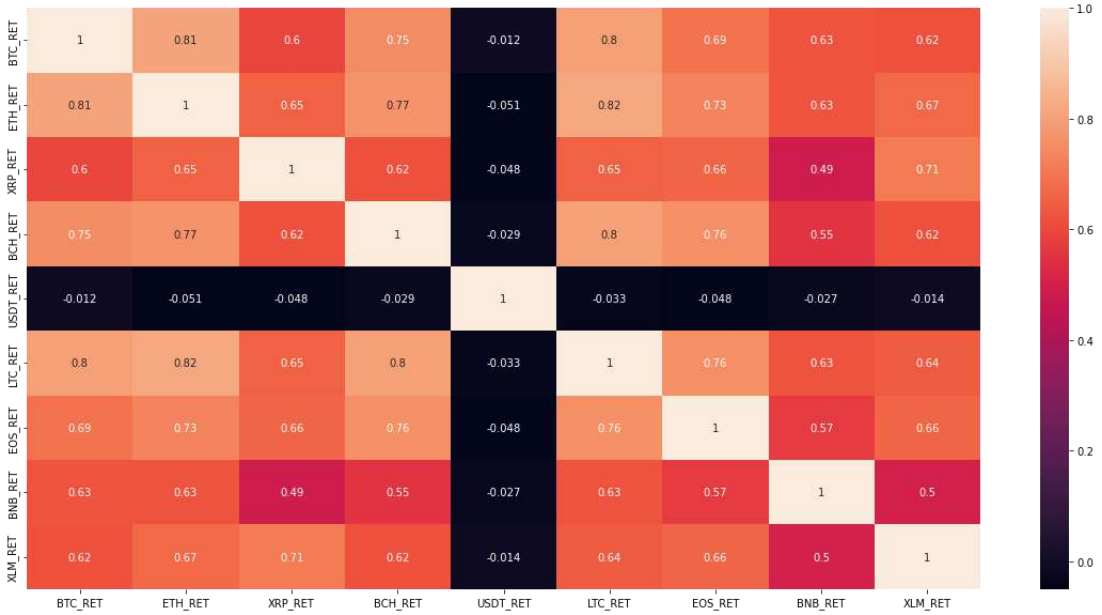
BTC_RET and BNB_RET: p-value = 0.0

BTC/ETH_RET Pair and the BTC/BNB Pair tend to give the lowest P-Values, however since the desired P-value is anything less than 0.05, all of the pairs would be considered to have passed the Cointegration test. Hence we decided to do the correlation test.

Correlation Metrix:

	BTC_RET	ETH_RET	XRP_RET	BCH_RET	USDT_RET	LTC_RET	EOS_RET	BNB_RET	XLM_RET
BTC_RET	1.000000	0.807768	0.595593	0.751616	-0.012364	0.798147	0.690993	0.626332	0.619044
ETH_RET	0.807768	1.000000	0.652851	0.767004	-0.050553	0.820644	0.731785	0.626451	0.668353
XRP_RET	0.595593	0.652851	1.000000	0.619114	-0.048362	0.653541	0.660953	0.490901	0.708861
BCH_RET	0.751616	0.767004	0.619114	1.000000	-0.029345	0.800901	0.760225	0.554579	0.621882
USDT_RET	-0.012364	-0.050553	-0.048362	-0.029345	1.000000	-0.033398	-0.047561	-0.027348	-0.014057
LTC_RET	0.798147	0.820644	0.653541	0.800901	-0.033398	1.000000	0.758343	0.625480	0.639208
EOS_RET	0.690993	0.731785	0.660953	0.760225	-0.047561	0.758343	1.000000	0.570990	0.661201
BNB_RET	0.626332	0.626451	0.490901	0.554579	-0.027348	0.625480	0.570990	1.000000	0.503437
XLM_RET	0.619044	0.668353	0.708861	0.621882	-0.014057	0.639208	0.661201	0.503437	1.000000

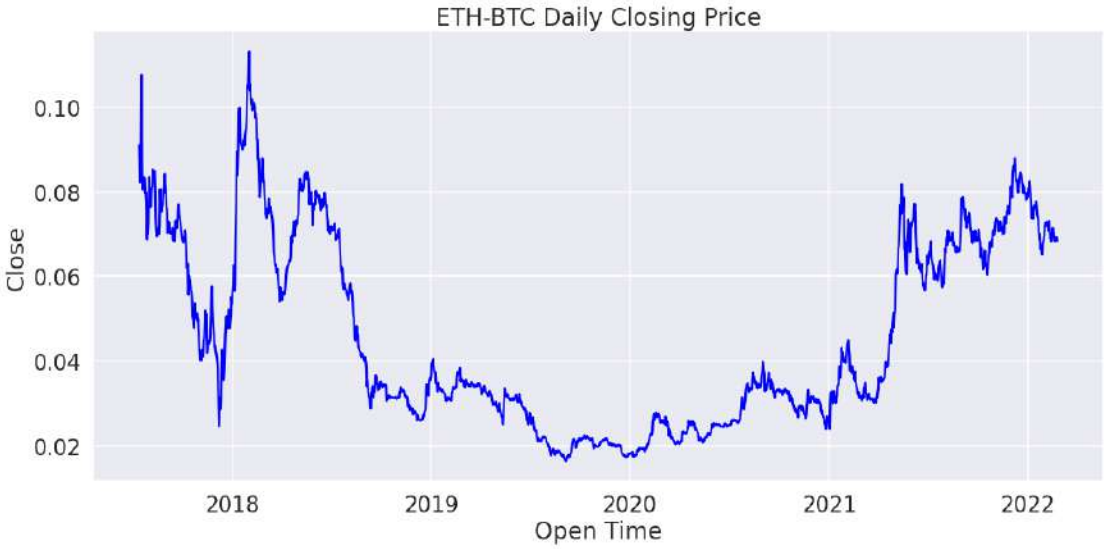
Correlation matrix Heatmap:



From the analysis and the values gotten we can see that although pairs all seen to have very low Cointegration value, but we observed that the Ethereum (ETH) had the best correlation value (81.2%). Thus we will be proceeding with the Bitcoin/Ethereum pair for the next two phases; Price forecasting and developing trade strategy

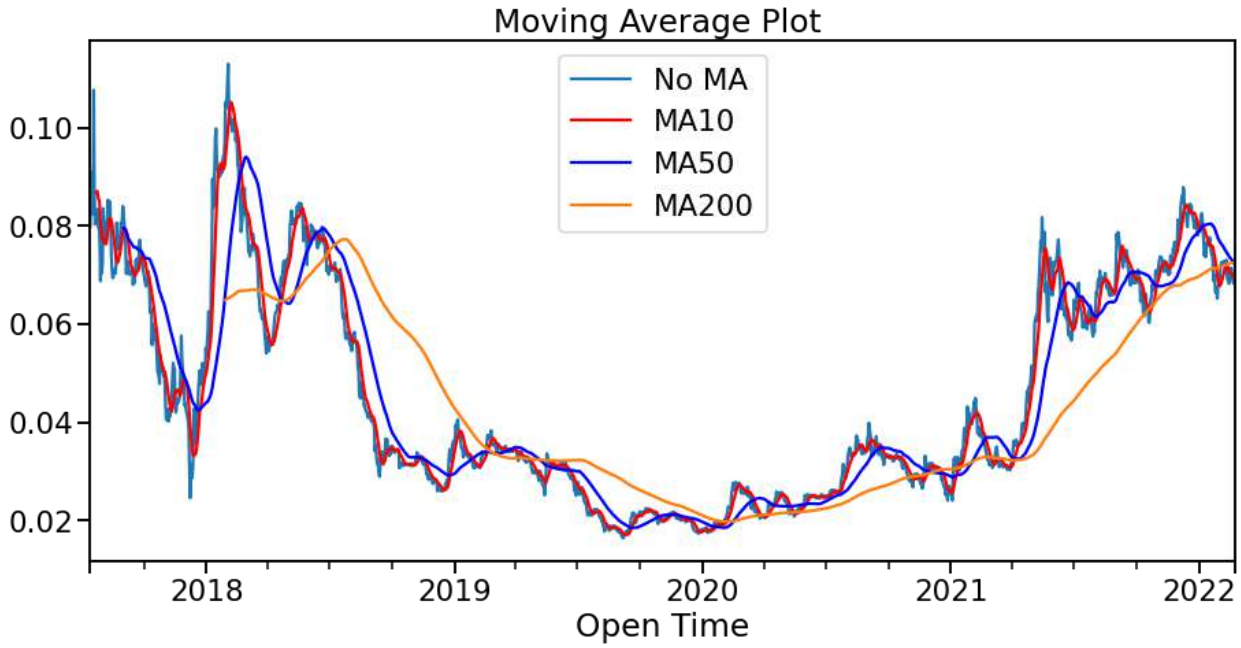
CLOSING PRICE ANALYSIS AND FORECASTING

Closing Price Plot of ETH-BTC:



Conclusion : From graph we can conclude that there is no Trend & Seasonality

Moving Average Plot:

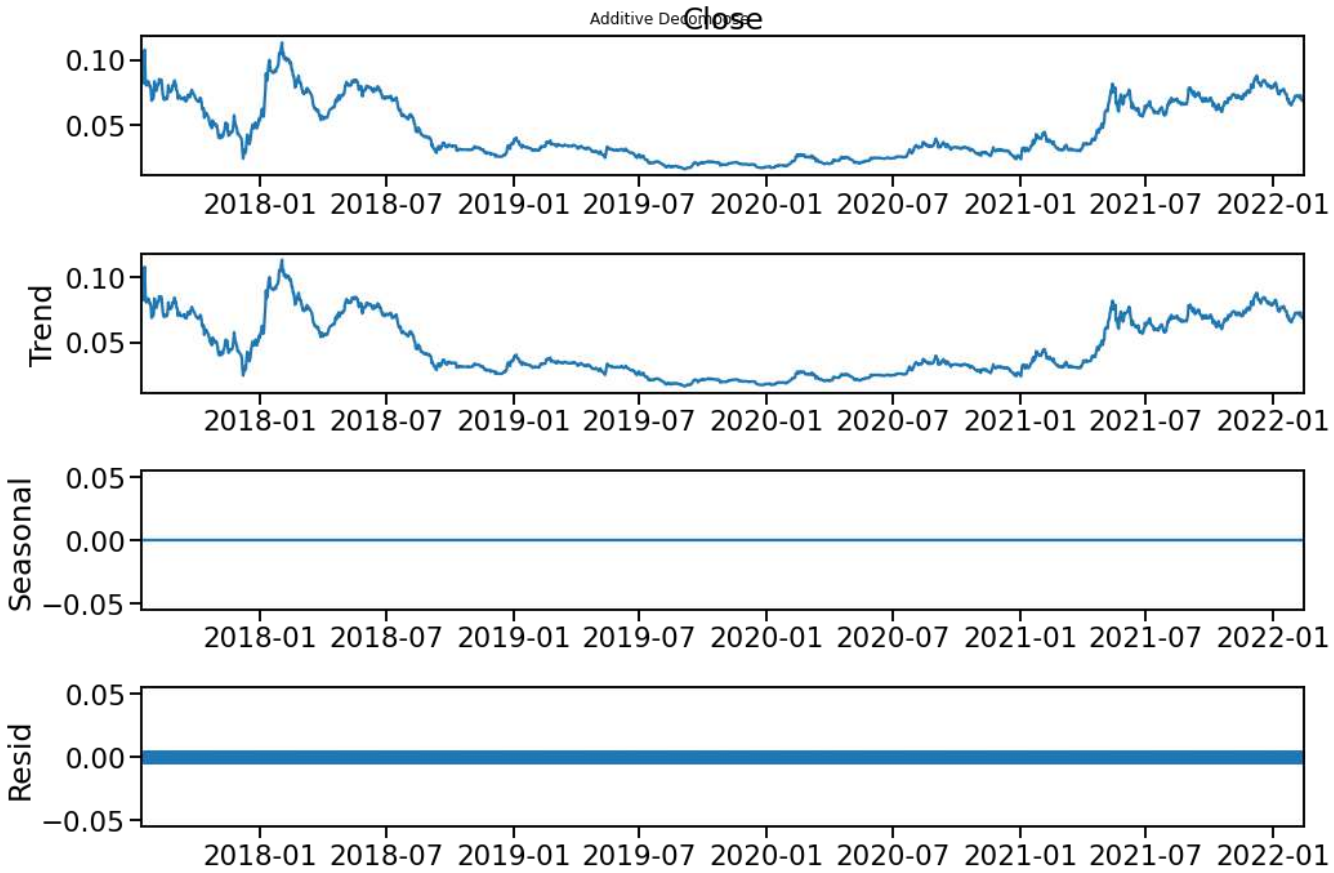


Conclusion: when MA is too large it not capable to capturing some spike and dips and when MA is small its again there is again noisy observation problem so there is compromise and we need to looking for some other methods like

AR,
ARMA,
ARIMA,
ACF,
EWMA

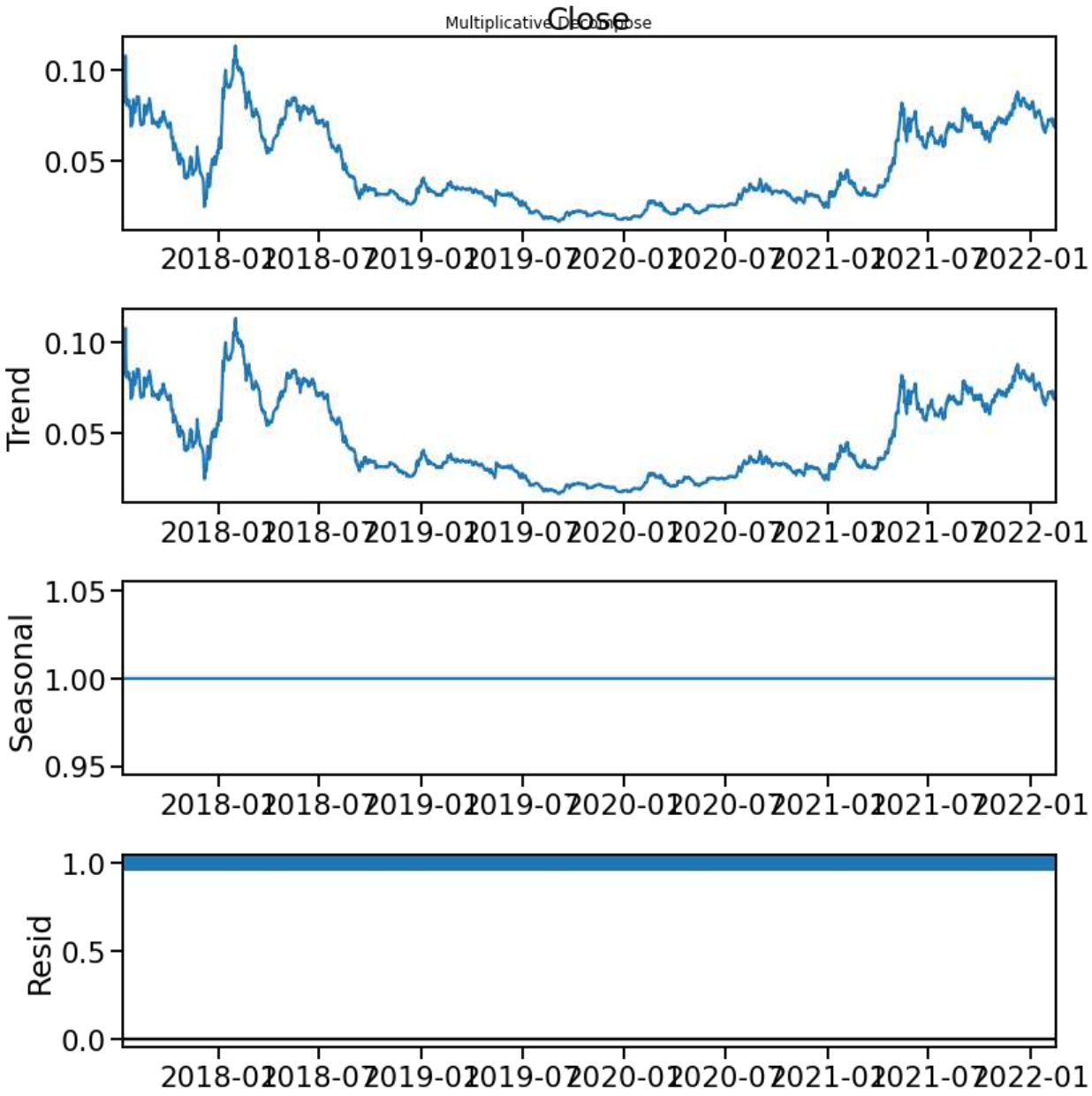
Decomposition of Close Prices:

Additive Decompose



Conclusion : There is may be Trend is present but Seasonality is not present.

Multiplicative Decompose



Conclusion : There may be Trend is present and Seasonality is not present.

Stationarity Checking by Augmented Dickey-Fuller (ADF) test:

Stationarity: A time series is said to be stationary if it does not show seasonal or trend effects on an aggregate level • This can be observed by analyzing the seasonal decomposition of the time series attribute to be forecast • So, to check if data is stationary or not, we used the Augmented Dickey-Fuller (ADF) test. It is the most popular statistical method to find if the series is stationary or not. It is also called as Unit Root Test.

ADF Statistic: -2.6280922757920044
p-value: 0.08732904025578841
Critical Values:
1%: -3.4343
5%: -2.8633
10%: -2.5677

Using a significant level alpha of 0.05, p-value for Close Prices are significantly larger than alpha, which means there's enough evidence to accept the Null Hypothesis.

H0: Process is not stationary

V/s

H1: Process is stationary

Here **p-value is too Large than alpha** Hence **accept H0** and conclude that **process is not stationary**.

Neural Network:

While GARCH remains the gold standard for volatility prediction within traditional financial institutions, there has been an increasing number of professionals and researchers turning to Machine Learning, especially Neural Networks, to gain insights into the financial markets in recent years.

Traders' theory of the market being inherently efficient (Efficient Market Hypothesis or EHM) states that share prices reflects all information and consistently outperforming the overall market is impossible. *The more efficient a market is, the more random and unpredictable the returns will be, and thus a perfectly efficient market will be completely unpredictable.*

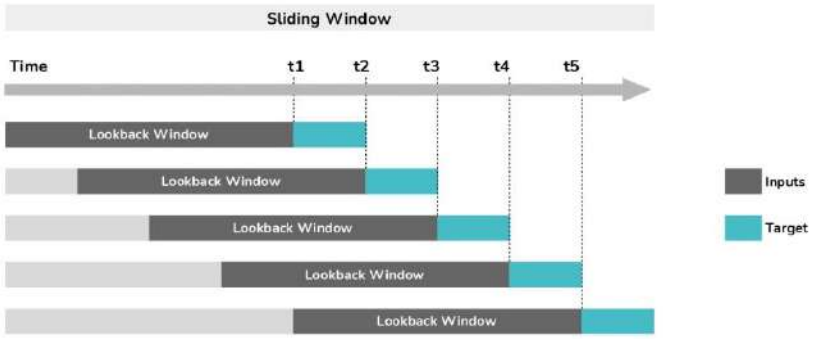
There are other arguments against EMH, and ones of the most prominent one is based on ***Behavioral Finance:*** compared to the human history of 200,000 years, the market has not

been around for that long. For example, equity options have only been traded in liquid, transparent market since the CBOE opened in 1973; and the average lifetime of an S&P500 company is approx. 20 years. It means that some psychological tendencies of human beings have 200,000 years of evidence behind them, and that a lot of the movements of the markets that were driven by participants' behaviors will likely repeat itself at a later point. Therefore the market system cannot be totally random, it must have some patterns. Those patterns are extremely difficult to exploit due to the multitude of factors that interact and drive the market.

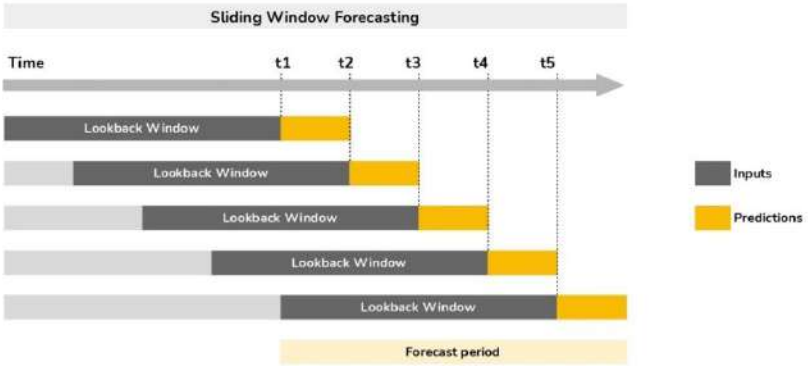
"It'd be interesting to see how Neural Networks perform compared to the traditional GARCH models."

For Neural Networks, instead of feeding all available datapoints into the network at once, I will use a sliding lookback window to extract uniform input arrays and target outputs. I will just be using a stride value of 1 to make sure I get all the inputs and output combinations available since 1650 datapoints is not really a whole lot to spare.

This is a demonstration of sliding window:



To generate predictions using Neural Networks, I will use the same sliding window concept again:



This means that to generate future predictions for a time step t, I will need to traverse back in time and collect the last n_past datapoints (from time step t - n_past + 1 to t inclusively). That's why the shape of the inputs into these Neural Networks need to be [batch_size, n_past, 1].

Neural Network Baseline Metrics - Fully Connected Network

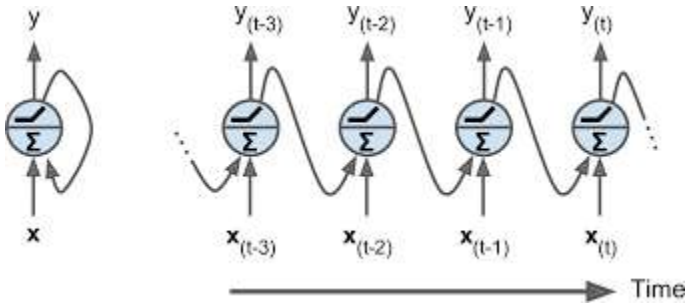
This is essentially *Linear Regression*.

(Training directly using scaled volatility produces inf RMSPE, so I'll stick with the original volatility, and scale the forecasts later).

I will start with using a lookback window n_past of 30, and then gradually adjust that value based on the outcomes of my models.

FIT MODEL TO TRAINING DATASET:

Univariate Long Short-Term Memory (LSTM): There's a class of Neural Networks called Recurrent Neural Networks (RNN) that can "predict the future". *RNN works well with time series data, such as stock prices, and can even process sentences, documents, audio samples as inputs.* Another application of RNN is in autonomic driving systems, where they're used to anticipate vehicle trajectories and help avoid accidents.



At each time step t, the recurrent neuron receives input xt as well as its own output from the previous time step yt−1. Since the output of a Recurrent Neuron is a function of all the inputs from the previous time steps, it has a form of *memory* and able to preserve some information through time.

However, due to some transformations that the data goes through when traversing down RNN, some information is lost at each time step, and for a long sequence, the RNN's state contains virtually no information from the first inputs.

Long Short-Term Memory (LSTM) was proposed in 1997 by Sepp Hochreiter and Jurgen Schmidhuber. The key part of LSTM is that the network can learn what's important and needs to be stored in the long-term state, and what can be ignored. LSTM looks very similar to a regular RNN cell, but its state is split into 2 vectors:

- *ht - for short term state*
- *ct - for long term state*

"LSTM is a black box, and can be used like a basic Recurrent Neural Network (RNN) cell. However, it tends to perform much better, helps training converge faster, and also detect long-term dependencies in the data."

LSTM uses an optimized implementation when running on a GPU

LSTM model would be very simple with only 1 hidden LSTM layer of 20 units. I am using the standard Adam optimizer here

Validation RMSPE is actually lower than Training RMSPE, but overall all the lines except for Training MSE look quite unstable.

Forecasting of Close Values by LSTM:

Epoch 1/25
66/66 [==============================] - 2s 9ms/step - loss: 3.0834e-04
Epoch 2/25
66/66 [==============================] - 1s 9ms/step - loss: 1.8928e-04
Epoch 3/25
66/66 [==============================] - 1s 9ms/step - loss: 1.5557e-04
Epoch 4/25
66/66 [==============================] - 1s 9ms/step - loss: 1.1823e-04
Epoch 5/25
66/66 [==============================] - 1s 11ms/step - loss: 8.1981e-05
Epoch 6/25
66/66 [==============================] - 1s 12ms/step - loss: 5.0988e-05
Epoch 7/25
66/66 [==============================] - 1s 12ms/step - loss: 3.2695e-05
Epoch 8/25
66/66 [==============================] - 1s 12ms/step - loss: 1.8494e-05
Epoch 9/25
66/66 [==============================] - 1s 12ms/step - loss: 1.3069e-05
Epoch 10/25
66/66 [==============================] - 1s 11ms/step - loss: 1.3811e-05
Epoch 11/25
66/66 [==============================] - 1s 13ms/step - loss: 1.1377e-05
Epoch 12/25
66/66 [==============================] - 1s 11ms/step - loss: 1.1476e-05
Epoch 13/25
66/66 [==============================] - 1s 11ms/step - loss: 1.2281e-05
Epoch 14/25

```
66/66 [==============================] - 1s 12ms/step - loss: 1.3339e-05
Epoch 15/25
66/66 [==============================] - 1s 12ms/step - loss: 1.2572e-05
Epoch 16/25
66/66 [==============================] - 1s 13ms/step - loss: 1.2845e-05
Epoch 17/25
66/66 [==============================] - 1s 13ms/step - loss: 1.1755e-05
Epoch 18/25
66/66 [==============================] - 1s 13ms/step - loss: 1.1386e-05
Epoch 19/25
66/66 [==============================] - 1s 14ms/step - loss: 1.4467e-05
Epoch 20/25
66/66 [==============================] - 1s 12ms/step - loss: 1.3417e-05
Epoch 21/25
66/66 [==============================] - 1s 17ms/step - loss: 1.6554e-05
Epoch 22/25
66/66 [==============================] - 1s 16ms/step - loss: 1.4734e-05
Epoch 23/25
66/66 [==============================] - 1s 14ms/step - loss: 1.1984e-05
Epoch 24/25
66/66 [==============================] - 1s 13ms/step - loss: 1.0115e-05
Epoch 25/25
66/66 [==============================] - 1s 14ms/step - loss: 1.2537e-05
```

Out[34]:

<keras.callbacks.History at 0x21f9f9daf40>

Validation on Test Data

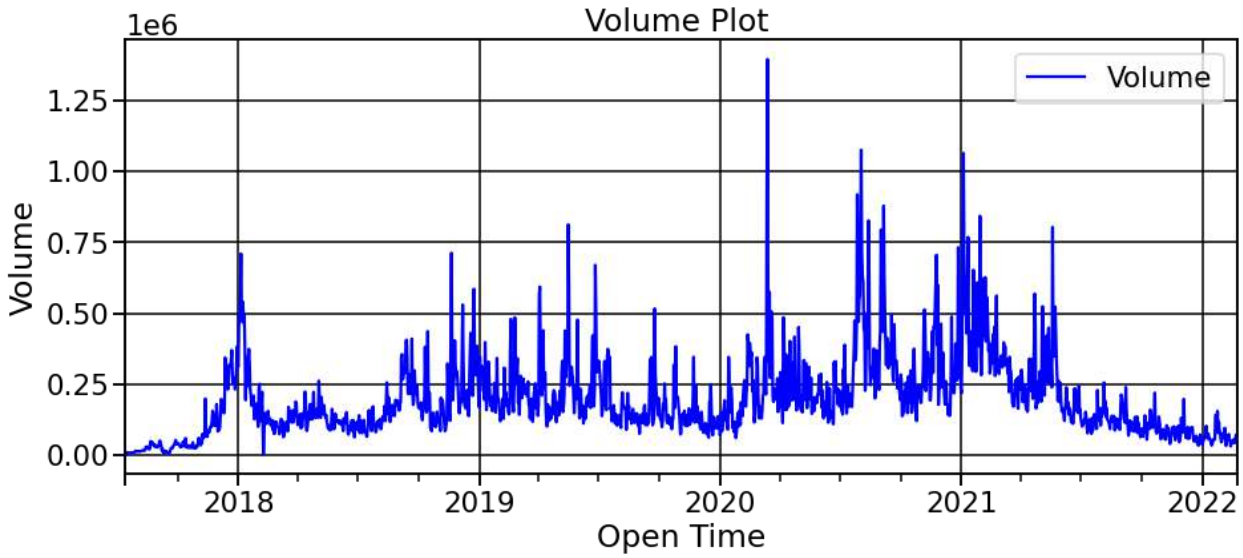
Conclusion : Our Model is good fit

Forecasting for next 30 days

Close prices are decreasing means BTC price increase or ETH price is decreasing

VOLUME ANALYSIS AND FORECASTING

Volume Plot of ETH-BTC:



Conclusion: In March-2020 there was highest no of traders treading in ETH-BTC and after that there was significant decay in no. of treaders

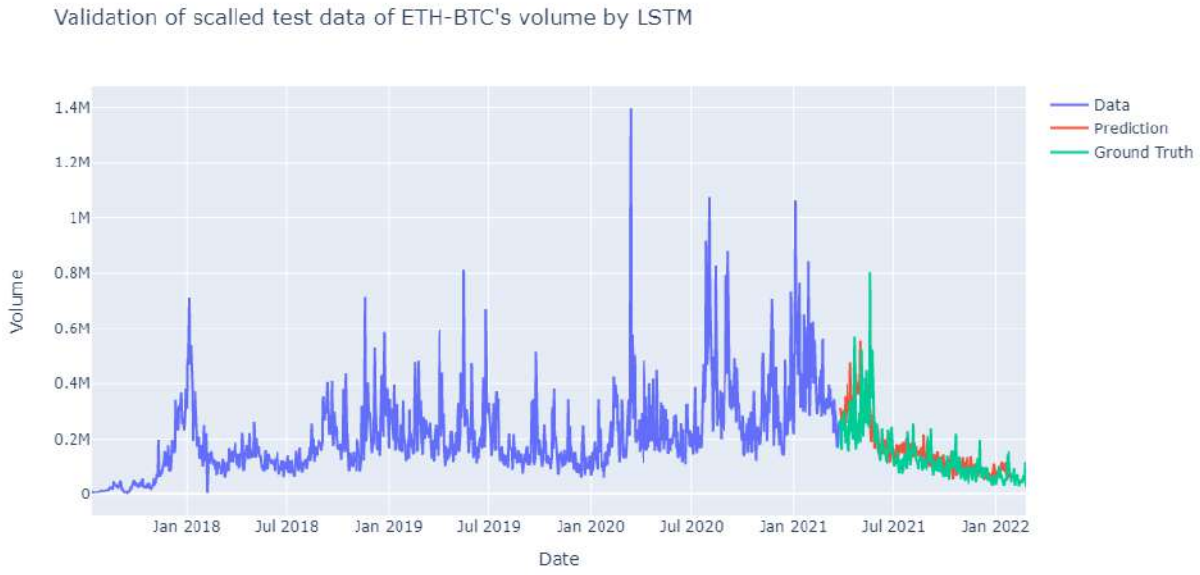
Forecasting of Volume by LSTM

Epoch 1/25
65/65 [==============================] - 2s 10ms/step - loss: 3.6228e-04
Epoch 2/25
65/65 [==============================] - 1s 9ms/step - loss: 1.9373e-04
Epoch 3/25
65/65 [==============================] - 1s 9ms/step - loss: 1.7726e-04
Epoch 4/25
65/65 [==============================] - 1s 9ms/step - loss: 1.5521e-04
Epoch 5/25
65/65 [==============================] - 1s 9ms/step - loss: 1.2727e-04
Epoch 6/25
65/65 [==============================] - 1s 9ms/step - loss: 1.0324e-04
Epoch 7/25
65/65 [==============================] - 1s 9ms/step - loss: 6.8212e-05
Epoch 8/25
65/65 [==============================] - 1s 9ms/step - loss: 5.9350e-05

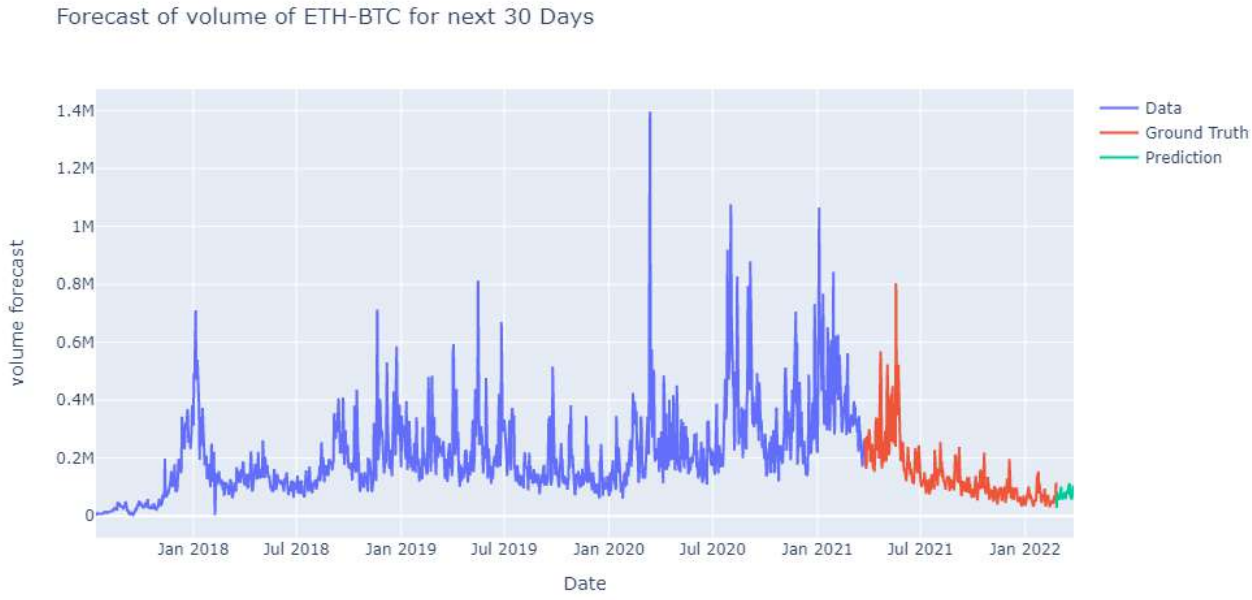
```
Epoch 9/25
65/65 [==============================] - 1s 9ms/step - loss: 3.7027e-05
Epoch 10/25
65/65 [==============================] - 1s 9ms/step - loss: 3.2350e-05
Epoch 11/25
65/65 [==============================] - 1s 9ms/step - loss: 3.6922e-05
Epoch 12/25
65/65 [==============================] - 1s 11ms/step - loss: 3.2319e-05
Epoch 13/25
65/65 [==============================] - 1s 11ms/step - loss: 3.0292e-05
Epoch 14/25
65/65 [==============================] - 1s 11ms/step - loss: 3.0585e-05
Epoch 15/25
65/65 [==============================] - 1s 11ms/step - loss: 2.6696e-05
Epoch 16/25
65/65 [==============================] - 1s 11ms/step - loss: 2.7013e-05
Epoch 17/25
65/65 [==============================] - 1s 11ms/step - loss: 2.3703e-05
Epoch 18/25
65/65 [==============================] - 1s 11ms/step - loss: 2.3350e-05
Epoch 19/25
65/65 [==============================] - 1s 10ms/step - loss: 2.6771e-05
Epoch 20/25
65/65 [==============================] - 1s 10ms/step - loss: 2.3959e-05
Epoch 21/25
65/65 [==============================] - 1s 11ms/step - loss: 2.6040e-05
Epoch 22/25
65/65 [==============================] - 1s 11ms/step - loss: 2.0893e-05
Epoch 23/25
65/65 [==============================] - 1s 10ms/step - loss: 2.0859e-05
Epoch 24/25
65/65 [==============================] - 1s 12ms/step - loss: 1.7278e-05
Epoch 25/25
65/65 [==============================] - 1s 13ms/step - loss: 1.7516e-05
```

Out[186]:

```
<keras.callbacks.History at 0x29064e0da00>
```



Model is good fit



<u>Volume Slightly increasing for next 30 days</u>

VOLATILITY ANALYSIS AND FORECASTING

Volatility

Volatility attempts to measure the **magnitude of price movements that a financial instrument experiences over a certain period of time**. The more dramatic the price swings are in that instrument, the higher the level of volatility, and vice versa.

"Volatility does **not measure the direction** of price changes, merely their dispersion. This is because when calculating standard deviation (or variance), all differences are squared, so that negative and positive differences are combined into one quantity.

Two instruments with different volatilities may have the same expected return, but the instrument with higher volatility will have larger swings in values over a given period of time." (source: Wikipedia)

Volatility can either be historical or implied; both are usually expressed in percentage terms.

- **Historical Volatility** (HV) or **Realized Volatility** is the actual volatility demonstrated by the underlying over a period of time, such as the past month or year.
- Realized Volatility is commonly calculated as the standard deviation of price returns, which is the dollar change in price as a percentage of previous day's price.

The standard deviation will be different for log returns computed over longer or shorter intervals. For this specific project, the volatility would be for a certain fixed interval window (INTERVAL_WINDOW) is the standard deviation of log returns, or the square root of the sum of squares of log returns:

$$\sigma_{interval} = \sqrt{\Sigma_t r_{t-1,t}^2}$$

And then to scale the daily volatility by a certain frequency (weekly, monthly, yearly), we can multiply the daily volatility by the square root of that frequency in terms of day divided by the interval minus 1 (INTERVAL_WINDOW - 1). For example:

$$\sigma_{daily} = \sqrt{\Sigma_t r_{t-1,t}^2} \times \sqrt{\frac{1}{interval-1}}$$

$$\sigma_{annualized} = \sqrt{\Sigma_t r_{t-1,t}^2} \times \sqrt{\frac{365}{interval-1}}$$

$$\sigma_{monthly} = \sqrt{\Sigma_t r_{t-1,t}^2} \times \sqrt{\frac{30}{interval-1}}$$

$$\sigma_{weekly} = \sqrt{\Sigma_t r_{t-1,t}^2} \times \sqrt{\frac{7}{interval-1}}$$

Returns :

$$\sigma_{interval} = \sqrt{\Sigma_t r_{t-1,t}^2}$$

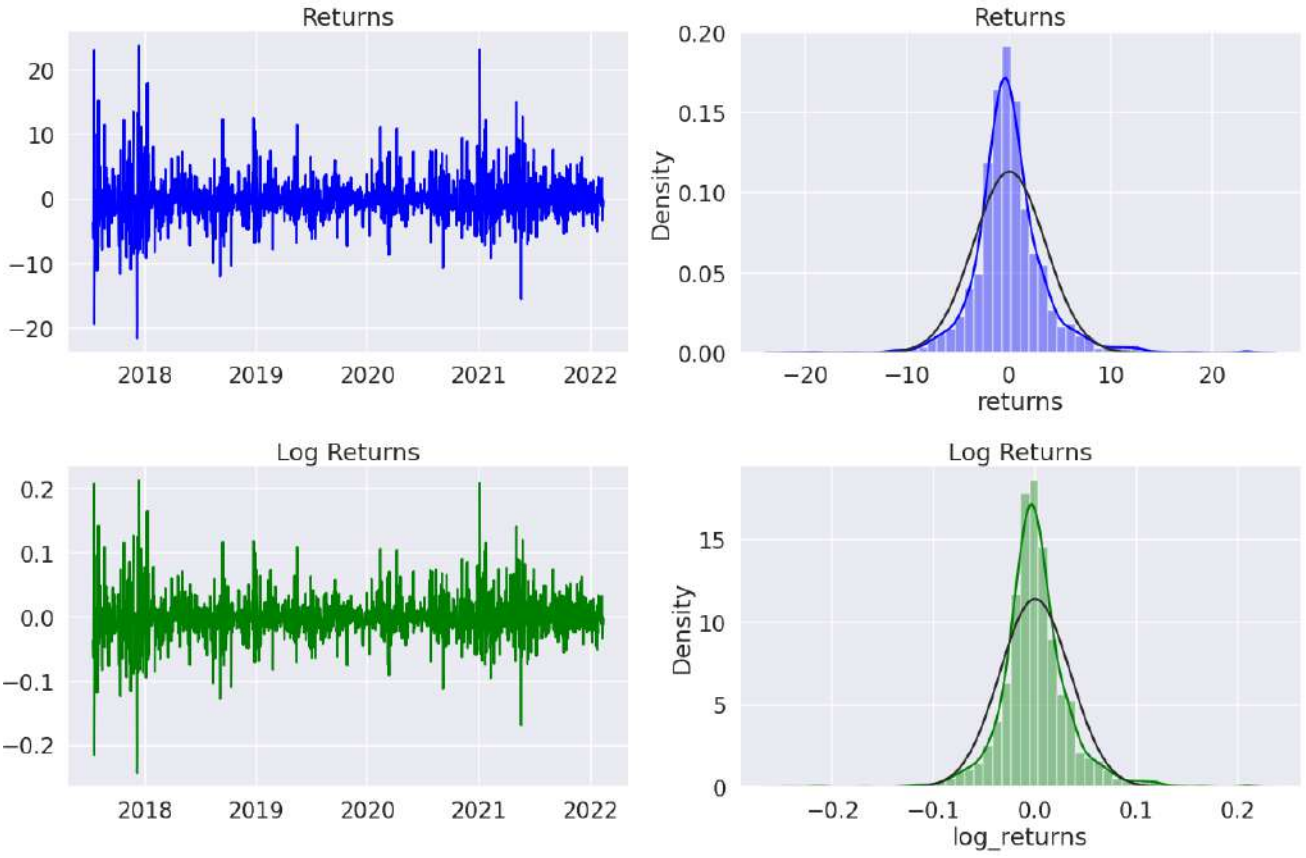
Log Returns:

For practicality purposes, it's generally preferable to use the log returns especially in mathematic modeling, because it helps eliminate non-stationary properties of time series data, and makes it more stable

There's another advantage to log returns, which is that they're additive across time:

Plot Distribution Plots of RETURNS & LOG RETURNS &

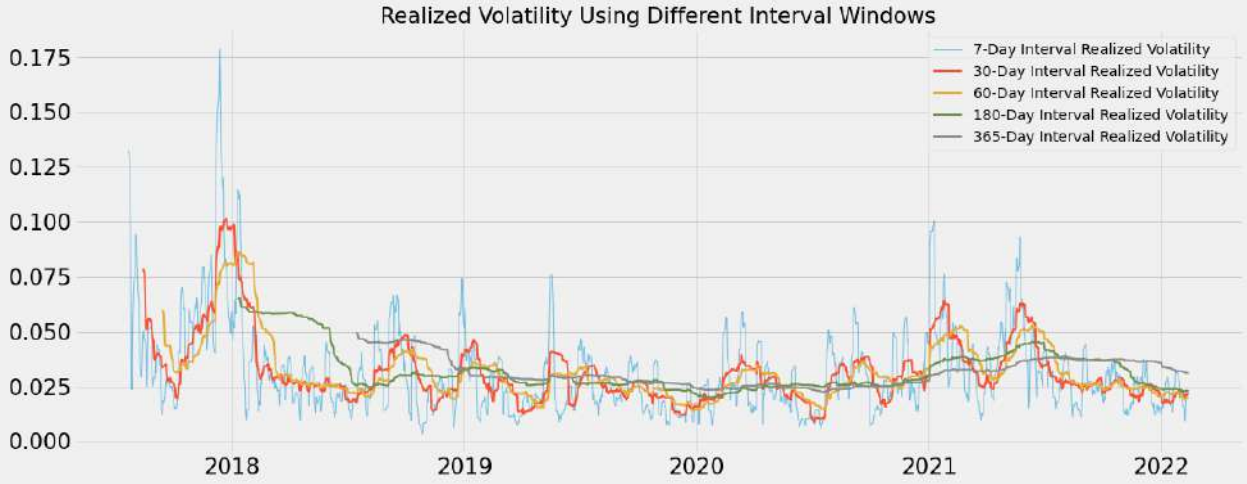
Visually compare them with Standard Normal Distribution



Conclusion: Both Returns & Log Returns show some : slight positive skewness, positive kurtosis (leptokurtic) - higher peak with thicker tails than the standard normal distribution.

Interval Window Selection:

For this specific project I'll use an **interval window of 30 days** (equivalent to roughly 1 month of trading for cryptocurrencies). The goal here is to **forecast the average realized volatility of the next n_future 7 days** using all previous available datapoint with GARCH models (expanding window forecasting), and using a number of immediate past/historical datapoints (n_past) with Neural Networks (sliding window forecasting).



Conclusion:

The reason I selected 30 days is because 7 days seems too noisy to observe meaningful patterns, while longer intervals seem to smooth the volatility down significantly and tend to revert back to the mean.

Using interval window of 30 days would also help avoid wasting too many datapoints at the beginning of the dataset.

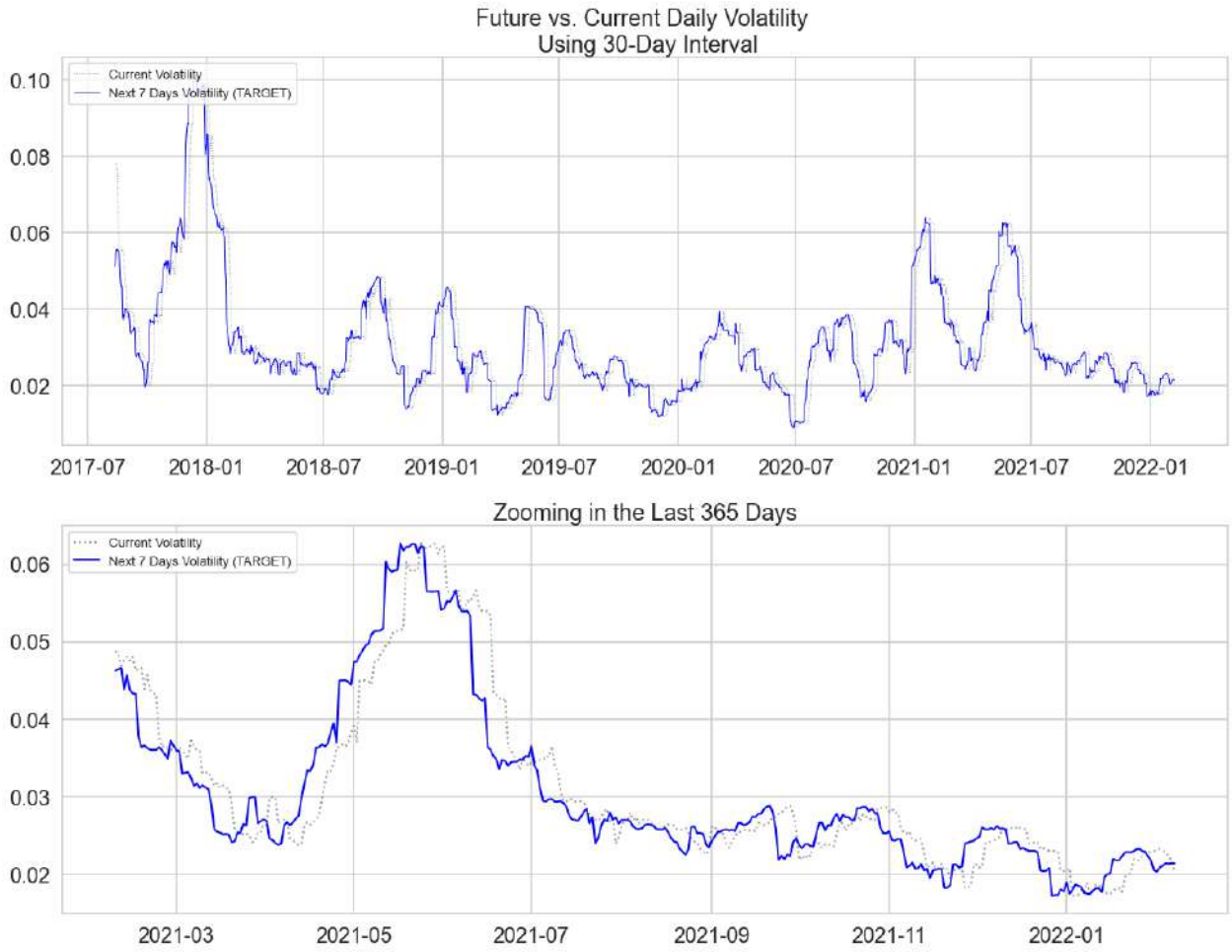
Time-series forecasting models are the models that are capable to predict **future** values based on previously observed values. Target "**future**" data in this case is obtained by **shifting the current volatility backward** by the number of n_future lags.

For example, respected to last week's Monday, this week's Monday is the "**future**"; therefore I just need to shift the volatility this week back by 7 days, and use it as the desired "**future**" output for last week's, which I would then use for Neural Networks training and model performance evaluation.

Since I am currently using an INTERVAL_WINDOW of 30 and a horizon n_future of 7, the volatilty of first 30 values as well as the last 7 values of the dataframe would be NaN, and therefore need to be dropped from the dataset.

Exploratory Data Analysis:

First I would just plot out my desired target outputs vol_future with respect to the current volatility vol_current. The first plot shows all the datapoints I have available that covers 7 years, but then the second plot is only zooming in the most recent 365 days.

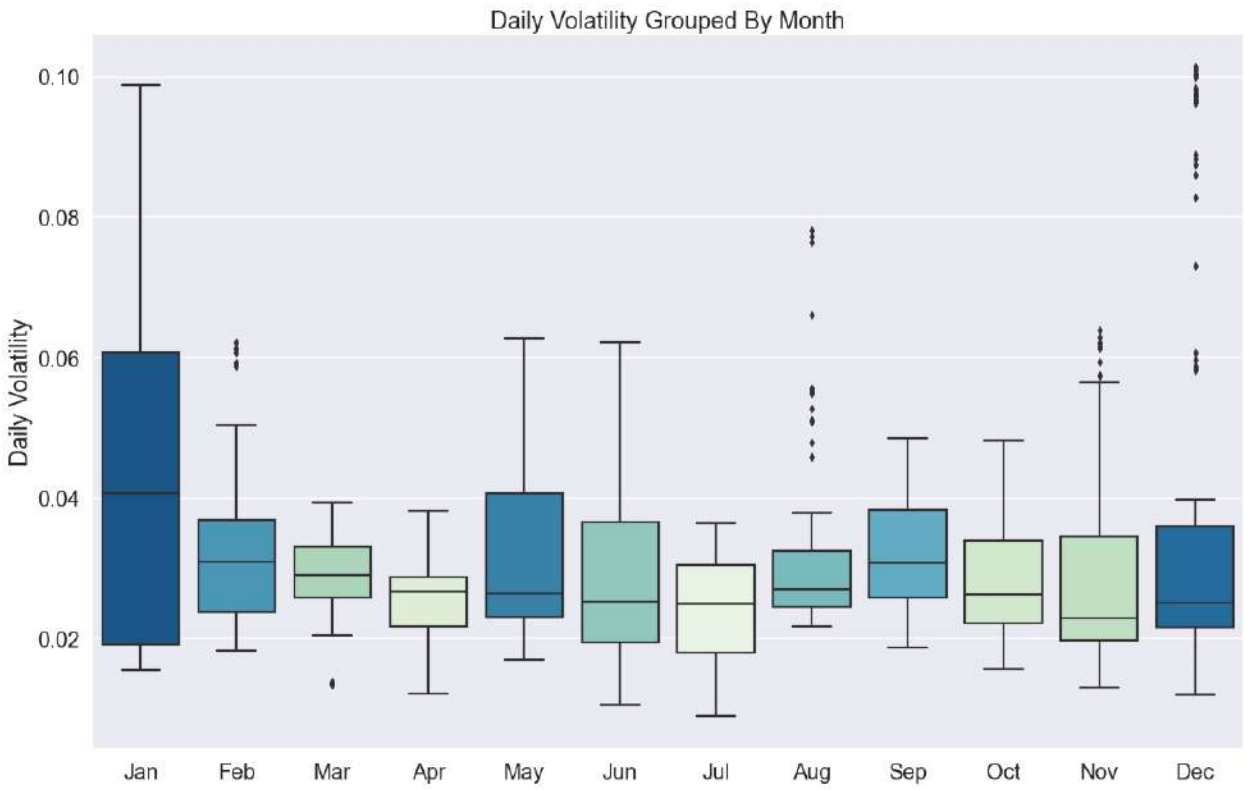


In the plot above, the **blue line** indicates the **target future** value that I ultimately try to match up to.

And the dotted **gray line** represents the **current volatility** in real-time.

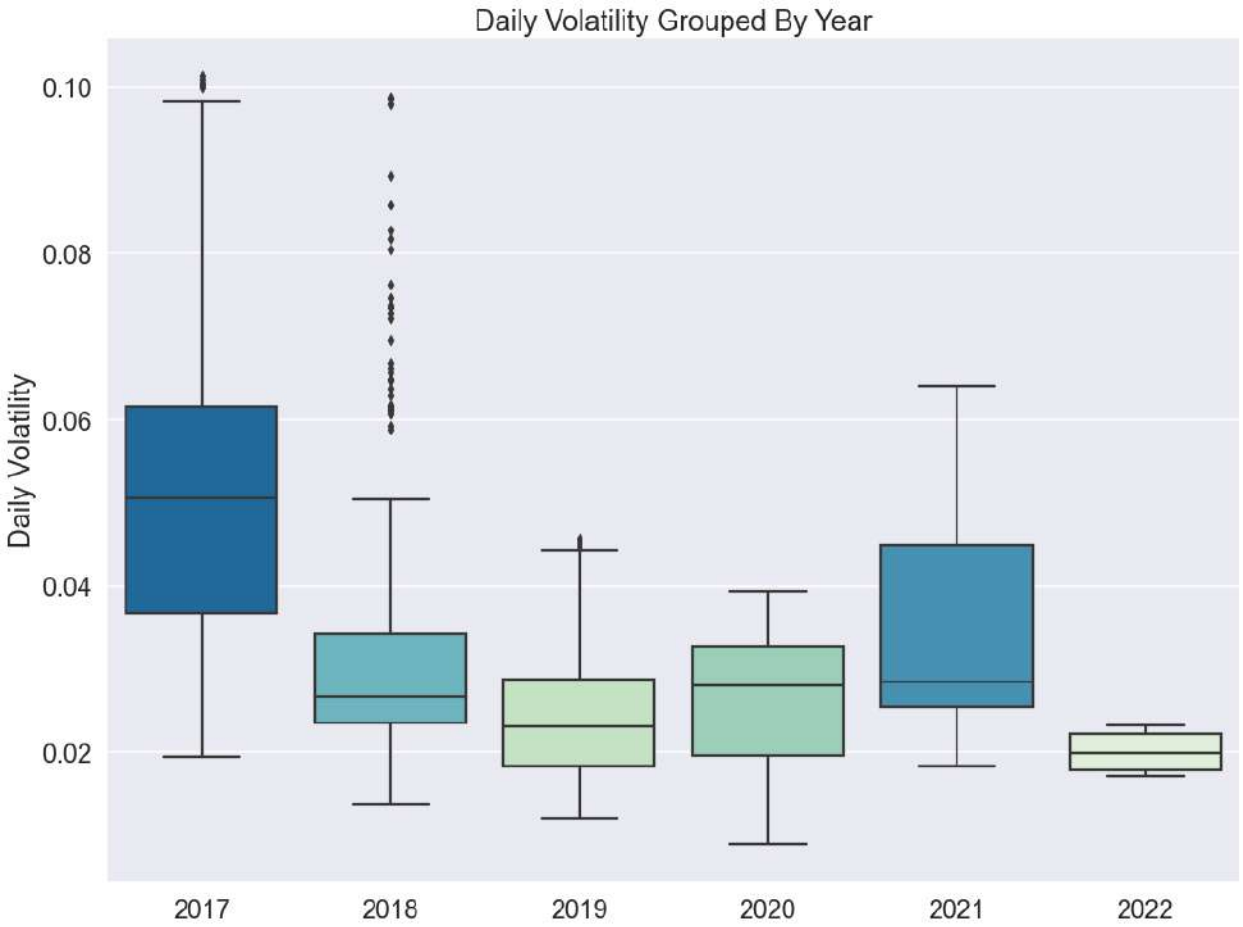
This is a visualization of how current volatility is shifted backward to become future values, which I want to eventually aim for.

Daily Volatility Grouped by Month:



- volatility has consistently reached some of its higher points in the in the months of January historically
- Aug and Dec have the most amount of large outliers

Daily Volatility Grouped by Year:



- volatility has consistently reached some of its higher points in the in the year of 2017 and 2018 historically.
- 2018 have the most amount of large outliers.

Cryptocurrencies have gone through some huge structural changes in the last few years that would've affected volatility directly, such as:

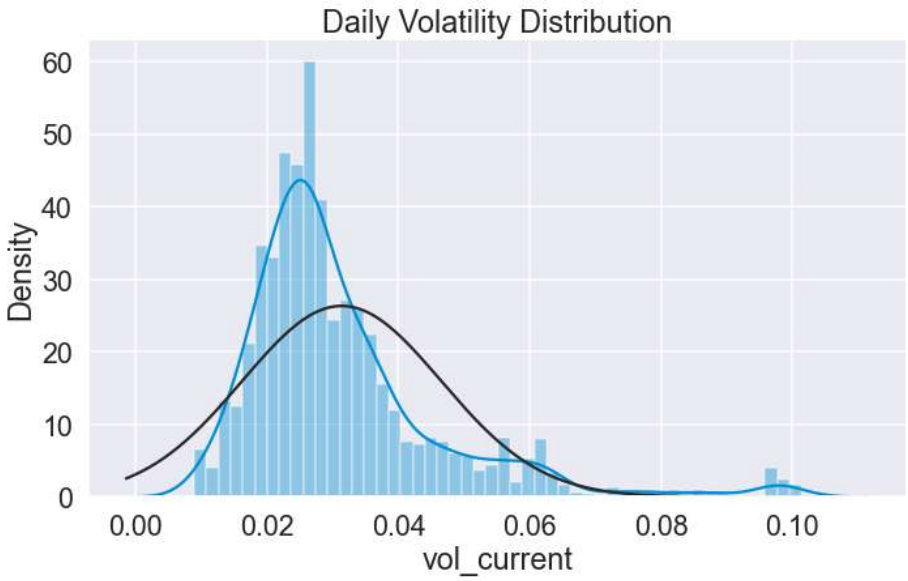
- Crypto Options became available on Deribit in 2016
- Bitcoin Futures was offered on CME in 2017
- and then CME Bitcoin Options in 2020

These events have allowed people to trade crypto volatility more efficiently, and therefore data pre-2016 are likely structurally different, and probably followed different patterns compared to data after 2016.

We can these big events being reflected in the plot above - Bitcoin's first record peak in 2017 (around USD 19,800 towards the end of December). And the outliers in 2020 corresponded with its over 200% surge in 2020 (Bitcoin started at USD 7,200 at the beginning of 2020). It reached USD 20,000 on most exchanges on 12/15/2020, and then proceeded to hit USD 30,000 just 17 days later, which is no small feat. To put things in perspective, it took the Dow Jones close to 3 years to make the same move. And then, on 01/07/2021 it broke USD 40,000. As of the time this report is written, BTC-USD is traded at high USD 49,700.

It can be observed that 2021's daily volatiliy overall has also been on the higher side.

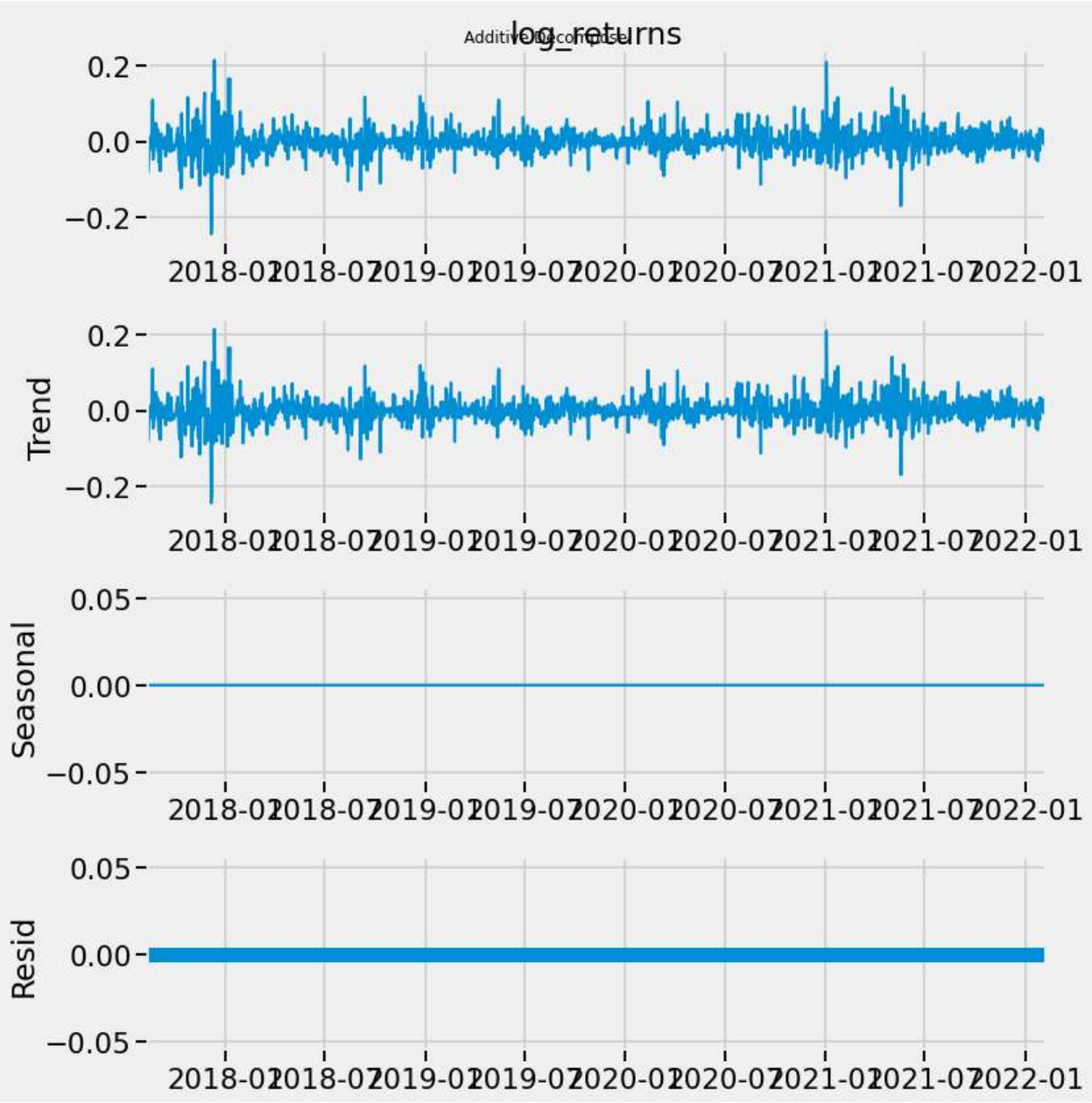
Daily Volatility Distribution:



The distribution of daily realized volatility is lightly right skewed, with a small number of larger values spreaded thinly on the right.

A skewed right distribution would have smaller median compared to mean, and mode smaller than median (mode < median < mean).

Returns and Log Returns Decomposition:



Conclusion : There is no Trend present \n There is no Seasonality present.

Returns and Log Returns Stationarity Checking

Returns

ADF Statistic: -7.156171946822528
p-value: 3.0506955923818777e-10
Critical Values:
1%: -3.4344
5%: -2.8633
10%: -2.5677

Log Returns

ADF Statistic: -7.28402390437071
p-value: 1.4748762010608668e-10
Critical Values:
1%: -3.4344
5%: -2.8633
10%: -2.5677

Using a significant level alpha of 0.05, p-value for both Returns and Log Returns are significantly smaller than alpha, which means there's enough evidence to reject the Null Hypothesis.

H0: Process is not stationary

V/s

H1: Process is stationary

Here p-value too small than alpha Hence reject H0 and conclude that process is stationary.

--> Returns and Log returns are both not dependent on time/trend

Train-Validation-Test Splits:

There're a total of 1650 usable datapoints in this dataset which covers a period of almost 4.5 years from January-2017 until today (end of january 2022). Since cryptocurrencies are not traded on a regulated exchange, the Bitcoin market is open 24/7, 1 year covers a whole 365 trading days instead of 252 days a year like with other stocks and commodities.

I would split the dataset into 3 parts as follows:

the most recent 30 usable datapoints would be used for Final Model Testing - approx. 1.81%

1 full year (365 days) for Validation and Model Tuning during training - approx. 22.12%

and the remaining for Training - approx.76.07%

The target here would be

vol_future which represents the daily realized volatility of the next

n_future days from today (average daily volatility from t + n_future - INTERVAL_WINDOW to time step t + n_future).

For example, using an n_future value of 7 and an INTERVAL_WINDOW of 30, the value that I want to predict at time step t would be the average daily realized volatility from time step t-22 to time step t+7.

MODELING:

Performance Metrics

Usually with financial time series, if we just shift through the historic data trying different methods, parameters and timescales, it's almost certain to find to some strategy with in-sample profitability at some point. However the whole purpose of "forecasting" is to predict the future based on currently available information, and a model that performs best on training data might not be the best when it comes to out-of-sample generalization (or overfitting). Avoiding/Minimizing overfitting is even more important in the constantly evolving financial markets where the stake is high.

The 2 main metrics I'd be using are **RMSPE (Root Mean Squared Percentage Error)** and **RMSE (Root Mean Square Errors)** with RMSPE prioritized. Timescaling is very important in the calculation of volatility due to the level of freedom in frequency/interval window selection. Therefore I think RMSPE would help capture degree of errors compared to desired target values better than other metrics. Also RMSPE would punish large errors more than regular MAPE (Mean Absolute Percentage Error), which is what I want to do here.

RMSE and RMSPE would be tracked across different models' performance on validation set forecasting to indicate their abilities to generalize on out-of-sample data.

Fitting of ARIMA

Autoregressive integrated moving average (ARIMA) is a statistical regression model, which can be utilized in time series forecasting applications, such as finance. ARIMA makes predictions while considering the lagged values of a time series. Lags of the stationarized series in the forecasting equation are called "autoregressive" terms, lags of the forecast errors are called "moving average" terms, and a time series which needs to be differenced to be made stationary is said to be an "integrated" version of a stationary series. Random-walk and random-trend models, autoregressive models, and exponential smoothing models are all special cases of ARIMA models. A nonseasonal ARIMA model is classified as an "ARIMA(p,d,q)" model, where: • p is the number of autoregressive terms, • d is the number of nonseasonal differences needed for stationarity, and • q is the number of lagged forecast errors in the prediction equation.

In terms of y, the general forecasting equation is: ŷt = μ + φ1 yt-1 +…+ φp yt-p - θ1et-1 -…- θqet-q Here the moving average parameters (θ's) are defined so that their signs are negative in the equation

Performing stepwise search to minimize aic
ARIMA(1,0,1)(0,0,0)[0] intercept : AIC=-4884.264, Time=0.69 sec
ARIMA(0,0,0)(0,0,0)[0] intercept : AIC=-4872.821, Time=0.28 sec
ARIMA(1,0,0)(0,0,0)[0] intercept : AIC=-4885.730, Time=0.23 sec
ARIMA(0,0,1)(0,0,0)[0] intercept : AIC=-4886.155, Time=0.46 sec
ARIMA(0,0,0)(0,0,0)[0] : AIC=-4874.154, Time=0.12 sec
ARIMA(0,0,2)(0,0,0)[0] intercept : AIC=-4884.245, Time=0.35 sec
ARIMA(1,0,2)(0,0,0)[0] intercept : AIC=-4882.361, Time=0.33 sec
ARIMA(0,0,1)(0,0,0)[0] : AIC=-4887.580, Time=0.17 sec
ARIMA(1,0,1)(0,0,0)[0] : AIC=-4885.686, Time=0.10 sec
ARIMA(0,0,2)(0,0,0)[0] : AIC=-4885.678, Time=0.12 sec
ARIMA(1,0,0)(0,0,0)[0] : AIC=-4887.179, Time=0.21 sec
ARIMA(1,0,2)(0,0,0)[0] : AIC=-4883.834, Time=0.53 sec

Best model: ARIMA(0,0,1)(0,0,0)[0]
Total fit time: 3.601 seconds

Out[93]:

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	1245
Model:	SARIMAX(0, 0, 1)	**Log Likelihood**	2445.790

Date:	Mon, 14 Feb 2022	**AIC**	-4887.580
Time:	12:00:13	**BIC**	-4877.326
Sample:	0	**HQIC**	-4883.724
	- 1245		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	0.1093	0.015	7.359	0.000	0.080	0.138
sigma2	0.0012	2.3e-05	50.145	0.000	0.001	0.001

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	2401.10
Prob(Q):	0.95	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.43	**Skew:**	0.34
Prob(H) (two-sided):	0.00	**Kurtosis:**	9.77

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	1245
Model:	SARIMAX(0, 0, 1)	**Log Likelihood**	2445.790
Date:	Mon, 14 Feb 2022	**AIC**	-4887.580
Time:	12:00:13	**BIC**	-4877.326
Sample:	0	**HQIC**	-4883.724
	- 1245		

Covariance Type:			opg			
	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
ma.L1	0.1093	0.015	7.359	0.000	0.080	0.138
sigma2	0.0012	2.3e-05	50.145	0.000	0.001	0.001

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	2401.10
Prob(Q):	0.95	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.45	**Skew:**	0.34
Prob(H) (two-sided):	0.00	**Kurtosis:**	9.77

so here our Best model: ARIMA(0,0,1)(0,0,0)[0] In our fitted SARIMA Model d=P=D=Q=0 so it is MA(1) model

but we dont consider this model because our best fitted model is chanching everytime due to incoming data and Heteroskedasticity

Heteroskedasticity(H) : 0.45

H0 : Homoskedasticity is present present (residuals are equally scattered) v/s H1 : Heteroskedasticity is present present (residuals are not equally scattered)

Prob(H) (two-sided) : 0.00 < Alpha (0.05)

Hence we __Reject H0__ and conclude that __significant Heteroskedasticity is pres ent (0.45)__

The approach is that the series should be stationary other than the change in variance meaning it does not consider TREND and Seasonal component other than the Heteroskedasticity.

Therefore ARCH(p) model will be udsed to predict future time steps.

But here q also Significant so we go for the GARCH(p,q) model

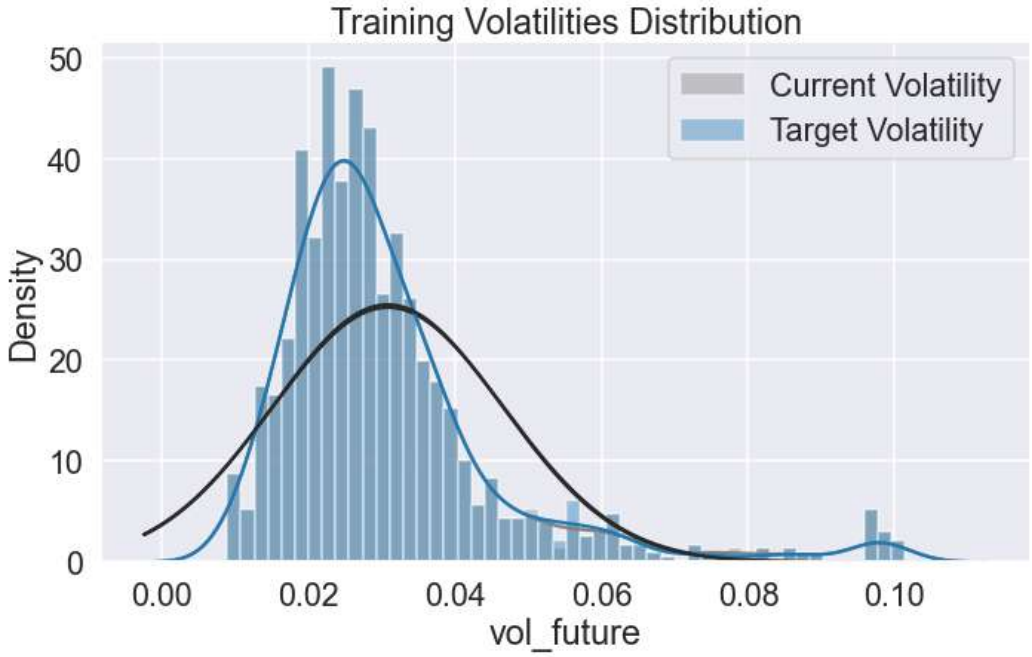
p : The no. of lag variances to include in the GARCH model

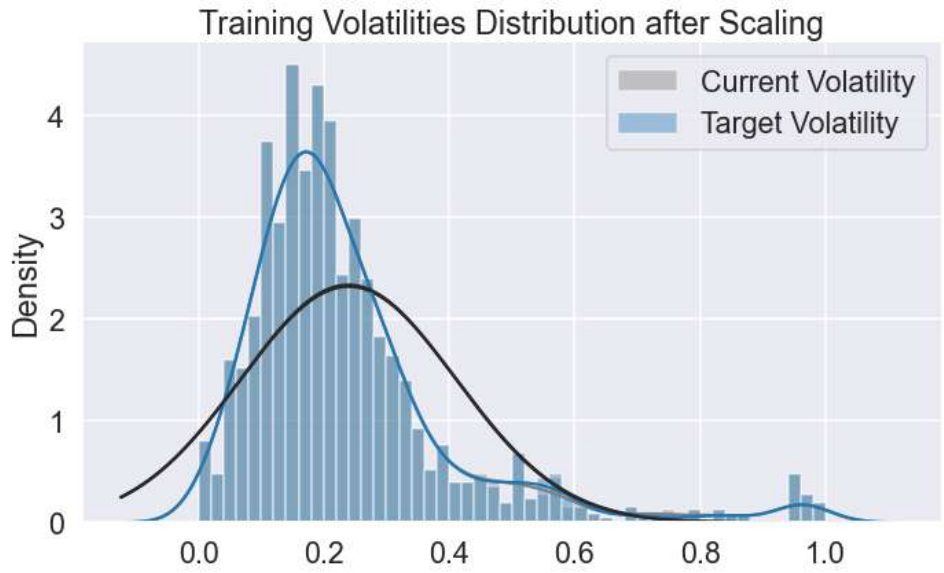
q : The no. of lag residual errors to include in the garch model

Feature Normalization:

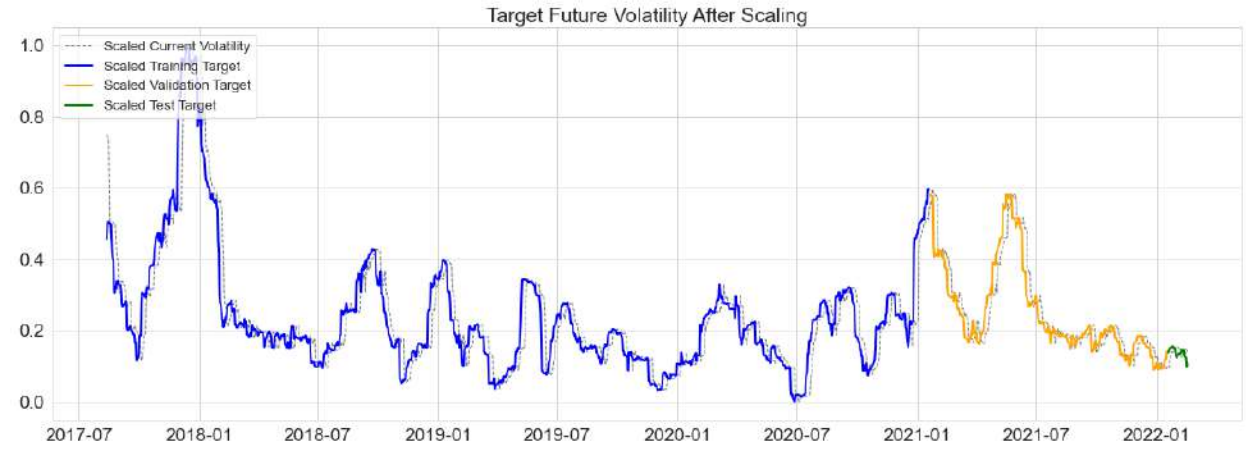
Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to use a common scale, without distorting differences in the ranges of values or losing information. We have performed min-max scaling for Normalization which scales the attributes to the range (-1,1).

As I am going to build various types of models based on different algorithms using different types of inputs, it would be better to normalize the volatilities to standardize the predictions generated by different models. After experimenting with different Scalers, I decided to use MinMaxScaler as it yielded best results overall.

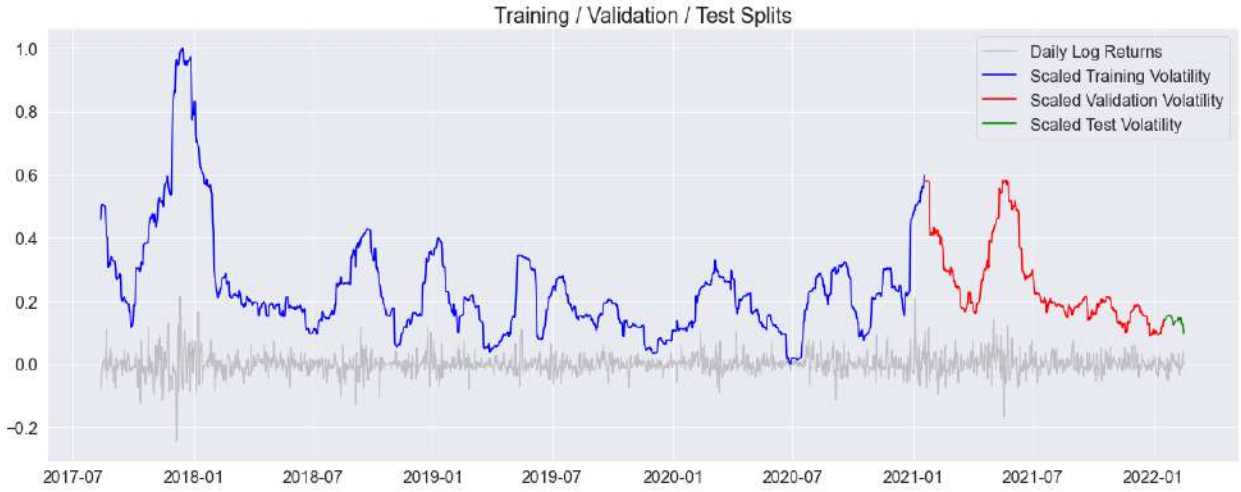




Train-Validation-Test Visualization



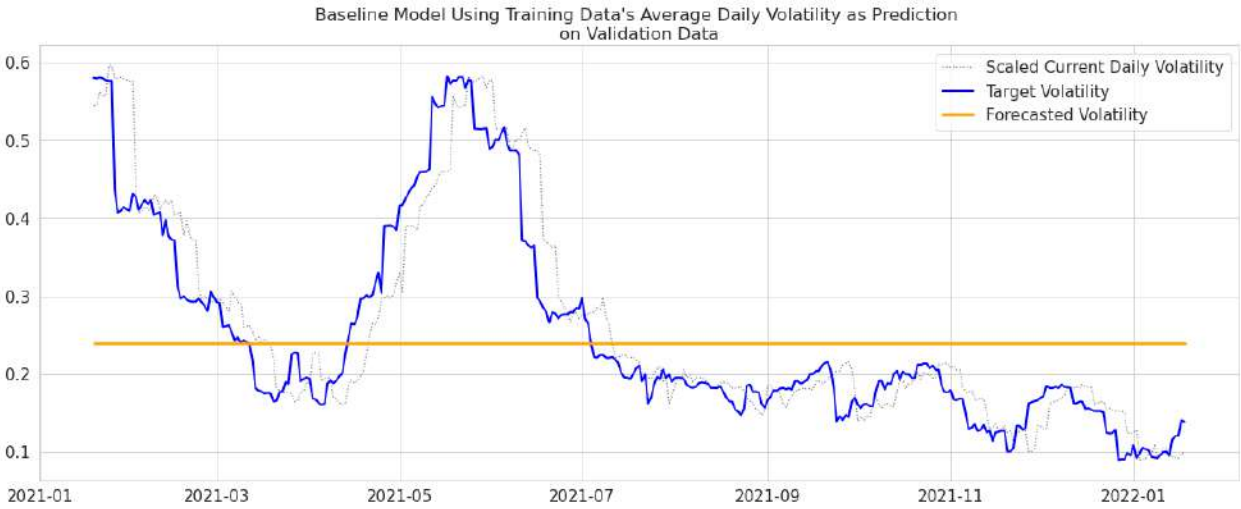
One of the most obvious difference between the 2 plots is the y-axis. After scaling, volatilities raise out of the range of [0,1.0] instead of [0, 0.10] as before.



Baseline Models:

Mean Baseline:

One of the essential characteristics of Volatility is it's **mean-revert** over the long term. Therfore my first baseline model his would be a very simple one that only outputs the average current realized volatility of the whole training set as predictions everything.



	Model	*Validation RMSPE*	*Validation RMSE*
0	*Mean Baseline*	*0.544414*	*0.11084*

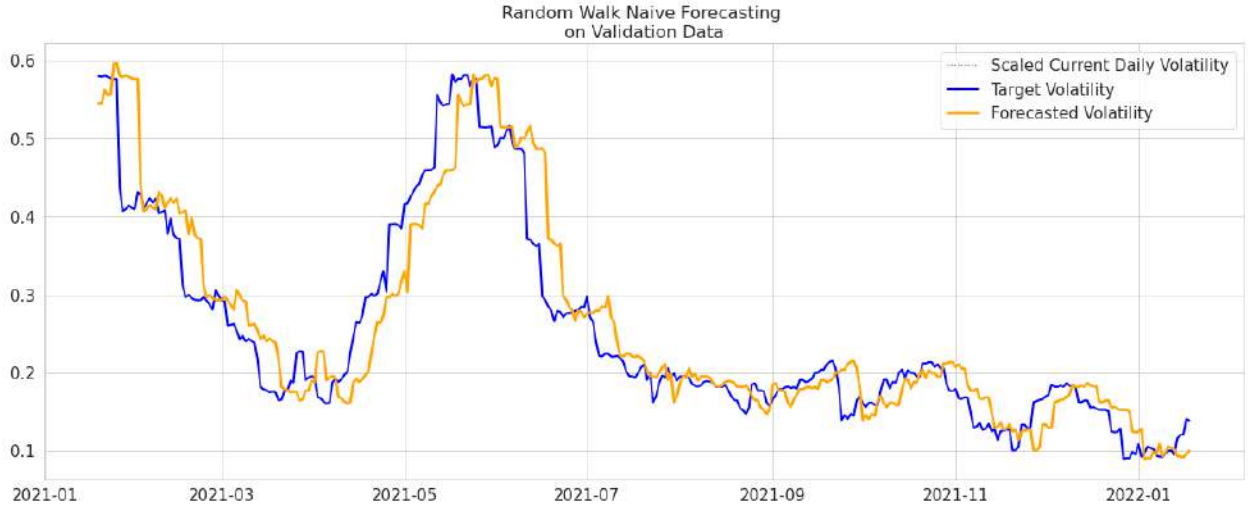
Random Walk Naive Forecasting:

A commonly known fact about volatility is that it tends to be autocorrelated, and clusters in the short-term. This property can be used to implement a naive model that just "predicts" future volatility by using whatever the daily volatility was at the immediate previous time step.

In this case, I'll use the average daily volatility of the most recent INTERVAL_WINDOW as predictions for the next 7 days, which is essentially using vol_current at time step t and prediction for vol_future at time step t.

One of the simplest and yet most important models in time series forecasting is the random walk model. This model assumes that in each period the variable takes a random step away from its previous value, and the steps are independently and identically distributed in size ("i.i.d."). This is equivalent to saying that the first difference of the variable is a series to which the mean model should be applied.

In other words, it predicts that all future values will equal the last observed value. This doesn't really mean you expect them to all be the same, but just that you think they are equally likely to be higher or lower, and you are staying on the fence as far as point predictions are concerned. Because of this, this method is usually known as **naive method**.



	Model	*Validation RMSPE*	*Validation RMSE*
0	*Mean Baseline*	*0.54441*	*0.11084*
1	*Random Walk Naive Forecasting*	*0.198058*	*NaN*

Conclusion:

So the both error mertrics have gone down by a decent amount. Naive forecasting in time series is sometimes surprisingly difficult to outperform.

GARCH Models:

GARCH stands for **Generalized Autoregressive Conditional Heteroskedasticity**, which is an extension of the ARCH model (Autoregressive Conditional Heteroskedasticity).

GARCH includes lag variance terms with lag residual errors from a mean process, and is the traditional econometric approach to volatility prediction of financial time series.

Mathematically, GARCH can be represented as follows:

$$\sigma_t^2 = \omega + \sum_{i}^{q} \alpha_i \epsilon_{t-i}^2 + \sum_{1}^{\rho} \beta_i \sigma_{t-i}^2$$

in which σ_t^2 is variance at time step t and ϵ_{t-i}^2 is the model residuals at time step t-1

GARCH(1,1) only contains first-order lagged terms and the mathematic equation for it is:

$\sigma_t^2 = \omega + \alpha\epsilon_{(t-1)}^2 + \beta\sigma_{(t-1)}^2$
where α, β and ω sum up to 1, and ω is the long term variance.

GARCH is generally regarded as an insightful improvement on naively assuming future volatility will be like the past, but also considered widely overrated as predictor by some experts in the field of volatility. GARCH models capture the essential characteristics of volatility: volatility tomorrow will be close to what it is today (**clustering**), and volatility in the long term will probably **mean revert** (meaning it'd be close to whatever the historical long-term average has been).

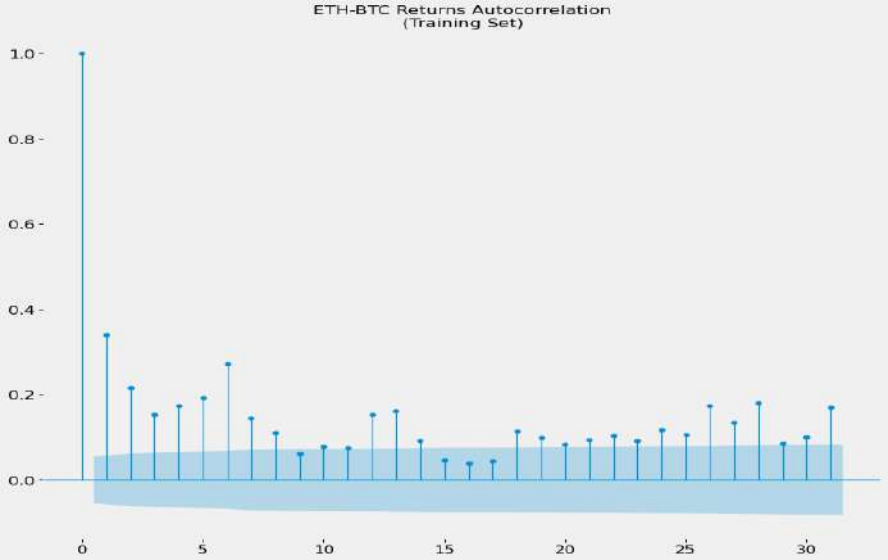
Generalized autoregressive conditional heteroskedasticity (GARCH) models aim to model the conditional volatility of a time series. Let r_t be the dependent variable, for example the returns of a stock in time t. We can model this series as:

$$r_t = \mu + \sigma_t \epsilon_t$$

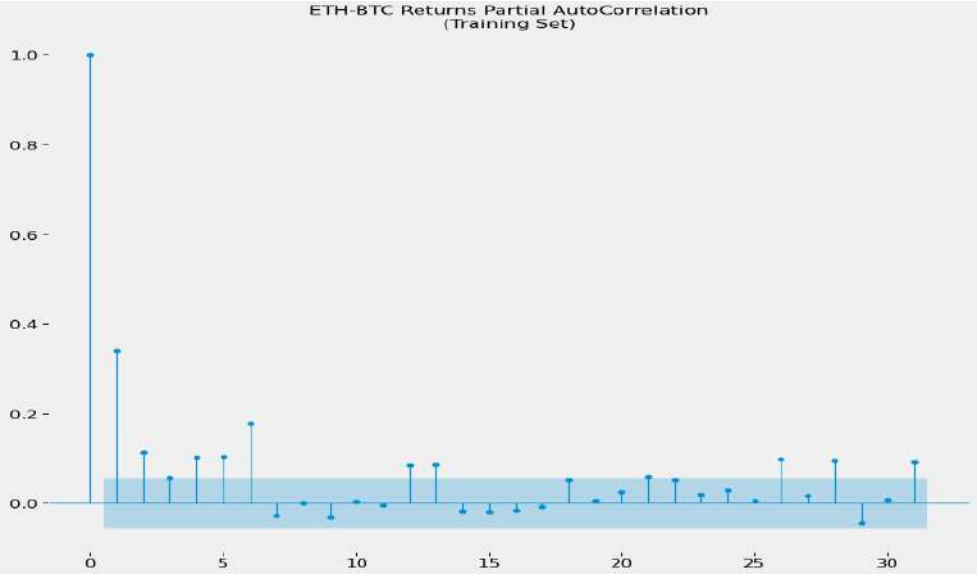
Here mu is the expected value of r_t, σ_t is the standard deviation of r_t in time t, and ϵ_t is an error term for time t.

Basic GARCH Model:

ACF Plot:



PACF Plot:



Conclusion : *From ACF and PACF plots we conclude thatour volatility is not Stationary*

GARCH(1,1):

GARCH(1,1) only contains first-order lagged terms and the mathematic equation for it is:

$$\sigma_t^2 = \omega + \alpha\epsilon_{(t-1)}^2 + \beta\sigma_{(t-1)}^2$$

where α, β and ω sum up to 1, and ω is the long-term variance.

GARCH is generally regarded as an insightful improvement on naively assuming future volatility will be like the past, but also considered widely overrated as predictor by some experts in the field of volatility. GARCH models capture the essential characteristics of volatility: volatility tomorrow will be close to what it is today (**clustering**), and volatility in the long term will probably **mean revert** (meaning it'd be close to whatever the historical long-term average has been)

Constant Mean - GARCH Model Results

Dep. Variable:	returns	R-squared:	0.000
Mean Model:	Constant Mean	Adj. R-squared:	0.000
Vol Model:	GARCH	Log-Likelihood:	-3154.54
Distribution:	Normal	AIC:	6317.07
Method:	Maximum Likelihood	BIC:	6337.61
		No. Observations:	1255
Date:	Thu, Feb 24 2022	Df Residuals:	1254
Time:	13:37:17	Df Model:	1

Mean Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
mu	-0.0541	7.765e-02	-0.696	0.486	[-0.206,9.813e-02]

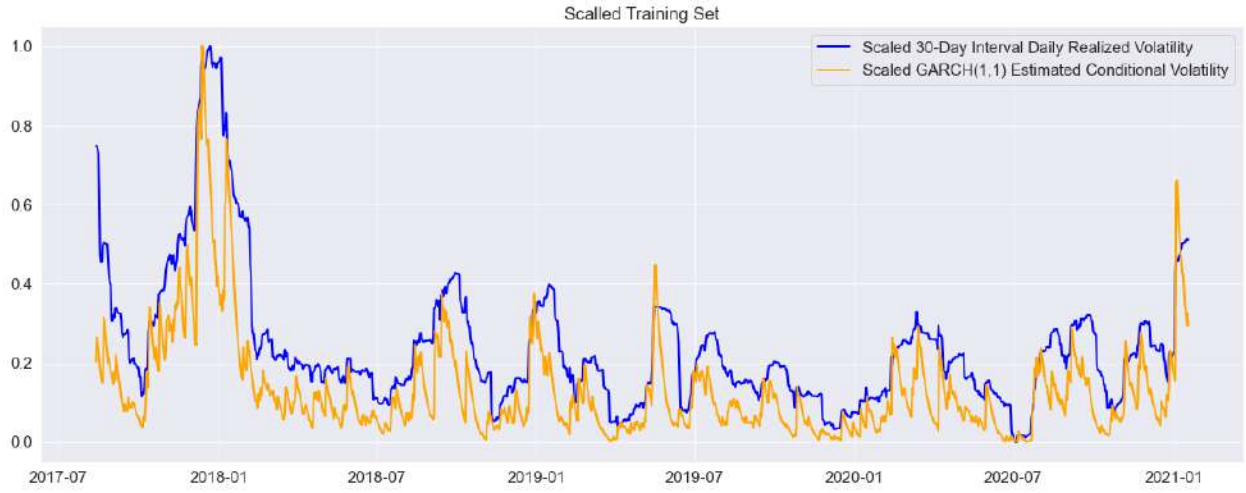
Volatility Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	0.3900	0.264	1.475	0.140	[-0.128, 0.908]
alpha[1]	0.1182	4.669e-02	2.533	1.132e-02	[2.673e-02, 0.210]
beta[1]	0.8522	6.390e-02	13.336	1.432e-40	[0.727, 0.977]

Covariance estimator: robust

All coefficients look statistically significant now.

After being fitted to the training data (percent returns), the GARCH model forecast contains an estimated conditional volatility attribute for the training portion of the time series. I am going to fit - transform the scaler to the training data's conditional volatility arrays, and then plot it out to compare it with the realized volatility calculated above (also scaled) x_train_scaled.

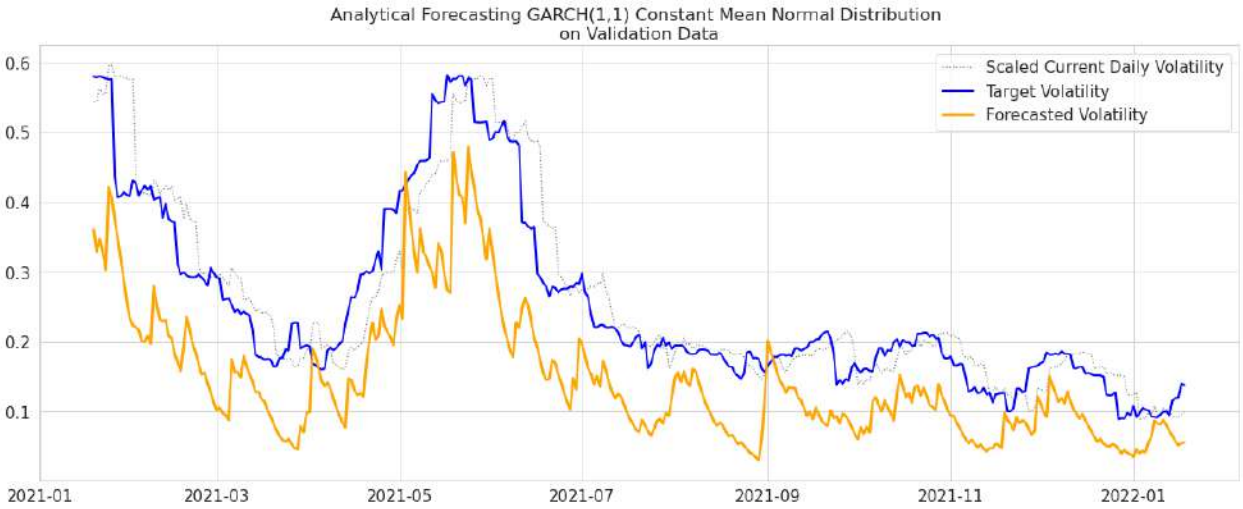


Analytical-based Forecasting:

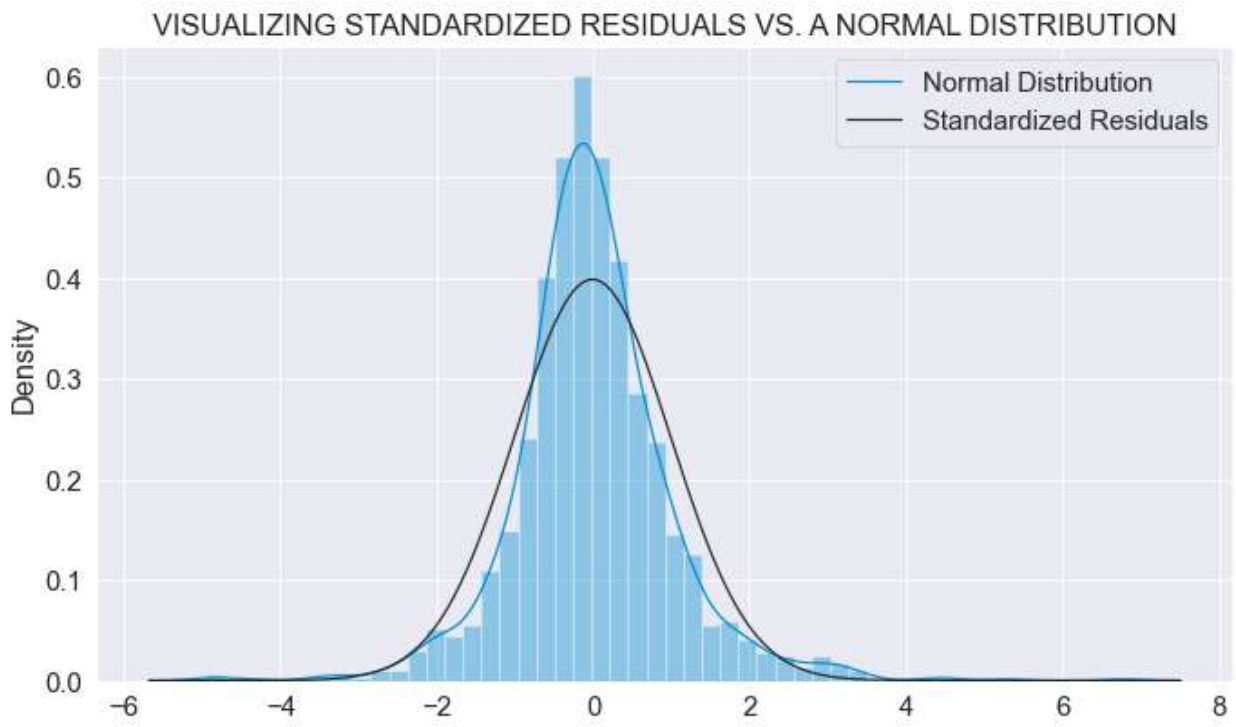
(Source: https://arch.readthedocs.io/en/latest/univariate/univariate_volatility_forecasting.html)

I am implementing rolling one-step forecasting here by refitting the model to ALL the datapoints available up to a certain time step before generating prediction for the next n_future days. For example, to make prediction on vol_future at time step t, I would fit the model to all the returns available up to time step t and then obtain the average forecasted volatility for a horizon of n_future = 7.

The volatility forecasts from the GARCH models using percent returns as input are on a totally different scale compared to the volatilities calculated from log returns. Therefore I am going to normalize the forecasted volatility based on the model's conditional_volatility output from the training data, and only comparing the scaled versions of volatilities on the Validation set.



	Model	*Validation RMSPE*	*Validation RMSE*
0	*Mean Baseline*	*0.544414*	*0.11084*
1	*Random Walk Naive Forecasting*	*0.198058*	*NaN*
2	*GARCH(1,1), Constant Mean, Normal Dist*	*0.456065*	*0.32271*



(Reference: https://goldinlocks.github.io/ARCH_GARCH-Volatility-Forecasting/)

By default, basic GARCH is based on some assumptions that the residuals and the mean return are both normally distributed. However, more often than not, financial time series data does not follow a normal distribution, and it's more likely to observe extreme positive and negative values that are far away from the mean.

Fortunately, there're certain parameters that can be specified in the arch_model library to make it more representative of real financial data:

'dist' can be set to t for Student's T or skewt for skewed Student's T distribution (Student's T distribution is also symmetric and bell-shaped like normal distribution; however it has higher peak and fatter tails allowing more values lying further away from the mean). Looking at the plot above of the standardized residuals, I think I can try skewed Student's T for the next fit.

GARCH Model with Asymmetric Shocks Responses:

GARCH model with asymmetric Shocks Responses

The basic GARCH model assumes positive and negative news have similar impact on volatility. The impact is usually asymmetric, and negative impacts tends to affect the volatility more than positive ones.

There's another member in the GARCH family that accounts for asymmetry of shocks responses called **GJR-GARCH**

model which adjusts even for asymmetric responses of volatility to innovation fluctuations. GJR-GARCH was developed by Glosten, Jagannathan, Runkle in 1993. Sometimes referred as T-GARCH or TARCH if just ARCH with GJR modification is used. GJR-GARCH(p, q, r) is defined as follows

$\boldsymbol{y_t = \mu_t + e_t}$, $\boldsymbol{e_t = \sigma_t \varepsilon_t}$,

$$\boldsymbol{\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i e_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 + \sum_{k=1}^{r} \gamma_k e_{t-k}^2 I_{t-k}} \ ,$$

$$I_t = \begin{cases} 1 & \text{if } e_t < 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\boldsymbol{\gamma_k}$ are leverage coefficients and $\boldsymbol{I_t}$ is indicator function. Observe that for $\boldsymbol{\gamma_k > 0}$ negative innovations $\boldsymbol{e_t}$ give additional value to volatility $\boldsymbol{\sigma_t^2}$ thus we achieve adjustment for asymmetric impact on volatility as discussed at the beginning of the article. For $\boldsymbol{\gamma_k = 0}$ we get GARCH(m = p, n = q) model and for $\boldsymbol{\gamma_k < 0}$ we get exotic result where upward swings in return or price have stronger impact on volatility than the downward moves. Need to mention that in most implementations of GJR-GARCH we will find GJR-GARCH(p,q) where leverage order $\boldsymbol{o}$ is automatically considered equal to order $\boldsymbol{p}$. Parameter's constraints are again very similar as for GARCH, we have

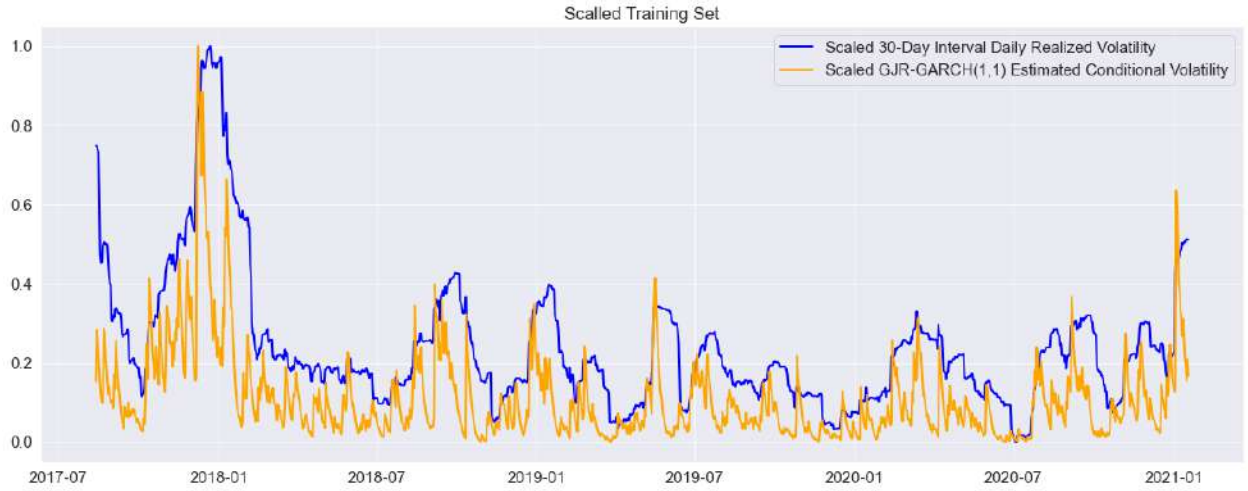
$$\boldsymbol{\alpha_0 = 0,\ \alpha_i \geq 0,\ \beta_j \geq 0,\ \alpha_i + \gamma_k \geq 0,\ \sum_{i=1}^{p} \alpha_i + \sum_{j=1}^{q} \beta_j + \frac{1}{2}\sum_{k=1}^{r} \gamma_k < 1}\ .$$

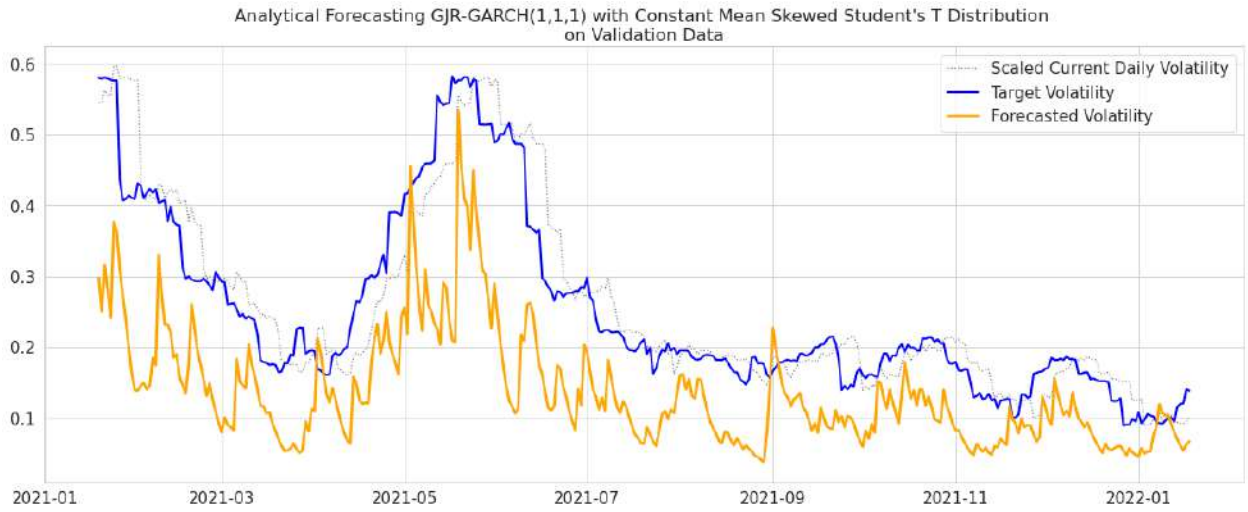
Constant Mean - GJR-GARCH Model Results

```
==============================================================================
==========
Dep. Variable:                  returns   R-squared:                  0.000
Mean Model:                Constant Mean   Adj. R-squared:             0.000
Vol Model:                     GJR-GARCH   Log-Likelihood:          -3035.64
Distribution:     Standardized Skew Student's t   AIC:                 6085.27
Method:                Maximum Likelihood   BIC:                    6121.21
                                            No. Observations:             1255
Date:                   Thu, Feb 24 2022   Df Residuals:                 1254
Time:                           13:37:33   Df Model:                        1
                          Mean Model
==============================================================================
            coef    std err          t      P>|t|      95.0% Conf. Int.
------------------------------------------------------------------------------
mu         -0.0605  7.437e-02     -0.813      0.416 [ -0.206,8.529e-02]
                        Volatility Model
==============================================================================
            coef    std err          t      P>|t|      95.0% Conf. Int.
------------------------------------------------------------------------------
omega        0.8464      0.328      2.583  9.785e-03   [  0.204,  1.488]
alpha[1]     0.1988  6.133e-02      3.241  1.192e-03 [7.855e-02,  0.319]
gamma[1]     0.1544  7.607e-02      2.030  4.239e-02 [5.299e-03,  0.304]
beta[1]      0.7240  6.564e-02     11.030  2.740e-28   [  0.595,  0.853]
                          Distribution
==============================================================================
            coef    std err          t      P>|t|      95.0% Conf. Int.
------------------------------------------------------------------------------
```

```
eta          3.2753      0.293     11.164  6.090e-29   [  2.700,  3.850]
lambda       0.1089  3.581e-02      3.041  2.362e-03 [3.869e-02,  0.179]
================================================================================
```

Covariance estimator: robust





	Model	*Validation RMSPE*	*Validation RMSE*
0	*Mean Baseline*	*0.544414*	*0.110840*
1	*Random Walk Naive Forecasting*	*0.198058*	*NaN*
2	*GARCH(1,1), Constant Mean, Normal Dist*	*0.456065*	*0.322710*
3	*Analytical GJR-GARCH(1,1,1), Constant Mean, Sk...*	*0.480666*	*0.338645*

Conclusion:
Compared to GARCH(1,1) with Normal Distribution setting, GJR-GARCH with Skewed Student's T forecasts have moved up quite a bit and shortened the gap between target values and model's predictions. This is also reflected in an improvement in both Validation RMSPE and RMSE.

Unfortunately I still haven't been able to get close to Naive Forecasting.

TARCH:

There's another member in the GARCH family called TARCH, which is short for Threshold Autoregressive Conditional Heteroskedasticity (and also known as ZARCH). TARCH models the volatility using absolute values (instead of squares). This model is specified using power=1.0 since the default power, 2.0, corresponds to variance processes that evolve in squares. In addition, asymmetric impact is also incorporated into the GARCH framework by using a dummy variable

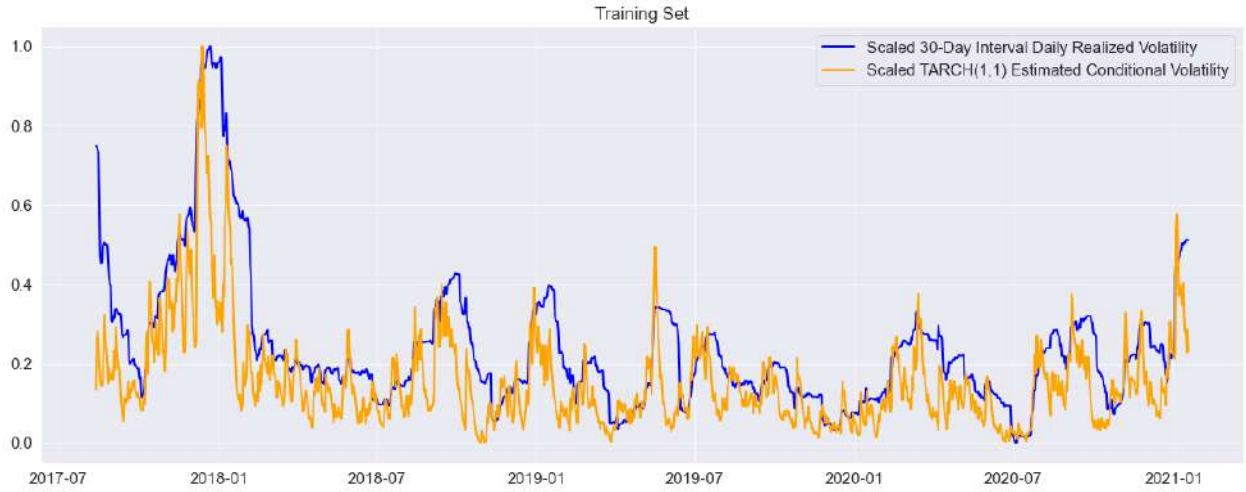
The volatility process in a TARCH(1,1) model is given by:

$$\boldsymbol{\sigma_t = \omega + \alpha|\epsilon_{t-1}| + \gamma|\epsilon_{t-1}|\mathbf{I}_{[\epsilon_{t-1}<0]} + \beta\sigma_{t-1}}$$

Constant Mean - TARCH/ZARCH Model Results
==========
Dep. Variable: returns R-squared: 0.000
Mean Model: Constant Mean Adj. R-squared: 0.000
Vol Model: TARCH/ZARCH Log-Likelihood: -3034.60
Distribution: Standardized Skew Student's t AIC: 6083.20
Method: Maximum Likelihood BIC: 6119.15
No. Observations: 1255
Date: Thu, Feb 24 2022 Df Residuals: 1254
Time: 13:38:34 Df Model: 1
Mean Model
==========
coef std err t P>|t| 95.0% Conf. Int.
---
mu -0.0291 7.032e-02 -0.414 0.679 [-0.167, 0.109]
Volatility Model
==========
coef std err t P>|t| 95.0% Conf. Int.

```
------------------------------------------------------------------------------
omega         0.2347     0.104     2.251  2.438e-02  [3.035e-02,  0.439]
alpha[1]      0.1889  4.443e-02     4.252  2.122e-05  [ 0.102,  0.276]
gamma[1]      0.0567  3.680e-02     1.541      0.123 [-1.543e-02,  0.129]
beta[1]       0.7827  5.716e-02    13.694  1.107e-42  [ 0.671,  0.895]
                          Distribution
==============================================================================
             coef    std err          t      P>|t|    95.0% Conf. Int.
------------------------------------------------------------------------------
eta          3.3688     0.294    11.469  1.881e-30  [ 2.793,  3.945]
lambda       0.1209  3.428e-02     3.525  4.231e-04 [5.366e-02,  0.188]
==============================================================================
```

Covariance estimator: robust
Log-likelihood and AIC/BIC are quite similar to the GJR-GARCH model. Next I'll inspect the estimated conditional volatility compared to the scaled vol_current.



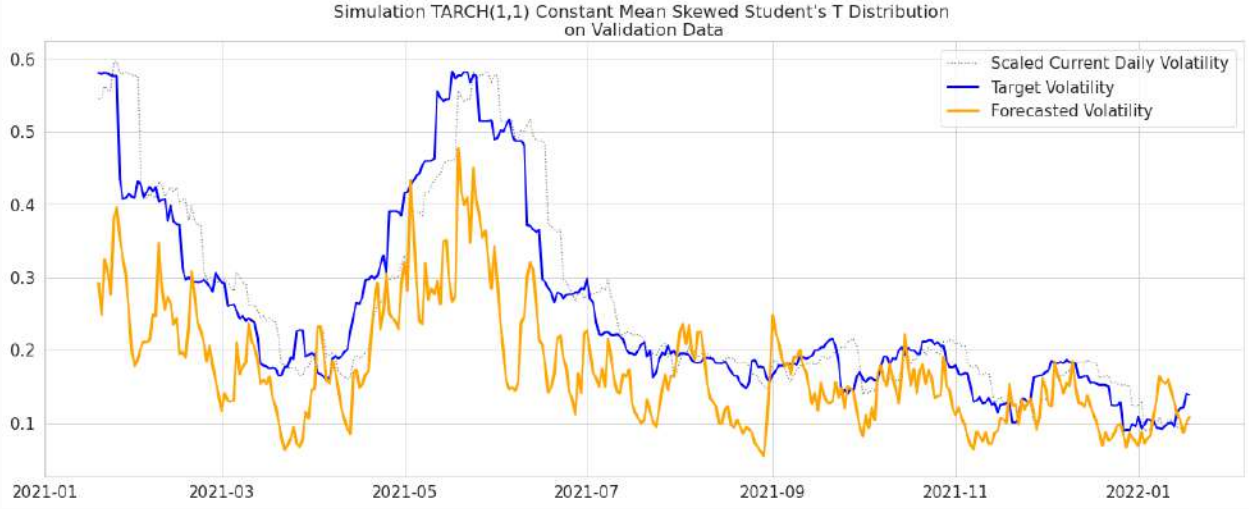
Simulation Forecasts:

Simulation-based forecasts use the model random number generator to simulate draws of the standardized residuals, e_{t+h}. These are used to generate a pre-specified number of paths of the variances which are then averaged to produce the forecasts. In models like GARCH which evolve in the squares of the residuals, there are few advantages to simulation-based forecasting. These methods are more valuable when producing multi-step forecasts from models that do not have closed form multi-step forecasts such as EGARCH models.
Assume there are B simulated paths. A single simulated path is generated using

$$\boldsymbol{\sigma^2_{t+h,b} = \omega + \alpha\epsilon^2_{t+h-1,b} + \beta\sigma^2_{t+h-1,b}}$$

$$\boldsymbol{\epsilon_{t+h,b} = e_{t+h,b}\sqrt{\sigma^2_{t+h,b}}}$$

where the simulated shocks are $e_{t+1,b}$, $e_{t+2,b}$, $\cdots$, $e_{t+h,b}$ where b is included to indicate that the simulations are independent across paths. Note that the first residual, ϵ_t, is in-sample and so is not simulated.



	Model	*Validation RMSPE*	*Validation RMSE*
0	*Mean Baseline*	*0.544414*	*0.110840*
1	*Random Walk Naive Forecasting*	*0.198058*	*NaN*
2	*GARCH(1,1), Constant Mean, Normal Dist*	*0.456065*	*0.322710*
3	*Analytical GJR-GARCH(1,1,1), Constant Mean, Sk...*	*0.480666*	*0.338645*
4	*Simulation TARCH(1,1), Constant Mean, Skewt Dist*	*0.351545*	*0.278526*

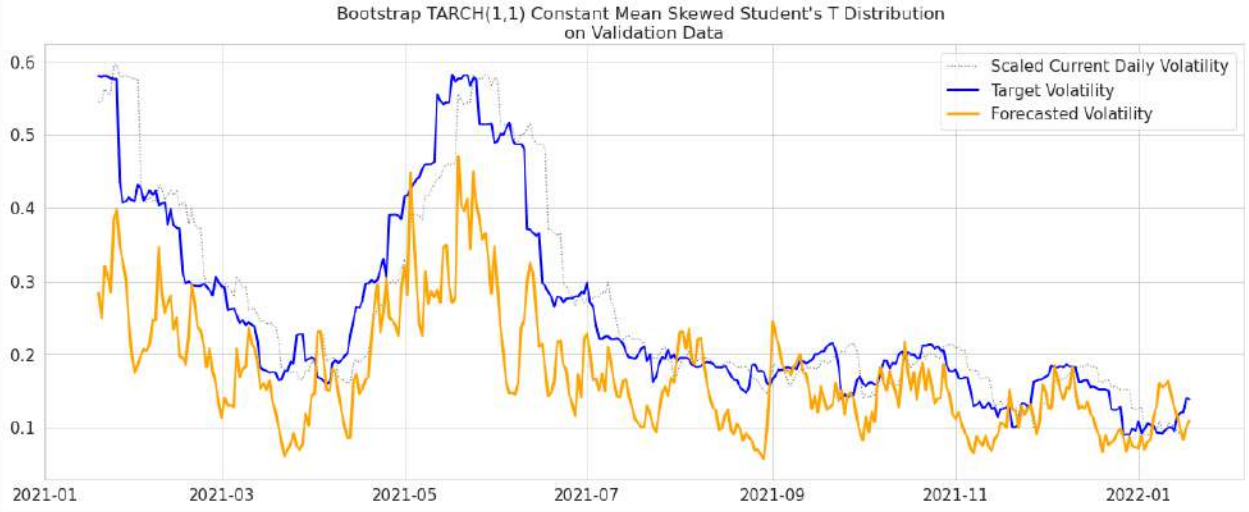
Bootstrap-based Forecasting

(Reference: https://arch.readthedocs.io/en/latest/univariate/univariate_volatility_forecasting.html#TARCH)

Bootstrap-based forecasts are virtually identical to simulation-based forecasts except that the standardized residuals are generated by the model. These standardized residuals are generated using the observed data and the estimated parameters as

$$\hat{e}_t = \frac{r_t - \hat{\mu}}{\hat{\sigma}_t}$$

The generation scheme is identical to the simulation-based method except that the simulated shocks are drawn (i.i.d., with replacement) from $\hat{e}_1, \hat{e}_2, \cdots, \hat{e}_t$. so that only data available at time t are used to simulate the paths.



	Model	Validation RMSPE	Validation RMSE
0	Mean Baseline	0.544414	0.110840
1	Random Walk Naive Forecasting	0.198058	NaN
2	GARCH(1,1), Constant Mean, Normal Dist	0.456065	0.322710
3	Analytical GJR-GARCH(1,1,1), Constant Mean, Sk...	0.480666	0.338645
4	Simulation TARCH(1,1), Constant Mean, Skewt Dist	0.351545	0.278526
5	Bootstrap TARCH(1,1), Constant Mean, Skewt Dist	0.352703	0.279399

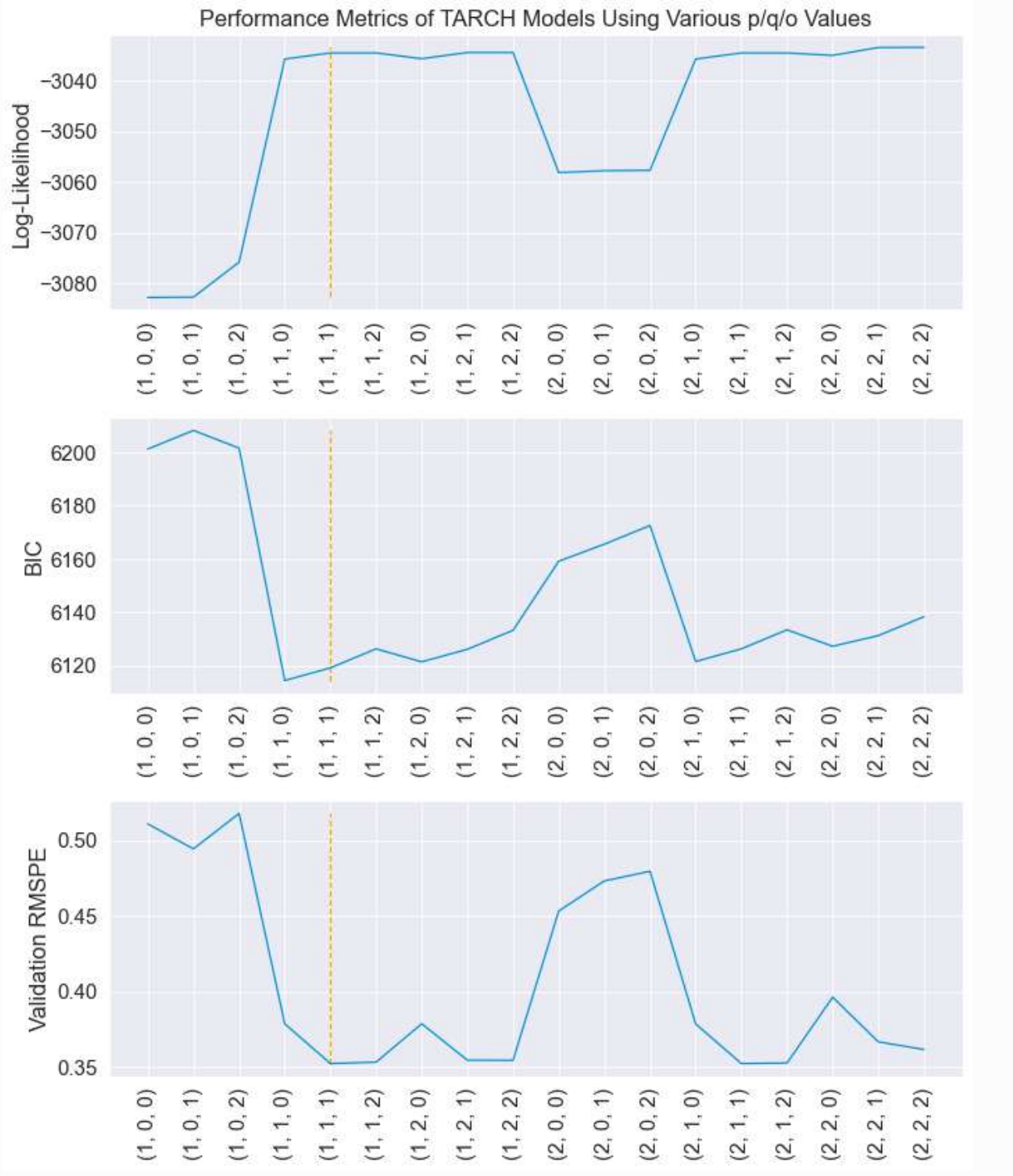
Conclusion :

Bootstrap TARCH(1,1) has managed to achieve a lower Validation RMSPE than Naive Forecasting!!! Even RMSE is only a bit higher, but so far this has been the best model among all GARCH family members that I've found.

Among the 2 forecasting methods, bootstrap seems to perform better on TARCH(1,1). It's probably because bootstrap uses historical data in to compute rather than using the assumed distribution of the residuals

Hyperparameter Tuning for TARCH

Next, I'll create a range of values for p, q and o from 0 to 3 (inclusive), and then record each's model's performance on unseen data using RMSPE. Since p cannot take a value of 0, I'll eliminate that from the list of permutations.



Constant Mean - TARCH/ZARCH Model Results

Dep. Variable:	returns	R-squared:	0.000
Mean Model:	Constant Mean	Adj. R-squared:	0.000
Vol Model:	TARCH/ZARCH	Log-Likelihood:	-3034.60
Distribution:	Standardized Skew Student's t	AIC:	6083.20
Method:	Maximum Likelihood	BIC:	6119.15
		No. Observations:	1255
Date:	Thu, Feb 24 2022	Df Residuals:	1254
Time:	13:59:57	Df Model:	1

Mean Model

	coef	std err	t	P>|t|	95.0% Conf. Int.
mu	-0.0291	7.032e-02	-0.414	0.679	[-0.167, 0.109]

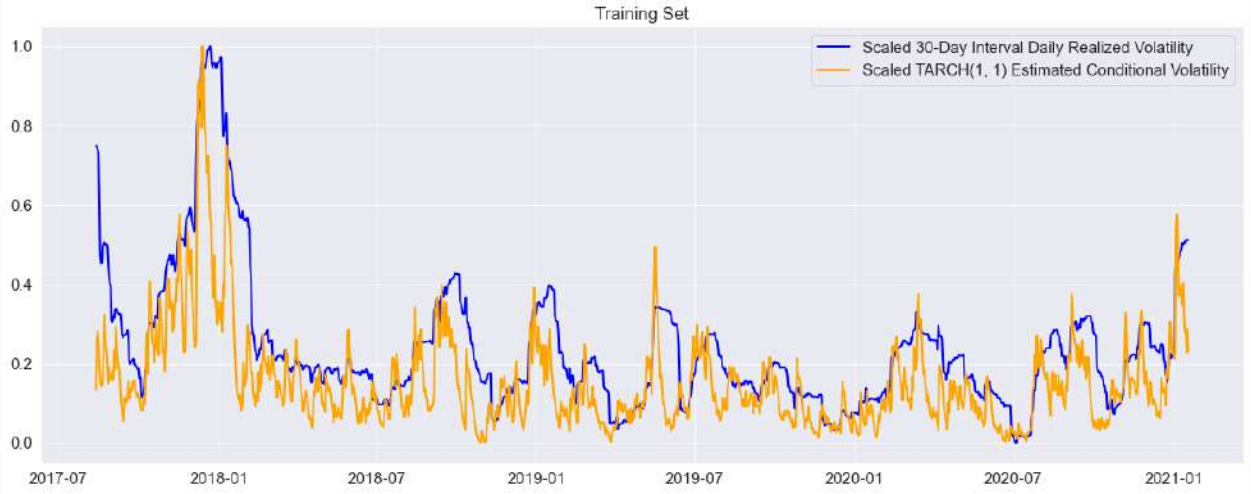
Volatility Model

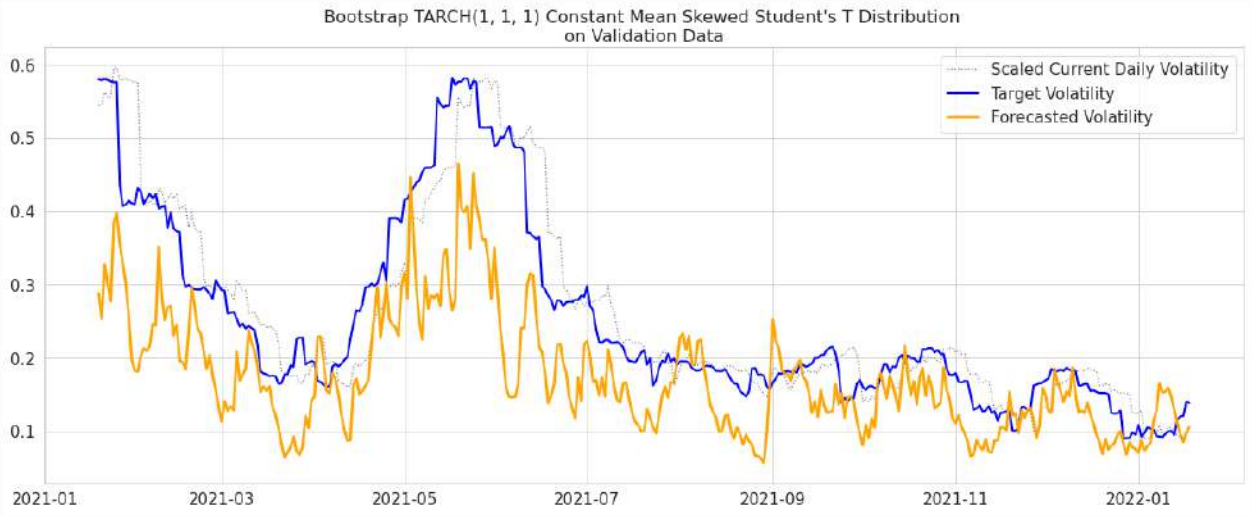
	coef	std err	t	P>|t|	95.0% Conf. Int.
omega	0.2347	0.104	2.251	2.438e-02	[3.035e-02, 0.439]
alpha[1]	0.1889	4.443e-02	4.252	2.122e-05	[0.102, 0.276]
gamma[1]	0.0567	3.680e-02	1.541	0.123	[-1.543e-02, 0.129]
beta[1]	0.7827	5.716e-02	13.694	1.107e-42	[0.671, 0.895]

Distribution

	coef	std err	t	P>|t|	95.0% Conf. Int.
eta	3.3688	0.294	11.469	1.881e-30	[2.793, 3.945]
lambda	0.1209	3.428e-02	3.525	4.231e-04	[5.366e-02, 0.188]

Covariance estimator: robust





Model		*Validation RMSPE*	*Validation RMSE*
0	*Mean Baseline*	*0.544414*	*0.110840*
1	*Random Walk Naive Forecasting*	*0.198058*	*NaN*
2	*GARCH(1,1), Constant Mean, Normal Dist*	*0.456065*	*0.322710*
3	*Analytical GJR-GARCH(1,1,1), Constant Mean, Sk...*	*0.480666*	*0.338645*
4	*Simulation TARCH(1,1), Constant Mean, Skewt Dist*	*0.351545*	*0.278526*
5	*Bootstrap TARCH(1,1), Constant Mean, Skewt Dist*	*0.352703*	*0.279399*
6	*Bootstrap TARCH(1, 1, 1), Constant Mean, Skewt...*	*0.352394*	*0.279253*

1-Layered LSTM RNN (20 units) with {n_past}-Day Lookback Window

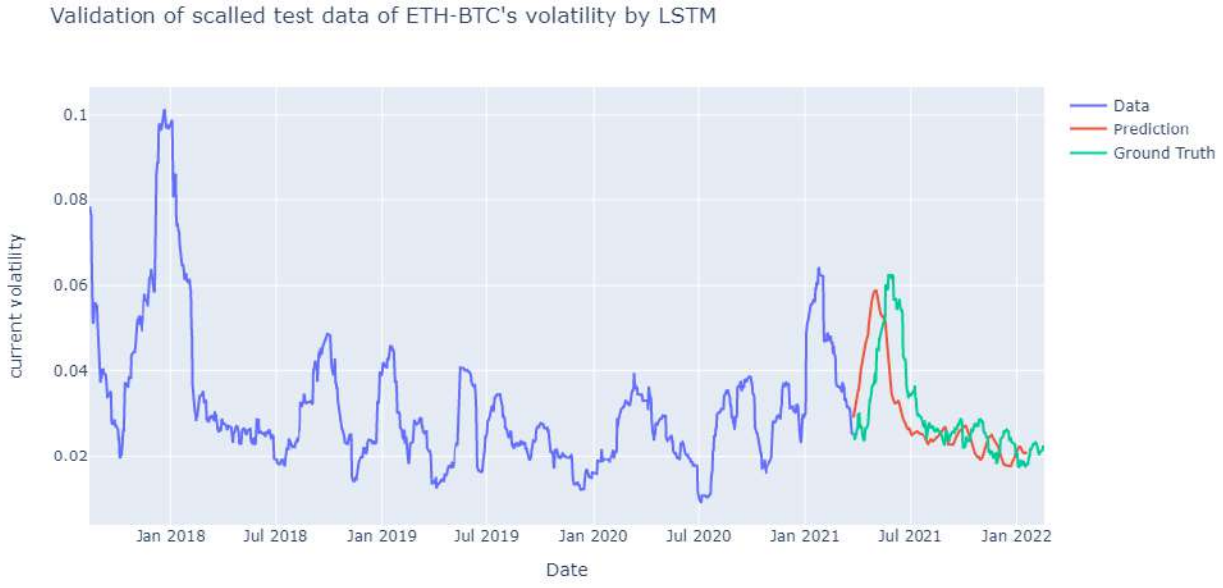
```
Epoch 1/25
65/65 [==============================] - 2s 13ms/step - loss: 1.1920e-04
Epoch 2/25
65/65 [==============================] - 1s 11ms/step - loss: 7.0685e-05
Epoch 3/25
65/65 [==============================] - 1s 11ms/step - loss: 6.1646e-05
Epoch 4/25
65/65 [==============================] - 1s 11ms/step - loss: 4.8064e-05
Epoch 5/25
65/65 [==============================] - 1s 11ms/step - loss: 3.5134e-05
Epoch 6/25
65/65 [==============================] - 1s 11ms/step - loss: 2.4411e-05
Epoch 7/25
65/65 [==============================] - 1s 12ms/step - loss: 2.3273e-05
Epoch 8/25
65/65 [==============================] - 1s 11ms/step - loss: 2.0404e-05
Epoch 9/25
65/65 [==============================] - 1s 12ms/step - loss: 2.2002e-05
Epoch 10/25
65/65 [==============================] - 1s 12ms/step - loss: 2.0569e-05
Epoch 11/25
65/65 [==============================] - 1s 12ms/step - loss: 1.7360e-05
Epoch 12/25
65/65 [==============================] - 1s 12ms/step - loss: 1.6364e-05
Epoch 13/25
65/65 [==============================] - 1s 11ms/step - loss: 2.2483e-05
Epoch 14/25
65/65 [==============================] - 1s 12ms/step - loss: 1.9562e-05
Epoch 15/25
65/65 [==============================] - 1s 13ms/step - loss: 1.7957e-05
Epoch 16/25
65/65 [==============================] - 1s 11ms/step - loss: 2.0270e-05
Epoch 17/25
65/65 [==============================] - 1s 12ms/step - loss: 1.8052e-05
Epoch 18/25
65/65 [==============================] - 1s 12ms/step - loss: 1.7122e-05
Epoch 19/25
65/65 [==============================] - 1s 13ms/step - loss: 1.4758e-05
Epoch 20/25
65/65 [==============================] - 1s 12ms/step - loss: 1.7883e-05
Epoch 21/25
```

```
65/65 [==============================] - 1s 12ms/step - loss: 1.7013e-05
Epoch 22/25
65/65 [==============================] - 1s 12ms/step - loss: 1.6245e-05
Epoch 23/25
65/65 [==============================] - 1s 12ms/step - loss: 1.3372e-05
Epoch 24/25
65/65 [==============================] - 1s 12ms/step - loss: 1.3996e-05
Epoch 25/25
65/65 [==============================] - 1s 12ms/step - loss: 1.5479e-05
```

Out[177]:

<keras.callbacks.History at 0x25a7de64e20>

Model		*Validation RMSPE*	*Validation RMSE*
0	*Mean Baseline*	*0.544414*	*0.110840*
1	*Random Walk Naive Forecasting*	*0.198058*	*NaN*
2	*GARCH(1,1), Constant Mean, Normal Dist*	*0.456065*	*0.322710*
3	*Analytical GJR-GARCH(1,1,1), Constant Mean, Sk...*	*0.480666*	*0.338645*
4	*Simulation TARCH(1,1), Constant Mean, Skewt Dist*	*0.351545*	*0.278526*
5	*Bootstrap TARCH(1,1), Constant Mean, Skewt Dist*	*0.352703*	*0.279399*
6	*Bootstrap TARCH(1, 1, 1), Constant Mean, Skewt...*	*0.352394*	*0.279253*
7	LSTM	0.112561	0.101456



The forecast is tracing the current volatility line very closely, but is mostly lagging behind compared to my target values.

Conclusion: Volatility forecast says it is decreases in next month

SUMMARY:

From Heatmap we select best pair of cryptocurrency ETH-BTC (Base currency-Quote Currency) for further analysis because both are top 2 Cryptocurrencies of Crypto market.

We fit MA on close price of ETH-BTC
when MA is too large it not capable to capturing some spike and dips and when MA is small it s again there is again noisy observation problem so there is compromise and we need to looki ng for some other methods like
AR,
ARMA,
ARIMA,
ACF,
EWMA

We check Stationarity of close price so Our Close prices is not stationary hence instead of Calculating volatility of close price we choose to formulate returns and then volatility of returns for further analysis.

But before that we forecast Close prices by LSTM and they are slightly decreasing.

we plot volume plot and then forecast it by LSTM so our forecast shows rate of change of decreasing volume is reduced, so its good indicator.

We calculate returns and then log returns For practicality purposes, it's generally preferable to use the log returns especially in mathematic modeling, because it helps eliminate non-stationary properties of time series data, and makes it more stable. There's another advantage to log returns, which is that they're additive across time:

Then we compare the distribution plots and stationarity plot of Returns and Log Returns and their distribution is normal and stationarity plots are also same.

Then we fit auto ARIMA model to log returns but there is high heteroskedasticity so instead of going to traditional time series model we have to choose time series models for High Volatility such as Baseline models and GARCH model

We fit Baseline models, GARCH model and LSTM machine learning model and compare them based on RMSPE & RMSE so our LSTM model have lowest RMSPE & RMSE values.

So we forecast volatility by LSTM model

Our LSTM forecast for Volatility are decreasing in future for next one month.

Overall conclusion is that its good time to tread in BTC Instead of ETH.

Close price forecast by LSTM shows that trend of future close price is decreasing hence Price of Bitcoin is increasing or price of Etherium is decreasing. So we can transfer our Etherium portfolio into Bitcoin.

References:

1. Géron, A. (2019). *In Hands-on machine learning with Scikit-Learn & TensorFlow: concepts, tools, and techniques to build intelligent systems.* O'Reilly Media, Inc.
2. Sinclair, E. (2020). *Positional option trading: An advanced guide.* John Wiley & Sons.
3. https://algotrading101.com/learn/yfinance-guide/
4. https://www.coursera.org/learn/tensorflow-sequences-time-series-and-prediction/supplement/DM4fi/convolutional-neural-networks-course
5. https://insights.deribit.com/options-course/
6. https://arch.readthedocs.io/en/latest/univariate/univariate_volatility_forecasting.html
7. https://www.investopedia.com/terms/v/vix.asp
8. https://www.hindawi.com/journals/complexity/2021/6647534/
9. https://github.com/ritvikmath/Time-Series-Analysis/blob/master/GARCH%20Stock%20Modeling.ipynb
10. https://github.com/ritvikmath/Time-Series-Analysis/blob/master/GARCH%20Model.ipynb
11. https://www.kaggle.com/c/optiver-realized-volatility-prediction
12. https://www.youtube.com/watch?v=NKHQiN-08S8
13. https://goldinlocks.github.io/ARCH_GARCH-Volatility-Forecasting/
14. https://towardsdatascience.com/time-series-analysis-on-multivariate-data-in-tensorflow-2f0591088502
15. https://deepmind.com/blog/article/wavenet-generative-model-raw-audio
16. https://github.com/philipperemy/keras-tcn
17. http://users.metu.edu.tr/ozancan/ARCHGARCHTutorial.html
18. https://towardsdatascience.com/8-commonly-used-pandas-display-options-you-should-know-a832365efa95

In [ ]:

DECLARATION:

We undersigned solemnly declare that the project report **Crypto-Pair Analysis** is based on my own work carried out during the course of our study under the supervision of **Mrs. Sarika Khirid**. I assert the statements made and conclusions drawn are an outcome of my research work. I further certify that

I. The work contained in the report is original and has been done by me under the general supervision of my supervisor **Mrs. Sarika Khirid**.

II. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.

III. We have followed the guidelines provided by the university in writing the report.

IV. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

Abhijeet Balasaheb Talole (2002877)

Aishwarya Sachin Mehendale(2002834)

Akshata Gangadhar Musale(2002971)