Cryptocurrencies Analysis

DS 1007 Project Presentation

Qiwen Zhang, Manli Zhao, Siyuan Sheng, Yijun Guo

Motivation

Cryptocurrencies worth more than $40 billion and are traded every day. A lucky few have become millions as a result of rapidly altering prices, while others have suffered devastating losses.

We have a dream to be a lucky few **millionaires** in this world! Despite the volatility and slumping price of the crypto market, we will see how building a model to predict the return can help us along this crazy journey!

Cryptocurrency Background

Cryptocurrency - a digital currency in which transactions are verified and records maintained by a decentralized system using cryptography, rather than by a centralized authority.

- Pros: cheaper and faster money transfers and decentralized systems that do not collapse at a single point of failure.
- Cons: price volatility, high energy consumption for mining activities, and use in criminal activities.

Here, we choose to analyze **Bitcoin**, since it is the world's first cryptocurrency. It has shown as steady a rise in value over the years on the market, now boasting 14 million Bitcoins in circulation.

Why choosing bitcoin?

Transaction Amount is the highest.

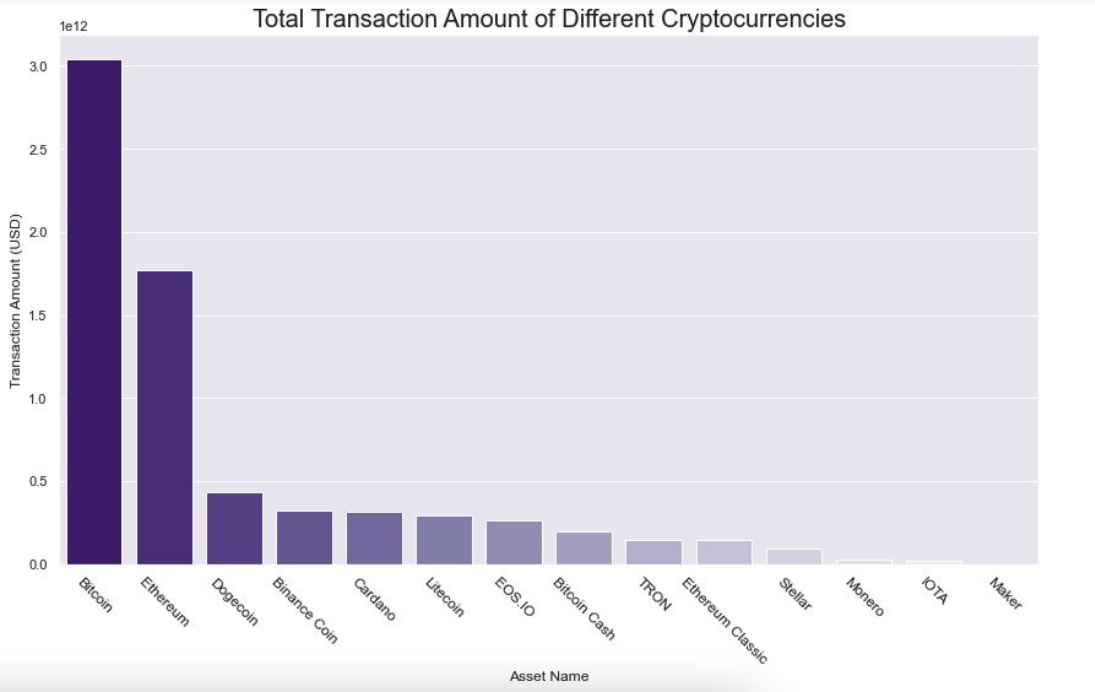
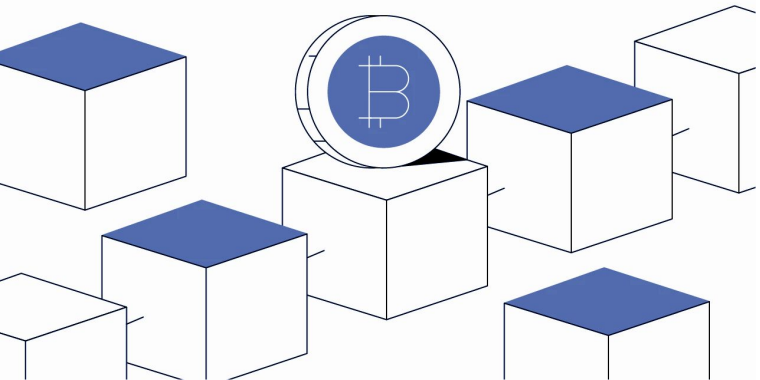


Diagram: Total transaction amount (USD) of cryptocurrencies from 2018 to 2021

The Bar plot shows Bitcoin and Ethereum are the two most popular cryptocurrency for trading or investment, with Bitcoin being the highest.

Task

- Analyze the multiple entries of the historic transaction data of crypto assets and use line plots to visualize the price trend.
- Test for seasonality and stationarity of historic transaction data.
- Train linear regression and random forest models to forecast 15-min future return of Bitcoin.



Data Overview

The dataset contains historical transaction record of various crypto assets, such as Bitcoin and Ethereum, from 12/31/2017 to 09/21/2021. There are total 24.2 million rows and 9 columns.

The columns are:

- **`Timestamp`** - A timestamp for the minute covered by the row.
- **`Asset_ID`** – ID code for the crypto asset.
- **`Count`** - The number of trades that took place this minute.
- **`Open`** - The price at the beginning of the minute.
- **`High`** - The highest price during the minute.
- **`Low`** - The lowest price during the minute.
- **`Close`** - The price at the end of the minute.
- **`Volume`** - The number of units traded during the minute.
- **`VWAP`** - The volume weighted average price for the minute.

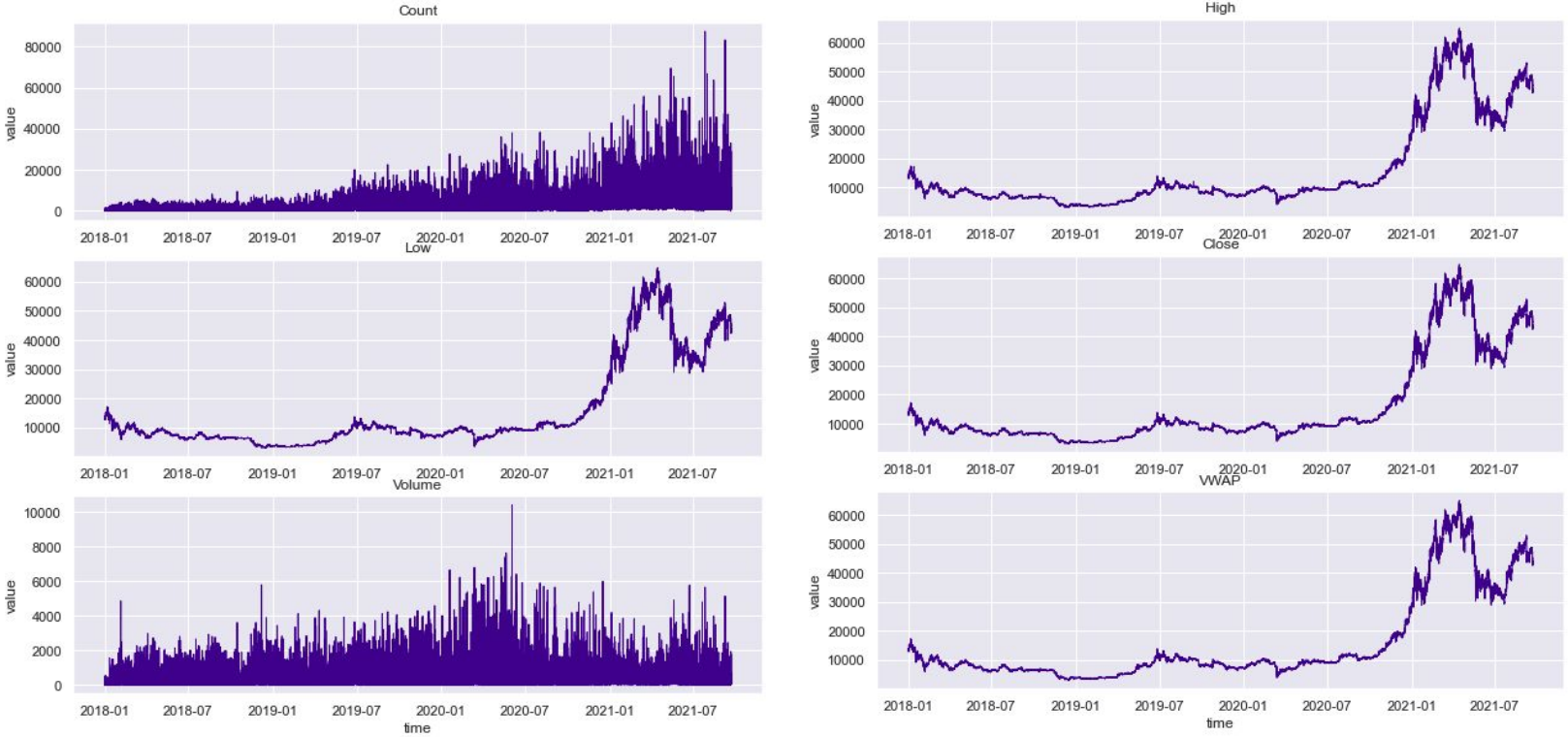
In our analysis, we focus on the best-known cryptocurrency, Bitcoin, which has taken up to 70% of the market transaction volume since 2021.

Dataset: G-Research Crypto Forecasting

- Since only Bitcoin is our interest, we select rows for Bitcoin from original dataset by its asset_ID.
- Changed unix timestamp to datetime index and save to a new dataframe.

timestamp	timestamp	Asset_ID	Count	Open	High	Low	Close	Volume	VWAP
2017-12-31 19:01:00	1514764860	1	229.0	13835.194	14013.8	13666.11	13850.176	31.550062	13827.062093
2017-12-31 19:02:00	1514764920	1	235.0	13835.036	14052.3	13680.00	13828.102	31.046432	13840.362591
2017-12-31 19:03:00	1514764980	1	528.0	13823.900	14000.4	13601.00	13801.314	55.061820	13806.068014
2017-12-31 19:04:00	1514765040	1	435.0	13802.512	13999.0	13576.28	13768.040	38.780529	13783.598101
2017-12-31 19:05:00	1514765100	1	742.0	13766.000	13955.9	13554.44	13724.914	108.501637	13735.586842

Line Plots for features in Bitcoin data from 2018 to 2021



Fill in data with gaps

Missing data: absence of rows

```
60      1956136
120          78
180          12
240          11
420           9
Name: timestamp, dtype: int64
```

```
#check timestamp difference if any missing
(bitcoin.index[1:]-bitcoin.index[:-1]).value_counts().head()
```

To fill the missing data: fill the gaps with previous valid value using .reindex() method

```
60    1956959
Name: timestamp, dtype: int64
```

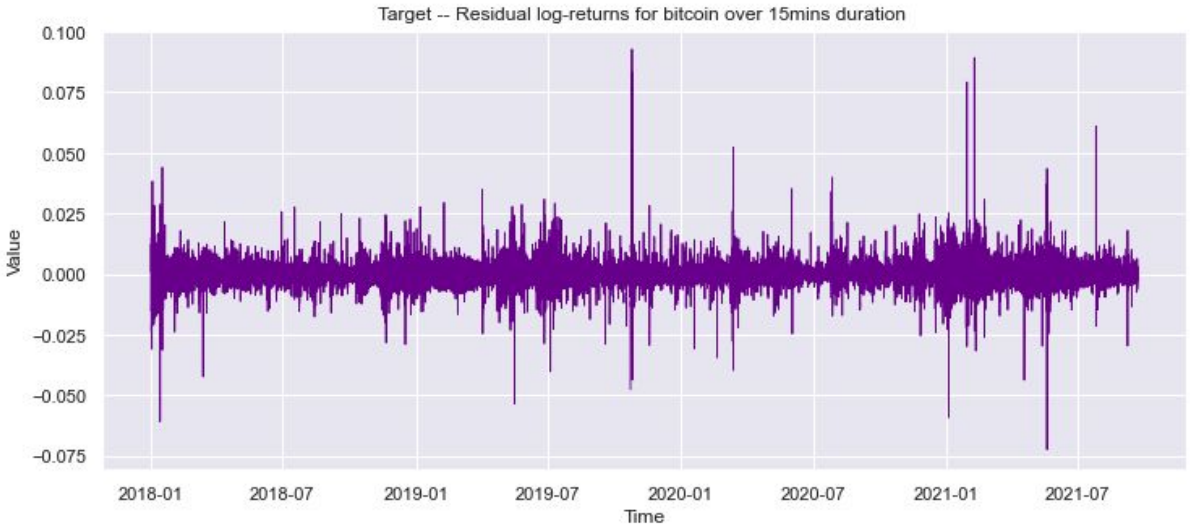
Analysis

Create the new Target Column based on close price

Target - Target is derived from Residual log returns (R) over 15 minutes.

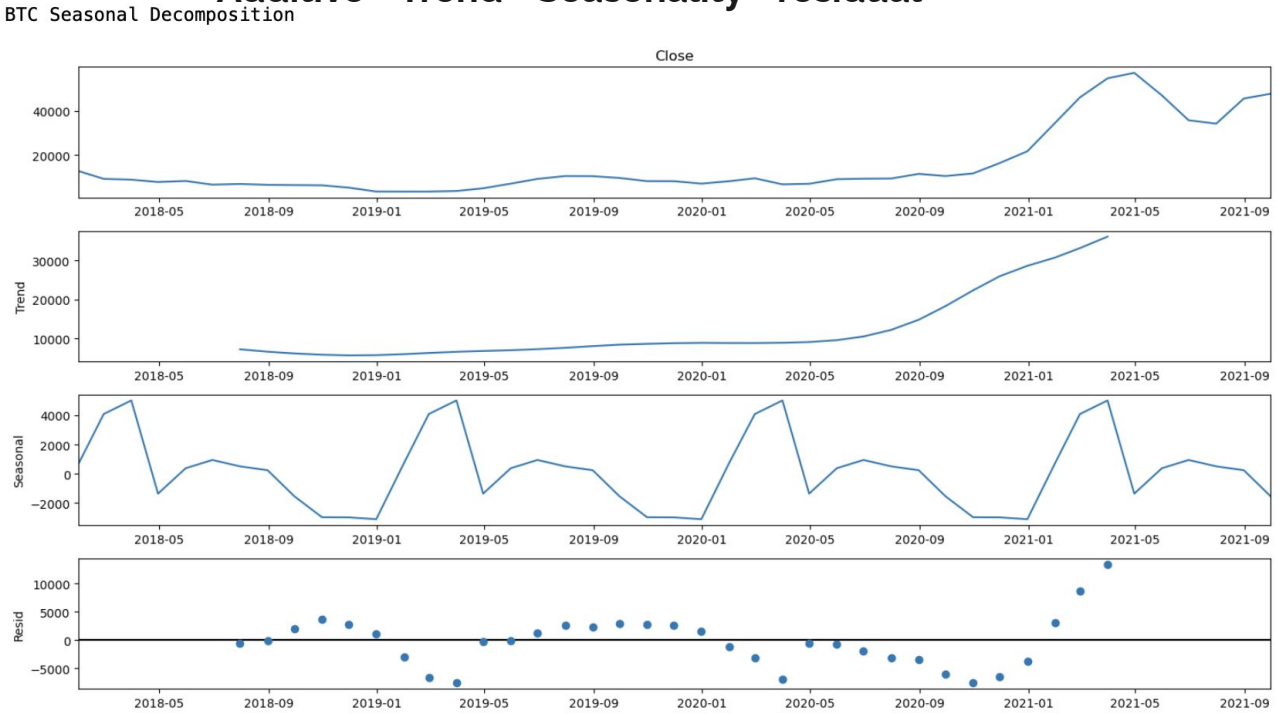
R(t) = log(P(t+15)/P(t))

Which P represent the close price for asset



Target values for Bitcoin from 2018-2021

Statistical model -Seasonal Decomposition
Additive = Trend + Seasonality + residual



Monthly average closed price

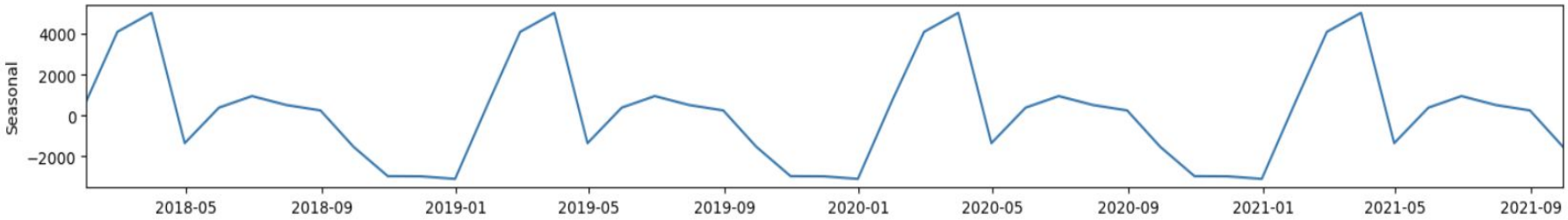
The general view of monthly average closed price

A repeated pattern due to seasonal fator

Excluded irregular component and analysis in the time series

A Statistical model, Seasonal Decomposition, was used to find a distinct, repeating pattern observed in regular intervals due to various seasonal factors within the period of one year

Choosing a training data size of all 12 months might be more helpful in identifying complete seasonal patterns.



Data Partitioning

Training: 12 months in 2020

```
btc2020= duration(start="01/01/2020", end="31/12/2020", data=bitcoin)
```

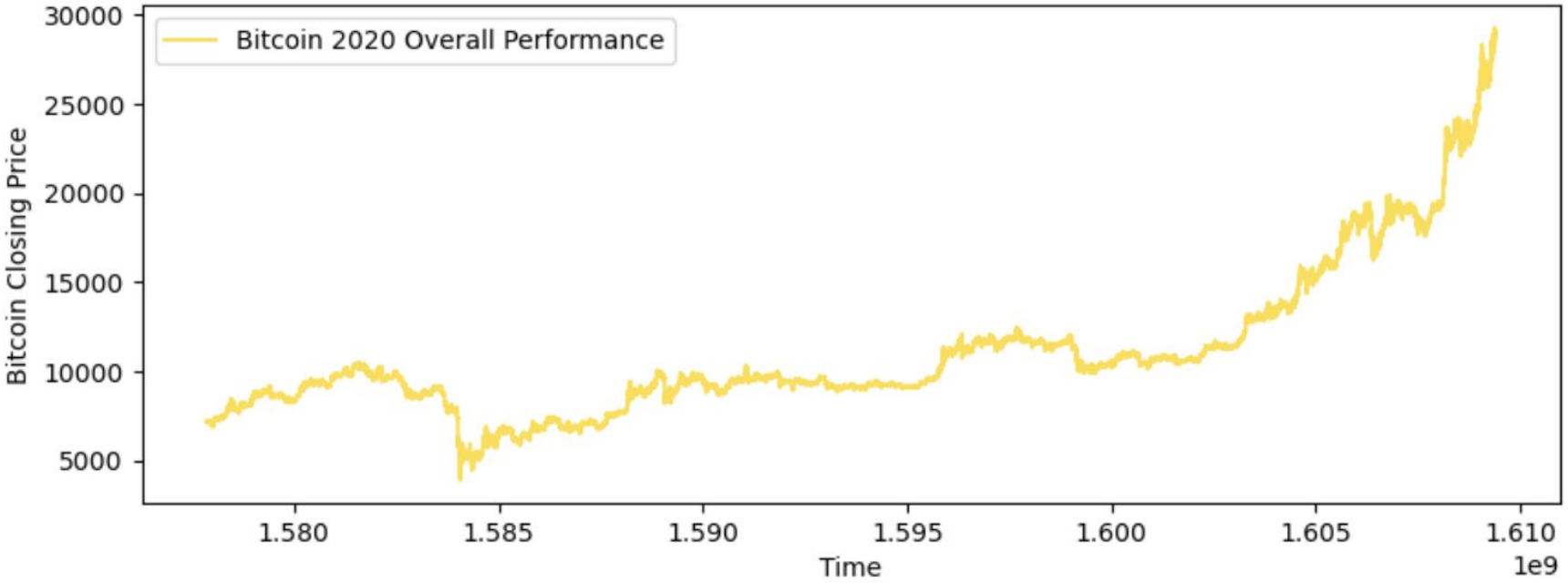
- Choose a whole year data which is closest to current time

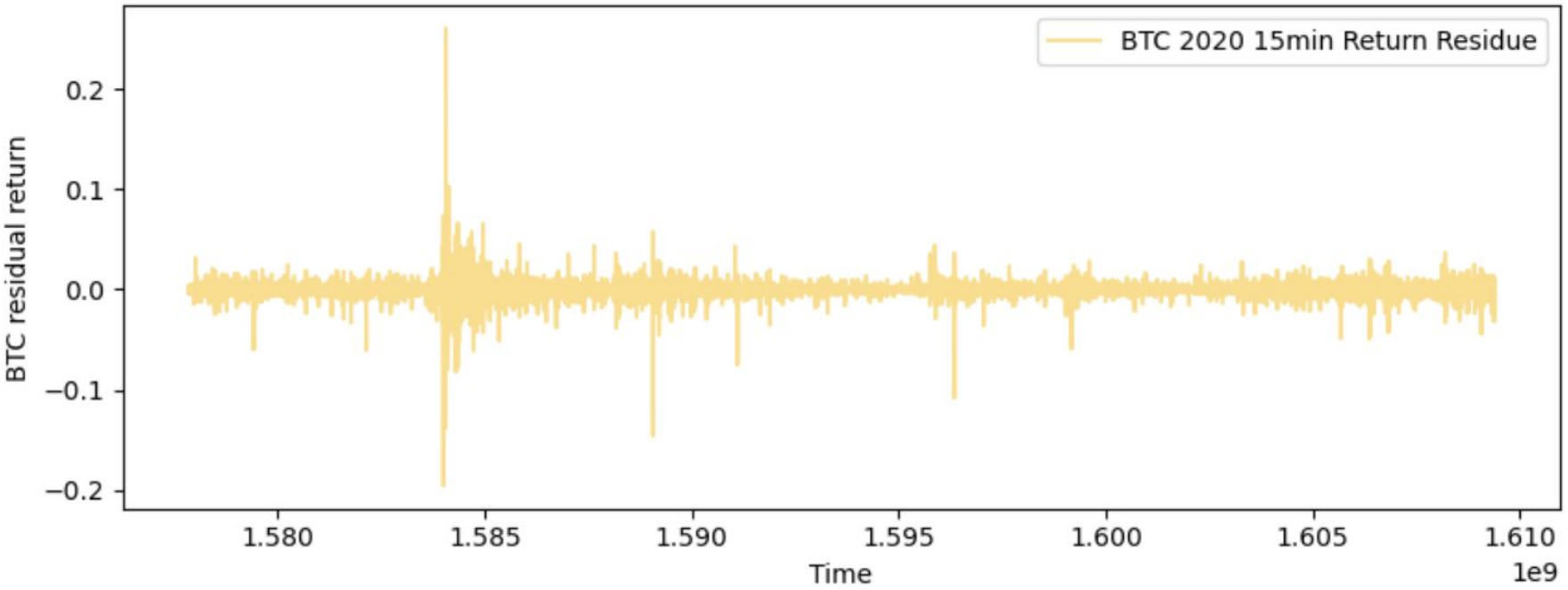
Testing: First 3 months in 2021

- With the pattern of the season time period, choosing to predict duration of three months right after the training data

Split into training and testing data set and output data sets to local path saved as .csv file.

Plotting the Close Price over Time (Bitcoin)





In 2020, the target value have fluctuate significantly between the interval +0.03 to -0.03 overall. Relatively more and greater volatility in the first half of the year

Predictive Model

We planned to train linear regression, random forest and ARIMA models.

Dickey-Fuller test was performed to test whether the data is stationary in order to use ARIMA. The test resulted in a p-value of 1 on the close price meaning that this set of data is not stationary. Hence, we did not train an ARIMA model on the data.

We trained a linear regression and a random forest model on the training set (all rows in year 2020) with default hyper-parameters and evaluated on both the training and testing test(all rows in the first 3 months in 2021).

Model Evaluation

It's observed that random forest model resulted in a much higher R^2 and lower RMSE compared to linear regression, meaning that much more variance in the target column was explained by the random forest model compared to linear regression on the training set.

Evaluation Dataset	Evaluation Metric	Linear Regression	Random Forest
Training	R^2	1.1e-3	0.88
	RMSE	1.6e-5	2.0e-6
Testing	R^2	-2.9e-4	-3.26
	RMSE	3.8e-5	1.6e-4

However, the R^2 evaluated on the test set are negative for both models, meaning that both models' predictions are worse than a constant function that always predicts the mean of the target. Especially, random forest resulted in a much worse R^2 and RMSE (10^4 and 10 times worse respectively). Based on the high R^2 obtained on the training data, we can reasonably conclude that random forest overfitted the training dataset and generalize poorly to unseen data.

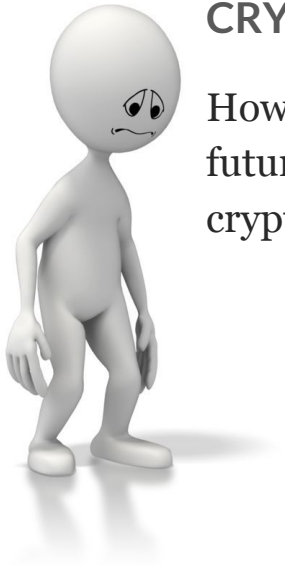
What could be Improved on Modeling?

First thing we could do is to try different hyperparameters. Since random forest model overfitted the training data, we could set appropriate values for hyperparameters such as max_depth, max_leaf_node, min_impurity_decrease to avoid building a too complicated tree model that would not generalize.

More models could be attempted such as LSTM and LightGBM. LSTM was trained sequentially with data in each timestamp and was able to predict future values based on previous, sequential data. For LightGBM, it's shown that a properly-tuned LightGBM will most likely win in terms of performance and speed compared with random forest[1], hence using lightGBM is also an option.

1. https://datascience.stackexchange.com/questions/63322/random-forest-vs-lightgbm

CONCLUSION



UNFORTUNATELY, WE ARE UNABLE TO BECOME MILLIONAIRE SINCE THE CRYPTO MARKET IS TOO VOLATILE!

However, besides re-training better models, we find out that one interesting direction of future investigation could be analyzing the correlation between different cryptocurrencies, which could potentially affect the prediction of Bitcoin return!!!