Business Cases for Data Science

MASTER'S DEGREE PROGRAM IN DATA SCIENCE AND ADVANCED ANALYTICS

4th Business Case

Forecasting Cryptocurrencies

Group W Members:

Diogo Bulhosa	m20210601
Mafalda Figueiredo	m20210591
Rodrigo Pimenta	m20210599
Francisco Costa	m20211022

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Table of Contents

Introduction *2*

Business Situation 2

Data Exploration *2*

Data Preparation *3*

Models *4*

Model Evaluation 5

Results 6

References *7*

Annex *8*

Figure 1 - Crypto Histograms 8

Figure 2- Correlation Heatmap of Log Returns 8

Figure 3 - Logarithm of the Highest Prices by Cryptocurrency 9

Figure 4 - Relative changes in closing prices 9

Figure 5 – Log Returns for the last 300 days 10

Figure 6 – Log Returns Gold for the last 500 days 10

Figure 7 – Log Returns Silver 11

Figure 8 – Log Returns Bronze for the last 1000 days 11

Figure 9 – Correlation heatmap for Bitcoin's feature selection 12

Table 1 – Feature selection for each cryptocurrency 13

Table 2 – Initial date for each cryptocurrency 13

Table 3 – Best parameters for each model, by cryptocurrency 14

Introduction

Forecasting is a technique used to predict future events based on historical data. For this project we used forecasting to predict some cryptocurrencies' future prices, applying statistical tools and some Machine Learning models, such as Random Forest, Neural Networks, Gradient Boost, and others that we will mention later.

The cryptocurrency market is booming, and it's only expected to get bigger. As the digital economy continues to grow at an enormous speed, cryptocurrencies will certainly have an important role to play in our future monetary system.

But what is a cryptocurrency after all? It is a digital and decentralized currency used in electronic payment transactions. There are no banks involved and zero to minimal transaction fees. Transactions are fast and not bound by geography and, like using cash, these are anonymous. Cryptocurrency received its name because it uses encryption to verify transactions in order to provide security and safety. Bitcoin was the first cryptocurrency created in 2009 and remains the best known today. In early 2010, Bitcoin's price was just a few cents. Over the next few years, new digital currencies entered the market and in January 2018, the total market cap for all cryptocurrencies reached $820 billion. The rise in the value and popularity of cryptocurrencies has raised speculation concerns. Like many other financial assets, the value of cryptocurrencies is very volatile, which means that investing in these assets is particularly risky, attracting mainly risk-loving investors.

Business Situation

Investments4Some is a long-standing Portuguese, privately held hedge funds management firm. Lately, the firm started to use Machine Learning models to forecast future market prices, unsuccessfully. Therefore, our team was asked to help the company do data-driven trading, building them a forecasting model using Machine Learning to predict the daily value of cryptocurrencies. To do this, we have been provided a dataset containing the daily prices (adjusted close, close, high, low, open and volume) of 10 cryptocurrencies (Cardano, Cosmos, Avalanche, Axie Infinity, Bitcoin, Ethereum, Chainlink, Terra, Polygon and Solana).

Data Exploration

To build our model we started by exploring the data that was provided. We did some visualizations to better understand the evolution of crypto prices and compare the various cryptos among themselves.

Plotting their histograms, we can see that all the crypto's distributions are right skewed because a natural limit prevents outcomes on one side (Figure 1). Additionally, we plotted a Pearson Correlation heatmap of the log returns, to see how much the price of a crypto influence the prices of others (Figure 2). The reason we used log returns instead of the raw price is because if the prices of different cryptocurrencies are not normalized, we can't use comparable metrics. Its possible to see that BTC and ETH were highly correlated (0.7775), followed by ADA and LINK that have a correlation with ETH of

0.6679 and 0.6479, respectively. This means that when ETH closing price increases, BTC, ADA and LINK follows. Nevertheless, this was just to have some insights from the data and we ended up not using it in our model predictions.

As shown in Figure 3, Bitcoin and Ethereum reached the higher prices, while Polygon and Cardano have the lowest prices. Based on this information, we divided the cryptos into 3 groups: gold, silver and bronze. Gold being the ones with the higher prices, bronze with the lowest prices and silver with the ones in between. We did this in order to have a better comparison of the different cryptos and their behaviour.

In Figure 4 it's possible to see the relative change of the price of the gold cryptos, where we used three different y-axis scales. We see that closing prices move in tandem, which means that when one coin closing price increases so does the other.

We also plot normalized changes of closing prices for last 300 days, as you can see in Figure 5 ,for all the cryptos together. Then in Figure 6 we can see the log returns for the gold cryptos on the last 500 days, in Figure 7 the log returns for the silver cryptos and finally in Figure 8 it's possible to see the bronze cryptos' log returns. Log differences can be interpreted as the percentage change.

Data Preparation

In this step we begun by creating new datasets, one for each coin, having the volume, the adjusted close, close, open, high and low prices as columns.

Proceeding to the creation of the coin individual datasets, we looked to use some financial indicators. After doing some research, we ended up producing 3 different Moving Average (Simple, Cumulative and Exponential) and observed its Convergence and Divergence (MACD). We also produced the Stochastic Oscillator, Bollinger Bands, Relative Strength Index, Average Directional Index and the Standard Deviation. Another two indicators we though that would be useful were the S&P500 value and the Dollar Selling Price. We thought this would be great to have information relative to things outside crypto.

When creating these datasets, there were some missing values, because some cryptos are older than others. Nevertheless, we kept those so we could compare datasets. However, when joining the indicators, we realized that S&P500 and Dollar Selling Price didn't trade on the weekends, so we decided to replace those missing values by the last Friday's price.

Afterwards, we applied MinMaxScaler to our dataset to do the feature selection, to exclude the variables that were giving redundant information. For each cryptocurrency we did a different feature selection, as we can see in Table 1, using a Spearman's correlation heatmap (Figure 9).

Additionally, for each crypto we did a scatter plot to see the evolution of the closing prices so we could choose the date window that we wanted to consider when applying our predictive model. For instance, in most of the cryptos, in the beginning of the timeseries, the closing price was much lower, so we decided to only consider when the prices were relatively similar to the present time. In Table 2 are the initial dates defined to predict each crypto future prices.

Models

To apply our models, we created three main and very important functions:

- **TimeBaseCV:** Since for this problem it is very important to take into consideration time when training our data models, TimeBasedCV allowed to make a Cross Validation that would not only move forward but also split the training datasets with information previous to the test sets. This worked in a sliding-window method and allowed the user to input the size of each train and test while observing the model.
- **Forecasting:** Once again, due to this being a time affected prediction, another problem surfaced: we could not use data from the specific date that we were testing. In essence, since the information about each day is only released on the next day, in order to predict the closing price, we have to carefully use information from past days. So for example, if we want to predict the values from next week, our model will have to be trained with the previous week, as this are the values that we will have available. This is what the function does, create a ready dataset and do a correct forecast with existing values.
- **ShowResults:** In order to make our work more efficient, we basically created this function that does almost everything. By passing the models to be tested, the data frame and also some other specifications like the days to test and others, the function will scale the data, pass it through both of the previously mentioned function, and in the end return a dataset with scores and also a very clear and interesting plot with the real price of the cryptocurrency compared to the predicted by the model.

To get our best forecasting model for each coin we tried four Machine learning models: Random Forest, Gradient Boost, XGradient Boost and Neural Networks. Below there's a brief description of how these models work.

Random Forest Regressor

Random Forest is a Supervised learning algorithm that uses the ensemble learning method. This method uses the combination of multiple predictions from a different set of ML algorithms to help with the accuracy of the main model. The basic block of RFs is a regression, which is a simple model based on the recursive partition of the space defined by the independent variables into smaller regions. In making a prediction, the tree is thus read from the first node (the root node); the successive tests are made; and successive branches are chosen until a terminal node (the leaf node) is reached, which defines the value to be predicted for the dependent variable (the forecast for the next return).

Gradient Boost

Gradient boosting like random forest uses the ensemble method. This model is most used in Tabular datasets because it can handle unwanted values better than most models, like outliers, missing values, among others.

This algorithm is an upgrade from the previous one. It has all the positives that its parent has and some improvements regarding the duration of the models training time.

XGradient Boost

This algorithm is an upgrade from the previous one. It has all of the positives that its parent has and some improvements regarding the duration of the models training time.

Neural Networks

Neural Networks was the only model we used that does not use ensemble method, instead it uses a multi-layer perception. This results in a training using backpropagation which uses the identity function as an activation function.

LSTM

This is an algorithm that artificial neural networks. This model differentiates from the previously Neural Networks by also having feedback connections. It is very commonly use in making predictions based on time series data, since there can be lags of unknown duration between important events in a time series.

Model Evaluation

In each model we did a tunning by hand, using the ShowResults function mentioned above, in order to get the best parameters suited for each coin (Table 3). To evaluate the models' performance we used four metrics:

1. The Mean absolute error (MAE), that represents the average of the absolute difference between the actual and predicted values in the dataset. The lowest this value is, the better (close to 0).

2. Mean Squared Error (MSE) represents the average of the squared difference between the original and predicted values in the data set. It measures the variance of the residuals. Like MAE, this metric should be as low as possible.

3. Root Mean Squared Error (RMSE) is the square root of Mean Squared error. It measures the standard deviation of residuals. The closest to zero this value is, the better.

4. The coefficient of determination or R-squared (R2) represents the proportion of the variance in the dependent variable that's explained by the linear regression model. The closest the R2 is to 1, the better. Although we must be aware that if this value is too high, the model may be overfitting.

Results

On the next table we have our results by coin and by model.

Model	Random Forest				Gradient Boost				XGBoost				Neural Networks			
Coin/Metric	MAE	MSE	RMSE	R2	MAE	MSE	RMSE	R2	MAE	MSE	RMSE	R2	MAE	MSE	RMSE	R2
ADA	0,011011	0,00015	0,012243	-0,337833	0,008044	0,000091	0,009539	0,187838	0,009325	0,000138	0,011751	-0,232544	0,008205	0,000095	0,009736	0,153949
ATOM	0,031834	0,001234	0,035122	-4,340504	0,014396	0,000302	0,017386	-0,308727	0,011005	0,000194	0,013939	0,158839	0,0132	0,000229	0,015128	0,009233
AVAX	0,023298	0,000571	0,023901	0,272391	0,022286	0,000632	0,025136	0,195247	0,018494	0,000567	0,023822	0,27722	0,019485	0,000499	0,022348	0,363877
AXS	0,012943	0,000189	0,013732	-3,149459	0,007583	0,000079	0,008902	-0,743605	0,013906	0,000254	0,015937	-4,588802	0,009555	0,000097	0,0009847	-1,133521
BTC	0,013372	0,000426	0,020647	-0,457271	0,01802	0,000721	0,026858	-1,465808	0,019366	0,000609	0,02467	-1,080458	0,015092	0,000279	0,016712	0,045325
ETH	0,019209	0,000382	0,019544	0,170431	0,021574	0,000512	0,022638	-0,113009	0,024232	0,000904	0,030059	-0,96234	0,01702	0,000375	0,019373	0,184888
LINK	0,014768	0,000229	0,015131	-0,031442	0,019506	0,00049	0,022133	-1,207017	0,012428	0,000298	0,017254	-0,341181	0,014875	0,000277	0,016655	-0,249668
LUNA1	0,086556	0,012635	0,112406	-0,705874	0,077163	0,012285	0,1108939	-0,658627	0,125584	0,017189	0,131105	-1,320643	0,077281	0,007644	0,087432	-0,032077
MATIC	0,018937	0,000615	0,024806	-0,924306	0,018422	0,000585	0,024187	-0,829538	0,018371	0,000707	0,026594	-1,211781	0,022777	0,000808	0,028428	-1,527288
SOL	0,42756	0,002312	0,048079	-7,922531	0,043685	0,002219	0,047104	-7,56425	0,04422	0,002228	0,047197	-7,598004	0,035175	0,001422	0,037713	-4,48967

As we can see there are some models that do not fit some of the crypto. Due to this we always tried to attribute the best model to each coin, considering the metrics on the table.

Based on the previous statement these were the models applied to each coin:

- ADA: Random Forest
- ATOM: XGBoost
- AVAX: XGBoost
- AXS: Random Forest
- BTC: Gradient Boost
- ETH: Random Forest
- LINK: XGBoost
- LUNA1: Gradient Boost
- MATIC: Gradient Boost
- SOL: XGBoost

Predictions ($)	
Coin	10/05/2021
ADA	0,78
ATOM	17,02
AVAX	51,72
AXS	30,4
BTC	35615,83
ETH	2637,5
LINK	10,86
LUNA1	70,53
MATIC	1,04
SOL	81,45

Conclusion

To conclude, we feel that our models could be more consistent. It is very hard to deal with such volatile data as the one present in the crypo market. Nevertheless, some of our models did a very decent job especially in training and with the previous data before the update. After researching why this could be happening we came to the conclusion that in the late period of times the market as been witnessing a bear trend, with all the coins falling aggressively down in price. Unfortunately, none of our models could predict this collapse, and even though some might have been close, we feel that more data from outside mere indicators could have helped. Technologies like text mining allow for this exploration and help predict moments like these, and this could bring huge value to the models (for example, Twitter or Reddit, two websites that are known to highly influence the market).

References

Brownlee, J. (2022). Extreme Gradient Boosting (XGBoost) Ensemble in Python. Retrieved 6 May 2022, from https://machinelearningmastery.com/extreme-gradient-boosting-ensemble-in-python/

Chugh, A. (2022). MAE, MSE, RMSE, Coefficient of Determination, Adjusted R Squared — Which Metric is Better?. Retrieved 9 May 2022, from https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e

Masui, T. (2022). All You Need to Know about Gradient Boosting Algorithm − Part 1. Regression. Retrieved 7 May 2022, from https://towardsdatascience.com/all-you-need-to-know-about-gradient-boosting-algorithm-part-1-regression-2520a34a502

1.17. Neural network models (supervised). (2022). Retrieved 9 May 2022, from https://scikit-learn.org/stable/modules/neural_networks_supervised.html

Orac, R. (2022). Cryptocurrency Analysis with Python — Buy and Hold. Retrieved 28 April 2022, from https://towardsdatascience.com/cryptocurrency-analysis-with-python-buy-and-hold-c3b0bc164ffa

Orac, R. (2022). Cryptocurrency Analysis with Python - Log Returns. Retrieved 29 April 2022, from https://romanorac.github.io/cryptocurrency/analysis/2017/12/29/cryptocurrency-analysis-with-python-part3.html

Random Forest Regression: A Complete Reference - AskPython. (2022). Retrieved 7 May 2022, from https://www.askpython.com/python/examples/random-forest-regression

Sebastião, H., Godinho, P. Forecasting and trading cryptocurrencies with machine learning under changing market conditions. *Financ Innov* **7,** 3 (2021). https://doi.org/10.1186/s40854-020-00217-x

What is cryptocurrency and how does it work?. (2022). Retrieved 5 May 2022, from https://www.kaspersky.com/resource-center/definitions/what-is-cryptocurrency

Annex

Figure 1 - Crypto Histograms

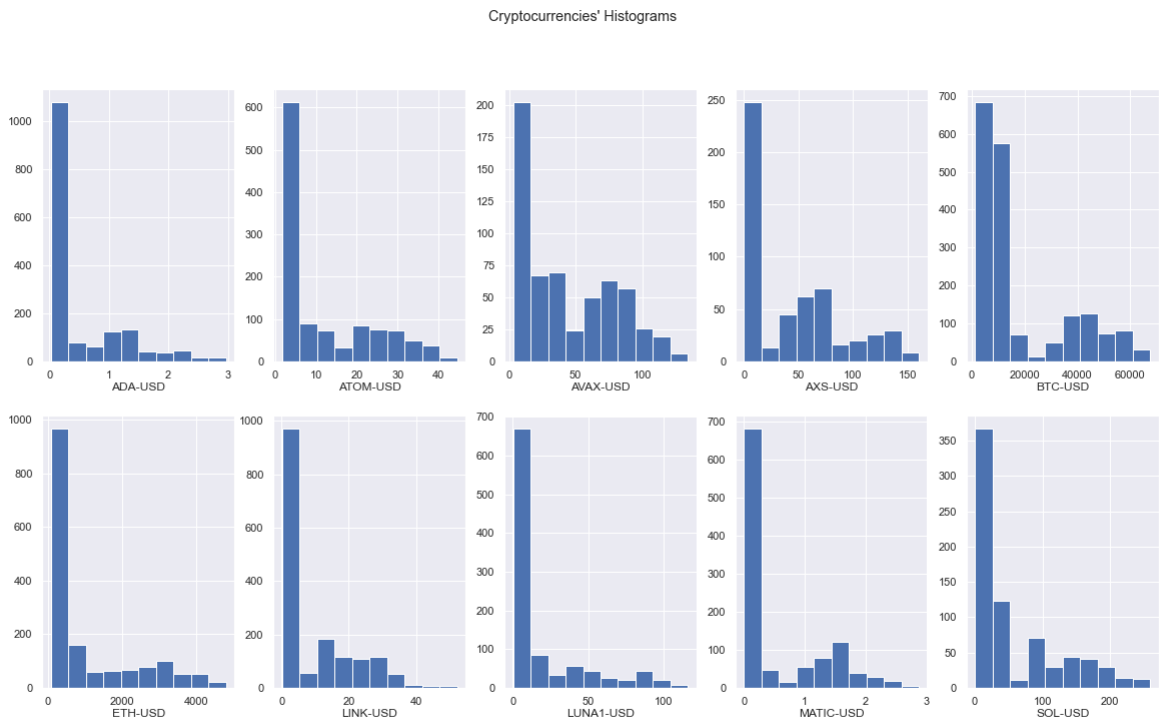

Figure 2- Correlation Heatmap of Log Returns

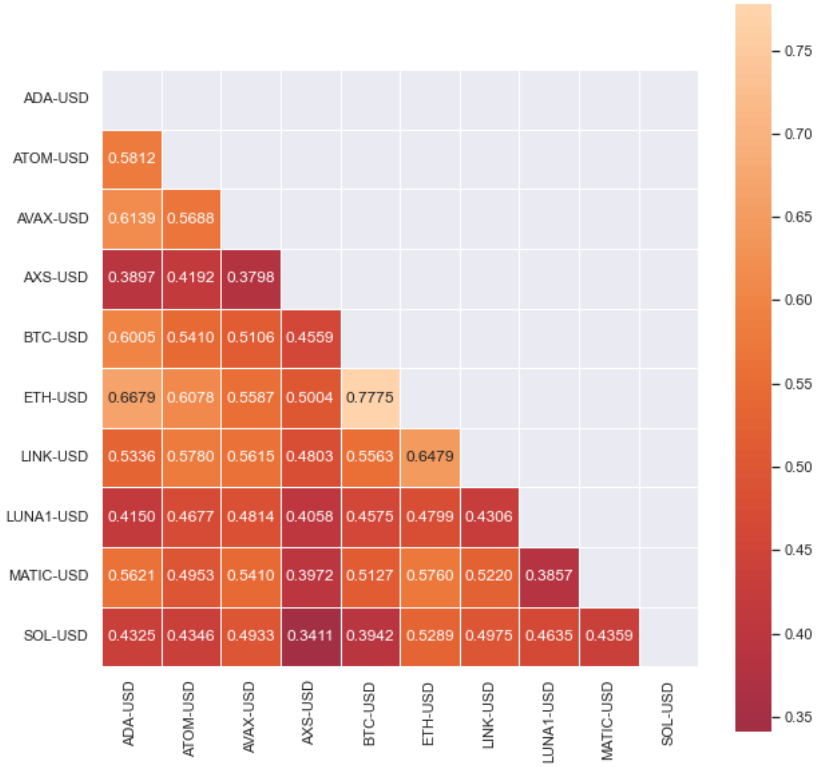

Figure 3 - Logarithm of the Highest Prices by Cryptocurrency

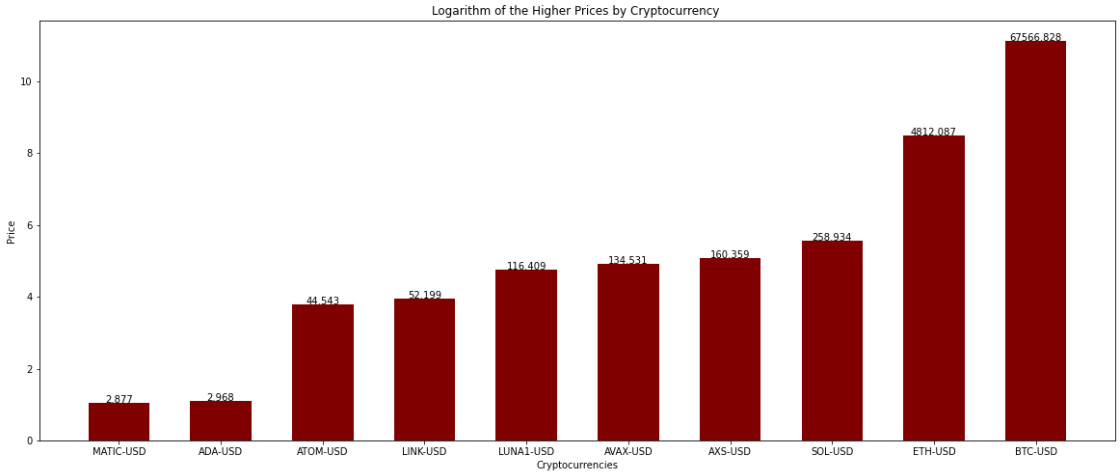

Figure 4 - Relative changes in closing prices

Figure 5 – Log Returns for the last 300 days

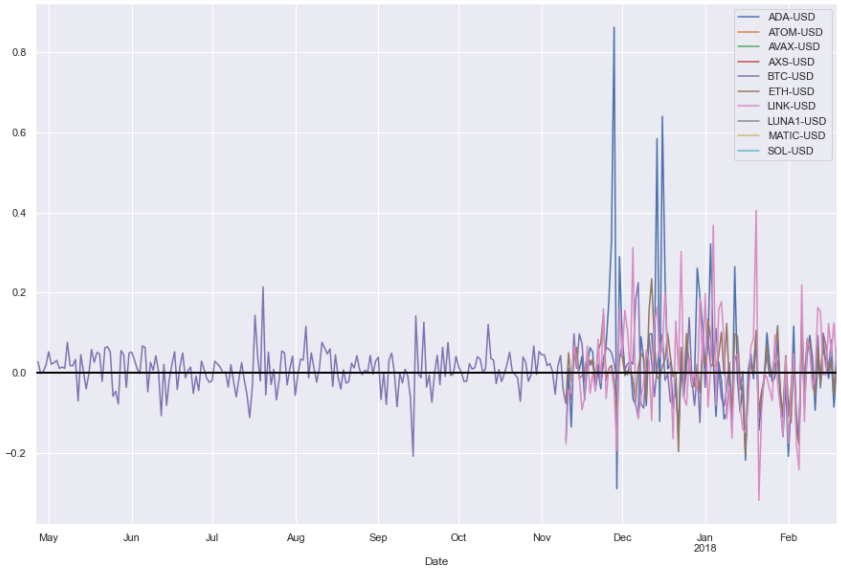

Figure 6 – Log Returns Gold for the last 500 days

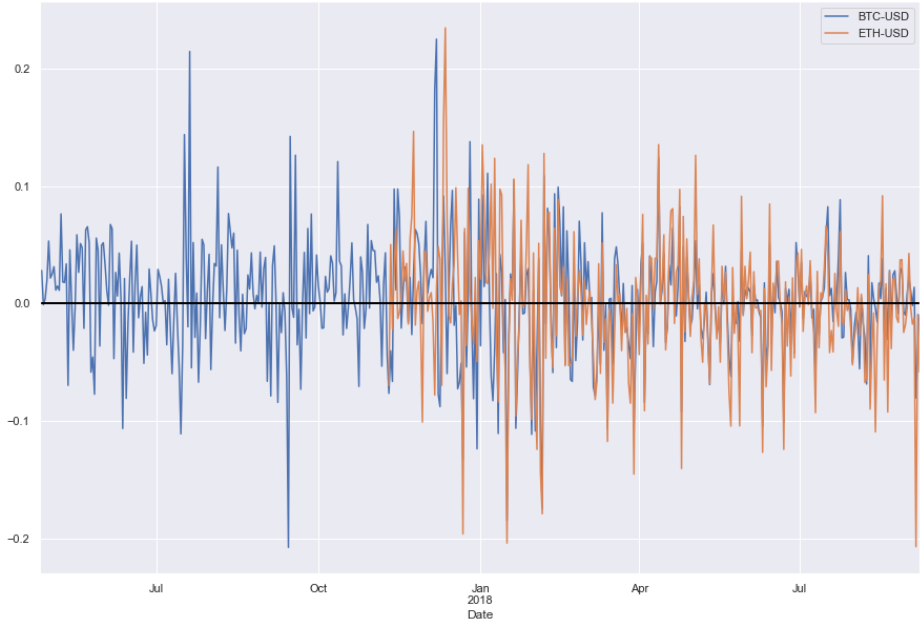

Figure 7 – Log Returns Silver

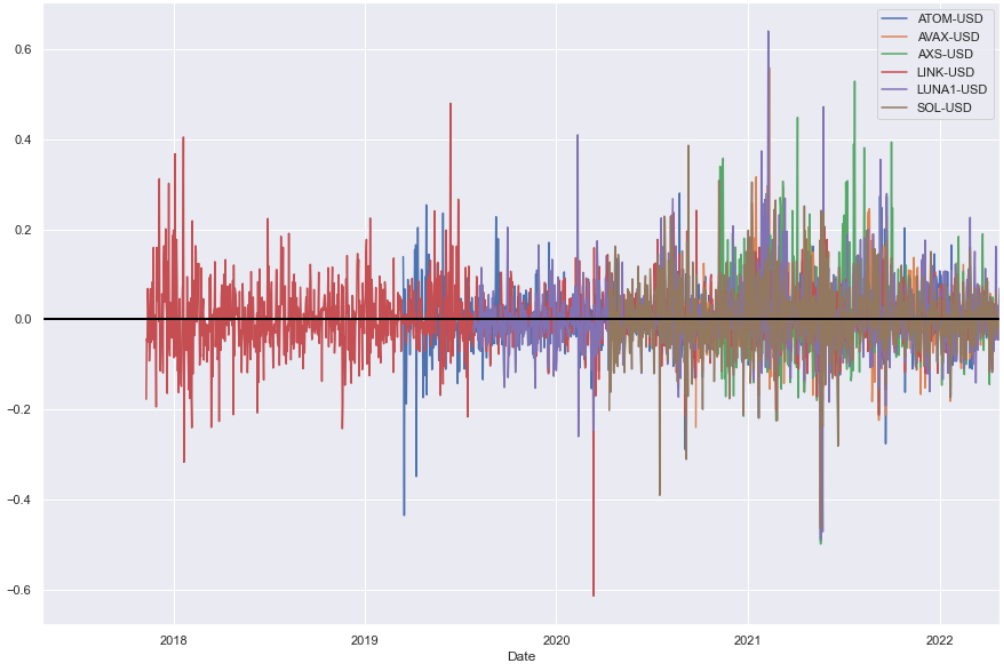

Figure 8 – Log Returns Bronze for the last 1000 days

Figure 9 – Correlation heatmap for Bitcoin's feature selection

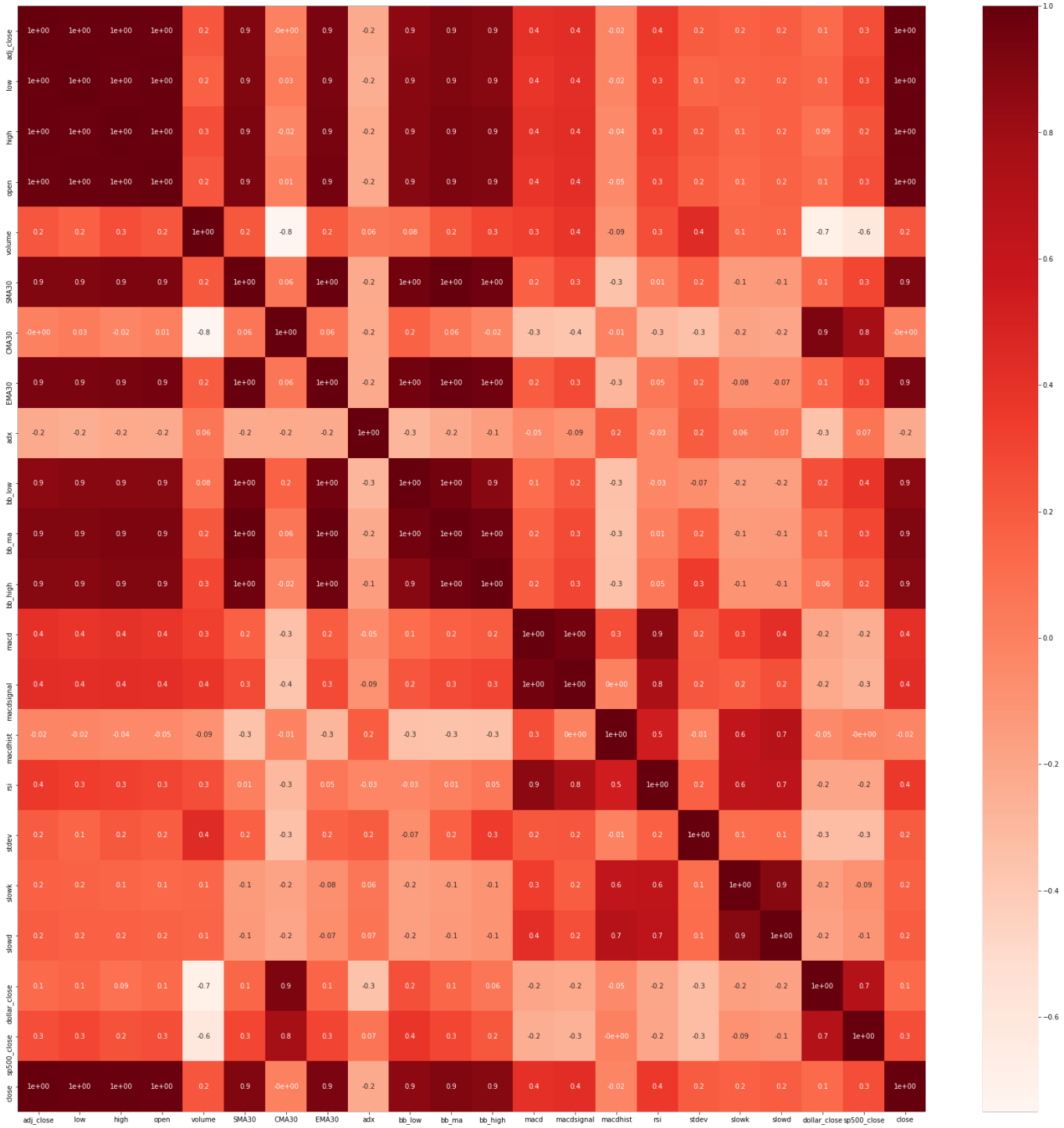

Table 1 – Feature selection for each cryptocurrency

	ADA	ATOM	AVAX	AXS	BTC	ETH	LINK	LUNA1	MATIC	SOL
adj_close	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗
low	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
high	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓
open	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
volume	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓
SMA30	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
CMA30	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗
EMA30	✗	✓	✗	✗	✗	✗	✗	✓	✓	✗
adx	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
bb_low	✓	✗	✓	✓	✗	✓	✓	✗	✗	✗
bb_ma	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
bb_high	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
macd	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗
macdsignal	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓
macdhist	✓	✗	✓	✓	✓	✓	✓	✗	✓	✓
rsi	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
stdev	✓	✓	✓	✓	✗	✓	✓	✗	✗	✓
slowk	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
dollar_close	✓	✗	✓	✓	✓	✗	✓	✗	✓	✗
sp500_close	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓

Table 2 – Initial date for each cryptocurrency

	ADA	ATOM	AVAX	AXS	BTC	ETH	LINK	LUNA1	MATIC	SOL
Starting Date	13/02/2021	11/01/2021	22/08/2021	10/08/2021	01/01/2021	14/04/2021	01/10/2021	01/11/2021	03/08/2021	20/09/2021

Table 3 – Best parameters for each model, by cryptocurrency

ADA	Random Forest	random_state=10,criterion='mae', max_depth=5, max_features='sqrt', n_estimators=150
	Gradient Boost	random_state = 10 , criterion='mse', n_estimators = 100, max_features= 'log2'
	Neural Networks	learning_rate = 'constant',solver = 'lbfgs', activation = 'identity',hidden_layer_sizes=(10,10,10,10), random_state=15,max_iter = 3000
	XGBoost	random_state = 10 , booster= 'gblinear', n_estimators = 130, validate_parameters = True,disable_default_eval_metric=False,eta = 0.3
ATOM	Random Forest	random_state=10,criterion='mae', max_depth=20, max_features='auto', n_estimators=100
	Gradient Boost	andom_state = 10 , criterion='mae', n_estimators = 150, max_features= 'auto'
	Neural Networks	learning_rate = 'adaptive',solver = 'lbfgs', activation = 'tanh',hidden_layer_sizes=(10), random_state=15,max_iter = 3000
	XGBoost	random_state = 10 , booster= 'gblinear', n_estimators = 130, validate_parameters = False,disable_default_eval_metric=True,eta = 0.3
AVAX	Random Forest	random_state=10,criterion='mse', max_depth=20, max_features='sqrt', n_estimators=50
	Gradient Boost	random_state = 10 , criterion='mse', n_estimators = 250, max_features= 'log2'
	Neural Networks	learning_rate = 'constant',solver = 'adam', activation = 'identity',hidden_layer_sizes=(10,10,10,10), random_state=15,max_iter = 3000
	XGBoost	random_state = 10 , booster= 'gblinear', n_estimators = 130, validate_parameters = True,disable_default_eval_metric=False,eta = 0.1
AXS	Random Forest	random_state=10,criterion='mse', max_depth=5, max_features='auto', n_estimators=100
	Gradient Boost	random_state = 10 , criterion='mae', n_estimators = 250, max_features= 'log2'
	Neural Networks	learning_rate = 'constant',solver = 'lbfgs', activation = 'tanh',hidden_layer_sizes=(10,10,10,10), random_state=15,max_iter = 3000
	XGBoost	random_state = 10 , booster= 'gblinear', n_estimators = 130, validate_parameters = False,disable_default_eval_metric=True,eta = 0.3
BTC	Random Forest	random_state=10,criterion='mae', max_depth=20, max_features='sqrt', n_estimators=150
	Gradient Boost	random_state = 10 , criterion='mse', n_estimators = 50, max_features= 'log2'
	Neural Networks	learning_rate = 'adaptive',solver = 'adam', activation = 'identity',hidden_layer_sizes=(10,10,10,10), random_state=15,max_iter = 3000
	XGBoost	random_state = 10 , booster= 'gblinear', n_estimators = 250, validate_parameters = False,disable_default_eval_metric=False,eta = 0.3
ETH	Random Forest	random_state=10,criterion='mae', max_depth=5, max_features='sqrt', n_estimators=300
	Gradient Boost	random_state = 10 , criterion='mse', n_estimators = 50, max_features= 'log2'
	Neural Networks	learning_rate = 'adaptive',solver = 'adam', activation = 'identity',hidden_layer_sizes=(10,10,10,10), random_state=15,max_iter = 3000
	XGBoost	random_state = 10 , booster= 'gblinear', n_estimators = 200, validate_parameters = False,disable_default_eval_metric=False,eta = 0.3
LINK	Random Forest	random_state=10,criterion='mse', max_depth=5, max_features='sqrt', n_estimators=10
	Gradient Boost	random_state = 10 , criterion='mae', n_estimators = 70, max_features= 'auto'
	Neural Networks	learning_rate = 'constant',solver = 'lbfgs', activation = 'tanh',hidden_layer_sizes=(20,20,20,20), random_state=15,max_iter = 3000
	XGBoost	random_state = 10 , booster= 'gblinear', n_estimators = 130, validate_parameters = True,disable_default_eval_metric=False,eta = 0.3
LUNA1	Random Forest	random_state=10,criterion='mse', max_depth=5, max_features='sqrt', n_estimators=250
	Gradient Boost	random_state = 10 , criterion='mse', n_estimators = 50, max_features= 'sqrt'
	Neural Networks	learning_rate = 'invscaling',solver = 'sgd', activation = 'relu',hidden_layer_sizes=(50,50), random_state=15,max_iter = 3000
	XGBoost	random_state = 10 , booster= 'gbtree', n_estimators = 130, validate_parameters = False,disable_default_eval_metric=True,eta = 0.5
MATIC	Random Forest	random_state=10,criterion='mae', max_depth=20, max_features='auto', n_estimators=50
	Gradient Boost	random_state = 10 , criterion='mse', n_estimators = 150, max_features= 'auto'
	Neural Networks	learning_rate = 'adaptive',solver = 'lbfgs', activation = 'relu',hidden_layer_sizes=(10,10,10,10), random_state=15,max_iter = 3000
	XGBoost	random_state = 10 , booster= 'gbtree', n_estimators = 130, validate_parameters = False,disable_default_eval_metric=True,eta = 0.05
SOL	Random Forest	random_state=10,criterion='mae', max_depth=5, max_features='auto', n_estimators=50
	Gradient Boost	random_state = 10 , criterion='mae', n_estimators = 50, max_features= 'auto'
	Neural Networks	learning_rate = 'adaptive',solver = 'lbfgs', activation = 'relu',hidden_layer_sizes=(10,10,10,10), random_state=15,max_iter = 3000
	XGBoost	random_state = 10 , booster= 'gblinear', n_estimators = 130, validate_parameters = False,disable_default_eval_metric=True,eta = 0.5