LUNDS UNIVERSITET

EKONOMI-HÖGSKOLAN

Master Essay – Finance Program

Inference and Prediction of Cryptocurrency Market Returns

Author: Emilia Alarcón
Supervisor: Anders Vilhelmsson
Date of the seminar: June 5th, 2020
Lund University School of Economics and Management
Lund, Sweden

Abstract. The potential for making profits investing in cryptocurrencies, the hedging benefits and the role in global economy, make it relevant to study the determinants of cryptocurrencies and to analyze different returns prediction models. Previous studies have focused one or some cryptocurrencies, this study analyzes the cryptocurrency market as a whole and finds the determinants of the cryptocurrency market and a returns prediction model using a machine learning approach. Evaluating the immediate impact of features, divided in cryptocurrency market data, information demand, financial markets, exchange rates and macroeconomics, it was found that the most important determinants of the cryptocurrency market returns is the cryptocurrency market data. For prediction of the next-day returns, the USD-CNY exchange rate emerged as the most important determinant. Different returns prediction models are evaluated using Lasso, Regression Tress, Random Forest and Boosting. Random Forest presents the best prediction accuracy and can be used to predict the cryptocurrency market returns.

Keywords: Cryptocurrency Market, Machine Learning, Prediction, Inference, Determinants

Contents

1. Introduction

Cryptocurrencies are a specific type of virtual currencies that use cryptography to protect financial transactions (Vejacka, 2014). As many discussions about their role in economy have been taking place in the last years, they can be considered one of the most controversial innovations in modern economy. Being decentralized, that is, existing outside the control of governments and central authorities, they are not restricted to any local jurisdiction or fiat currency (Brookins, Rinaldo & Zhao, 2019). Thus, cryptocurrencies seem to be detached from macroeconomic features and behave independently from other financial markets.

Low transaction cost, anonymous exchange and easy access, have created worldwide interest in cryptocurrencies. The importance of cryptocurrencies is undeniable. As Facebook develops their Libra cryptocurrency, China´s pilot program for the upcoming digital yuan has been taking place in Xiong'an (Cheng, 2020), representing an initiative that can change the way Central Banks manage money. Cryptocurrencies have reached the mainstream and this trend is expected to grow the next years (Dempere, 2019).

As their popularity rises, the number of cryptocurrencies in the market increases existing about 2 817 different cryptocurrencies at the time of collecting data for this study. Despite fluctuations, their prices have been going up since 2016 keeping the interest very substantial. Many cryptocurrency users think of them as assets rather than currencies and use them as an alternative investment to stocks and funds (Parashar & Rariwalla, 2019). It is considered, see for example Sun, Liu and Sima (2020), that cryptocurrencies can reduce portfolio risk by offering diversification and hedging benefits given their low correlation with other financial assets.

Thus, the potential for making profits investing in cryptocurrencies stimulates the development of methods and models for predicting prices which become very relevant for financial analysts, investors, traders and motivates this study. Moreover, the unique features of cryptocurrencies and their controversial role, not only in global economy but also in the digital future, make it essential to study their behavior, features and returns.

Recent studies about cryptocurrency price and returns prediction have focused on one or some cryptocurrencies and analyzed one currency at a time. For instance, Alahmari (2019) and Bezkorovainyi et al. (2019) predicts and forecast the price for Bitcoin, Ethereum and Ripple establishing similar but independent models for each. Brookins, Rinaldo and Zhao (2019) study returns prediction of Bitcoin, Ethereum and Litecoin to determine the method with best classification accuracy and use it to create trading strategies. Gomez-Espinosa, Valdés-Aguirre and Valencia (2019) focuses on predicting price direction of Bitcoin, Ethereum, Ripple and Litecoin using social and market data. Dempere (2019) analyzes the predictive power of selected financial variables over Bitcoin, Ethereum and Ripple. The results of these studies are important because they examine the cryptocurrencies with largest market. However, as most investors will invest in different cryptocurrencies and as the number of other cryptocurrencies available increases, a more comprehensive research approach about the whole cryptocurrency market is needed.

This study contributes to previously mentioned literature by studying the market as a whole and not analyzing one or some cryptocurrencies individually. The purposes are to find the determinants of the cryptocurrency market returns and their relationship (inference) and to determine a returns prediction model for the cryptocurrency market (prediction). Thus, the results from the present study can help to understand if the market behaves independently from other financial assets or macroeconomic features and find an approach to predict the cryptocurrency market returns. For this, machine learning techniques, that have gained a lot of attention for their forecast efficiency in the recent years will be used. Thus, while finding the best prediction model, new technology will be analyzed.

The study will be structured as follows: In Section 2 previous literature is analyzed. The data used in the study is presented in Section 3 followed by the Methodology in Section 4. The results of the determinants of the cryptocurrency market and the returns prediction models are presented in Section 5 leading to conclusions in Section 6.

2. Literature Review

The present study stablishes on the two main concepts of statistical learning. Hastie et al. (2013) define statistical learning as the approaches for estimating a function $\hat{f}$ that connects an output with one or more inputs in order to make inference about their relation or predict the output. Then according to them, the objectives for estimating a function are inference and prediction, where inference refers to the relationship between the inputs and the output (explanatory power) and prediction refers to using available data in the inputs to predict an output that is not available yet (predictive power). Thus, these two concepts, inference and prediction, match the purposes of this study and are used for its development. Inference relates to finding the determinants of the cryptocurrency market returns and prediction to determine a returns prediction model.

This section is divided in tree parts. Firstly, theory about inference is presented as the first step of the study will be to find the determinants of the cryptocurrency market returns. Then prediction and methods used for prediction are described. Finally, previous literature about cryptocurrency price and returns prediction is presented.

2.1 Inference

As mentioned and stated by Hastie et al. (2013), given a quantitative response Y of predictors X and the irreducible error term ϵ (equation 1), inference refers to estimating a function $\hat{f}$ in order to explain how changes in the input $X = \{X_1, X_2 ... X_P\}$ affects the response variable Y, which predictors are associated with the response and what is the relationship between the response and the predictors.

$$\mathrm{Y} = f\ (\mathrm{X}) + \epsilon \tag{1}$$

When determining a model, some predictors might not be associated with the response causing unnecessary complexity in the model (Hastie et al. 2013). Thus, in order to determine a model for cryptocurrency returns, a variable selection approach which penalize or remove unrelated features is needed. Hastie et al. (2013) describe approaches that could be applied to perform this task and divide them in three groups: Subset Selection, Shrinkage and Dimension Reduction. In this case, the more suitable are the Shrinkage approaches, where Lasso is the most appropiate since some variables could be irrelevant and thus removing them from the model by setting their respective coefficient estimates to zero becomes relevant. Hastie et al. (2013) emphasize that selecting a good value of the tuning parameter λ is critical for Lasso since setting λ to zero produces a least square fit and a large value of λ sets all the coefficient estimates to zero. Thus, depending in the value of λ different models are produced by Lasso. The authors suggest using Cross-validation to define the appropriate value of λ, that is, the value corresponding to the smallest Cross-validation error.

Lasso involves a linear relationship between the predictors and the response, which by removing irrelevant variables contributes to better inference, but may not produce accurate predictions as other non-linear models (Dalalyan, Hebiri & Lederer, 2017).

2.2 Prediction

Hastie et al. (2013) state that highly non-linear approaches can yield more accurate predictions because when prediction is the objective the exact form of the estimated function $\hat{f}$ is not important as the predictions $\hat{Y}$ obtained.

Following, some of the most popular and widely applicable non-linear approaches are presented.

2.2.1 Regression Trees

Breiman et al. (1984) remark that tree-based methods add a flexible non-parametrical approach to the data. Regression trees are more efficient than classical methods, like linear regressions, when the relationship between the predictor and the response variable is complex and non-linear (Hastie et al. 2013).

Cutler, Cutler and Stevens (2008) explain Regression Trees with the tree analogy as follows. The "root node", containing the mean value of the response variable of all the observations, is divided into two nodes based on a predictor variable. This split creates two new nodes where the observations are divided according to the value in the predictor, that is, if the value is smaller than the split-point the observations go to the left and otherwise go to the right. The process continues until non-partitioned nodes, called "terminal nodes", are reached. The value in each terminal node corresponds to the mean value of the response variable for that terminal node (Breiman et al. 1984). Further, Hastie et al. (2013) describe the terminal nodes as "leaves" or regions, implying that the predictor space is split in several simple regions R_j, and that the segments connecting the nodes and the leaves are known as "branches".

The Regression Trees use a top-down and greedy approach as stated by Hastie et al. (2013). The authors explain the top-down consideration noting that each split creates two new brunches and greedy because in each node the best split is made until some stopping criterion is met resulting in a tree with the lowest Residual Sum of Squares (RSS). Regarding prediction, Hastie et al. (2013) explain that once the regions R_j or the terminal nodes are defined by the model, the predicted response for a test observation which correspond to some terminal node, is the mean value of the training observations in that terminal node. However, this procedure might lead to overfitting the training data and poor test data prediction, so the authors suggest a tree pruning method where a very large tree is grown to be pruned back into a subtree. The objective is to select the subtree with lowest test rate which can be estimated by Cross-validation.

According to Cutler, Cutler and Stevens (2008) the largest drawback of regression trees is prediction accuracy. Therefore, tree-based ensemble methods, as Bagging, Random Forest and Boosting, can be used to increase efficiency. The authors further explain that in these methods multiple trees are combined to produce an aggregated prediction, however, the methods differ in how the predictions are aggregated.

2.2.2 Bagging

Bagging, introduced by Breiman (1996), stands for "bootstrap aggregating". Breiman (1996) explains that the method consists in using multiple version of a predictor in order to get an aggregated predictor, which averages the values obtained in each version. He further remarks that the multiple versions are generated by bootstrap replicates on the training set, meaning that datasets are randomly generated with replacement from a single training set containing the same number of observation as training set.

The different regression trees constructed using bootstrapped training sets should not be pruned since each individual tree has high variance but low bias and the variance is naturally reduced by averaging the trees (Hastie et al. 2013). Determining the number of regression trees is not a critical part of the process as using a many trees does not generate overfitting and typically a large number of trees, for instance 100, is sufficient (Hastie et al. 2013).

Breiman (1996) emphasizes that Baggging increases prediction accuracy by lossing interpretability. Cutler, Cutler and Stevens (2008) indicate that Bagging seldom have a worse perfomance than indivudual trees. In this regard, Breiman (1996) states that the key factor is the stability of the prediction model. If small changes in training set can generate large changes in predictor, bagging will improve the model. He additionally comments that Bagging can help a good but unestable model to become more optimal, however, it can degrade a stable model.

In this context, the test error can be estimaded without using Cross-validation. Hastie et al. (2013) suggest using the out-of-bag (OOB) obervations in each bootstrapped training set to predict OOB responses. This implies that the predictions are generated from the trees where the observations were OOB and averaged to a single prediction. Thus, a OOB prediction can be generated for all the obervations, leading to an OOB Mean Suared Error (MSE) that is considered a valid test error for bagged models.

2.2.3 Random Forest

Furthermore, Breiman (2001) also introduced Random Forest, a method that includes randomness by using bootstrapped predictor samples in the tree building process. In that sense, he explains that Random Forest is an effective tool in prediction, which incorporates random predictor selection to bagging.

In Random Forest, trees are fitted using a bootstrapped subset m of the total amount of predictors p; thus, these m predictors are randomly and independently chosen at each node, where the best split is determined following the single tree approach (Breiman , 2001). Trees are grown, and not pruned, until terminal nodes consisting only of a small number of observations are reached (Cutler, Cutler & Stevens, 2008).

Accordingly, m, the bootstrapped subset of predictors and the number of trees are tuning parameters in this method. Hastie et al. (2013) indicate that a commonly use value of m is $\sqrt{p}$, where p is total amouth of predictors. It is suggested that a small value of m should be considered if the predictors are correlated. Although, Breiman (2001) states that Random Forest's results are insensitive to m and the number of trees, and that adding more trees does not generate overfitting. Furthermore, Cutler, Cutler and Stevens (2008) point out that the number of predictors m can be determined by the OOB error rate, implying that m can be identified by applying Random Forest to few trees to then choose m from the OOB data. They also note that the depth of tree, that is the number of observations in the terminal nodes, can also be selected using OOB data.

Random Forest can be seen as a Bayesian procedure where accuracy comes from low bias and low correlation (Breiman , 2001). Low bias results from growing large trees and low correlation from not generating similar trees while maintaining the low bias (Cutler, Cutler & Stevens, 2008). In comparison with Bagging, it is said that in Bagging if the model contains one very strong predictor, that predictor will be at the top of the split producing similar bagged trees each time, implying that the predictions will be highly correlated and averaging them will not generate a large reduction in variance as if they were uncorrelated like in Random Forest (Hastie et al. 2013). In Bagging trees differ within each other because they come from different bootstrapped training sets, in Random Forest trees further differ because they are fitted in bootstrap samples of predictor sets at each node reducing correlation and increasing accuracy (Cutler, Cutler & Stevens, 2008).

Likewise Bagging, the estimated OOB error rate is a valid test rate and a test set of observations is not needed.

2.2.4 Boosting

Boosting is another tree-ensemble method, as Bagging and Random Forest, that focuses on improving prediction performance of an individual tree by fitting and combining multiple trees for prediction, but unlike these, Boosting does not involve randomness by using bootstrap samples.

Elith, Leathwick and Hastie (2008) explain that Boosting is a sequential, forward and stepwise procedure which fits trees to the training set repeatedly, setting focus on observations that are modeled poorly. There are many Boosting algorithms which vary in the way lack-of-fit is measured and the settings selected for each repetition. The authors note that Boosting Trees try to minimize the RSS by adding trees that best minimizes it at each step. Initially, the first regression tree minimizes the RSS, then, the next tree focuses on the residuals, that is the variation that was not explained by the model. This implies that the second tree is fitted to the residuals of the first tree, and then these two trees are combined to calculate the residuals that will be used for the third tree and so on.

As trees grow sequentially each tree can be small but have different variables and splits. Hastie et al. (2013) describe the tuning parameters of the model. These involve, as usual, the number of trees and two new parameter, the shrinkage parameter λ, which make the process even slower in order to let trees with different shapes to influence the residuals, and the parameter *d,* that determine the splits in each tree and thus control the complexity of the trees. According to Elith, Leathwick and Hastie (2008), an optimal level for the tuning parameters can be estimated using Cross-validation. Furthermore, the test error can also be computed using Cross-validation.

Hastie et al. (2013) further note that unlike bagging and random forest, Boosting depends in trees that have already been grown and in this method smaller trees can be enough. Boosting is characterized by learning slowly and that kind of approaches usually involve better performance.

2.3 Previous cryptocurrency studies

Cryptocurrencies have been attracting many researchers in the last years. Although most studies have focused on price prediction of Bitcoin, new literature about other cryptocurrencies can be found. In this section, previous studies about cryptocurrency price and returns prediction are presented according to linear and non-linear models. These studies have different purposes and use different performance metrics making it difficult to compare efficiency within each other. Nevertheless, a summary table showing the key elements of the cryptocurrency literature review is presented at the end of this section.

2.3.1 Linear models

Arora, Bhatia and Mittal (2018) used a Multivariate linear regression to predict the highest price for ten cryptocurrencies. The independent variables analyzed for each cryptocurrency model were open price, low price, close price, volume and market capitalization, however, only open price, low price and close price were found significant and used for the prediction.

Other linear models used by researchers to predict prices of cryptocurrency are ARIMA models. Bezkorovainyi et al. (2019) used Binary Auto Regressive Tree (BART), ARIMA(1,0,1) and ARFIMA (1,*d*,1), to forecast the prices of Bitcoin, Ethereum and Ripple during: a stable period, a falling trend, transition dynamics (change of trend) and rising trend. The models for each cryptocurrency were compared by the Root Mean Squared Error (RMSE) and it was concluded that the BART models presented higher efficiency in all the periods and for all cryptocurrencies. Alahmari (2019) evaluated price prediction and short-term direction for Bitcoin, Ethereum and XRP applying ARIMA models. The data was divided in daily, weekly and monthly, and volume was defined as an independent variable. The results determined that the ARIMA model with daily sample outperformed other models.

2.3.2 Non-linear models

Chowdhury et al. (2020) used Gradient Boosted Trees, Neural Network, Ensemble Learning and K-Nearest Neighbor, in order to predict and forecast the close price of the Cryptocurrencies Index 30 (cci30) and nine cryptocurrencies that constitute the index. The study was divided in prediction and forecast, using Gradient Boosted Trees, Neural Network and Ensemble Learning as prediction methods, and date, open price, high price and low price as features. Forecasting was done by K-Nearest Neighbor using date and close price. Gradient Boosted Trees and the Ensemble learning method presented better performance. However, the model was more efficient, in terms of RMSE, for each cryptocurrency than for the index.

Furthermore, other studies used Twitter sentiment as a feature to analyze cryptocurrency´s prices. One of these is presented by Fong et al. (2019) which applied a Extreme Gradient Boosting Regression Tree to analyze if user sentiments can predict price movements of a cryptocurrency with small market capitalization as ZClassic. They collected tweets during 3.5 weeks and classified each tweet as positive, neutral or negative to construct an hourly based sentiment index. Twitter sentiments and trading volume were used as features in the model, concluding that this method can serve as a viable approach for predicting cryptocurrency prices. Gomez-Espinosa, Valdés-Aguirre and Valencia (2019) applied a similar approach and used Neural Networks, Support Vector Machines and Random Forest to analyze social media and market data for predicting the price movements of Bitcoin, Ethereum, Ripple and Litecoin. Three models of different inputs were considered for each currency, one that included social data, another with market data and a third that used both market and social data. Market data consisted in high price, low price, close price and volume. Social data was collected from raw tweets from Twitter where VADER (Valence Aware Dictionary and sEntiment Reasoner) was used to quatify the emotions in each tweet. The results showed that Neural Networks outperformed other models and that Twitter data by itself could not be used to predict certain cryptocurrencies. The study concluded that Twitter data can be used by itself to predict Ripple and Litecoin, but it is not superior to the utilization of exclusively market data.

Brookins, Rinaldo and Zhao (2019) and Aiello et al. (2018) focused on returns´ prediction in order to create trading strategies. Brookins, Rinaldo and Zhao (2019) used Suport Vector Machines, Random Forest and Extreme Gradient Boosting, and clasiffied the returns of Bitcoin, Ethereum and Litecoin in "up": $r_t > 0{,}5$, "down": $r_t < 0{,}5$ and "same": $r_t = 0{,}5$. Additionally, they defined a tuning parameter γ , which served as a threshold for making trading decisions and predicting class probability from the regression functions. For the three classes, a Bayes' rule established $\gamma = 1/3$ and suggested "buy" whenever the predicted probability weight on "up" exceeded 1/3, likewise, "sell" if "down" exceeded 1/3. The study concluded that Support Vector Machines outperformed other methods and that their "all-in" strategy performed well and could avoid big losses of the market.

The other approach, presented by Aiello et al. (2018), studied three models, two of them based on Gradient Boosting: one analyzing all cryptocurrencies and the other each cryptocurrency separately, and a the third model based in Neural Networks analyzing each currency separatly.

For this, cryptocurrencies with daily trading volume higher than 10^5 USD from a sample of 1681 cryptocurrencies were analyzed. The daily price was computed as the volume weighted average of all prices and was then transformed in daily returns on investment (ROI). The features for the first and the second model were mean, standard deviation, median, close price, the difference between last and first value of: price, market capitalization, market share, rank and volume. In the third model, prediction was based on previous returns. The study concluded that the first and second model, based on Gradient Boosting, worked best when predictions were based on short-term window lengths of 5-10 days, while Neural Networks performed best when predictions were based on windows of about 50 days. Among the two methods based on Boosting, the one considering a different model for each currency performed best.

2.3.3 A wider view of predictors

The previous studies used almost the same variables in their respective models. These variables include cryptocurrency market data: open, high, low, volume, market cap and historical prices, and social media data: Tweets. Since one of the objectives of this study is to determine the variables that affect the cryptocurrency market returns, other studies that analyzed more variables are further presented.

Dempere (2019) analyzed the predictive power of selected financial variables over principal cryptocurrencies: Bitcoin, Ethereum and Ripple. The independent variables included in the study were the daily Google trend values of the words “cryptocurrencies”, “Bitcoin”, “Ethereum”, and “Ripple”, the daily log returns of exchange rates of several major currencies USD, GBP, JPY, EUR, RUB, CNY per Special Drawing Rights, the daily log returns of the S&P500 index, gold and oil. The results provided evidence that the log returns of each studied cryptocurrency had significant explanatory over each other. The study also found significant results for the daily Google trend values of the search terms “Bitcoin”and “Ripple”, the log returns of the exchange rate of Chinese Yuan per SDRs and the S&P500 index. The log returns of oil prices have a significant relationship with Bitcoin and Ethereum, but not with Ripple.

Panagiotidis, Stengos and Vrabosinos (2018) studied the determinants of Bitcoin using a Lasso approach. The variables examined in the study were the oil and gold prices, the Fed Fund effective rate and ECB deposit facility rate, the exchange rates USD to EUR, GBP, CNY, JPY. The stock markets index: Dow Jones, Nasdaq, Nikkei 225, S&P 350, SSEC, the VXD volatility index, the Policy Uncertainty Index for US, Europe and China and the Google and Wikipedia trend for the term "Bitcoin". Search intensity and gold return were found as the most important variables for Bitcoin returns.

As a summary, table 1 present the important characteristics of the previous studies about cryptocurrencies.

Source	Objective	Explanatory variables	Empirical Method	Performance Metrics
Arora, Bhatia and Mittal (2018)	Predict highest price of 10 cryptocurrency (individually)	Open Low Close Volume Market Cap Type of coin Delta of currency	Linear Regression	Accuracy
Bezkorovainyi et al. (2019)	Short-term forecasting model for prices of Bitcoin, Ethereum and Ripple	Price past values	Binary Auto Regressive Tree (BART), ARIMA(1,0,1) and ARFIMA	RMSE
Alahmari (2019)	Price prediction and short-term direction for Bitcoin, Ethereum and XRP	Volume	ARIMA	Mean absolute error (MAE), MSE, RMSE
Chowdhury et al. (2020)	Predict and forecast close price of cci30 and 9 cryptocurrencies	Date Open High Low Date (forecast) Price (forecast)	Gradient Boosted Trees, Neural Networks, Ensemble Learning, K-Nearest Neighbor	RMSE, prediction trend, accuracy, absolute error, relative error, squared error
Fong et al. (2019)	Twitter sentiment as a feature to analyze cryptocurrency´s prices: Zclassic	Positive Negative Neutral Unweighted Index Weighted Index (retweets) Trading volume	Extreme Gradient Boosting Regression Tree	Pearson correlation

Gomez-Espinosa, Valdés-Aguirre and Valencia (2019)	Predict price movements of Bitcoin, Ethereum, Ripple and Litecoin	High (market data) Low (market data) Open (market data) Volume (market data) Neutral (tweets-social data) Negative (tweets-social data) Positive (tweets-social data) Polarization (tweets-social data) Norm (tweets-social data)	Neural Networks -MLP, Support Vector Machines, Random Forest	Accuracy
Brookins, Rinaldo and Zhao (2019)	Returns´ prediction to create trading strategies for Bitcoin, Ethereum and Litecoin	Open High Low Close Volume Other features based on historical prices and volumes On-chain data	Support Vector Machines, Random Forest, Extreme Gradient Boosting	Accuracy
Aiello et al. (2018)	Anticipate cryptocurrency prices. Construct investment portfolios	Price: last, mean, trend, std, median ROI: mean, trend, std, median Age: last, mean, trend, std, median Volume: last, mean, trend, std, median Rank: last, mean, trend, std, median Market share: last, mean, trend, std, median Market Cap: last, mean, trend, std, median NN: ROI past values	Gradient Boosting, Neural Networks, Baseline model: simple moving average strategy (SMA)	Profits (expressed in Bitcoin) over the entire considered period and for a large set of shorter trading periods
Dempere (2019)	Predictive power of selected financial variables over: Bitcoin, Ethereum and Ripple.	Google trends USD, GBP, JPY, EUR, RUB, CNY per SDRs Log returns of S&P500 Log returns of gold Log return of oil prices	PGARCH, ECGARCH, TGARCH and GARCH models	Statistical significance at the 0.1%, 1%, 5%, and 10% significance
Panagiotidis, Stengos and Vravosinos (2018)	Examine the significance of twenty-one potential drivers of Bitcoin returns	Gold and oil prices EFFR ECB deposit facility rate USD to EUR, GBP, CNY, JPY Dow Jones Index, Nasdaq, Nikkei 225 S&P 350, SSEC VXD EPU for US, China and Europe Google and Wikipedia trend "Bitcoin"	Lasso	Lasso coefficients and direction movements

Table 1. Summary literature review about the cryptocurrency market

3. Data

A dataset including all cryptocurrencies was extracted from coinmarketcap.com, a source frequently used in previous literature. At the time of the data collection, eleven cryptocurrencies with largest market capitalization represent the 90% of the total market capitalization (see table 2). Since these cryptocurrencies highly represent the market, they will be used as the base for this study. It is considered that other cryptocurrencies, following in market capitalization, have a smaller market share and will not make a big contribution to the main results. Following a CAPM market portfolio determination, where assets are weighted in proportion to their presence in the market, the close price of these top eleven cryptocurrencies are weighted according to their market capitalization resulting in a daily capitalization-weighted index (“cap-weighted index”).

#	Name	Symbol	Market Cap	% of Total MarketCap
1	Bitcoin	BTC	$170,651,691,691	67%
2	Ethereum	ETH	$23,261,371,032	9%
3	XRP	XRP	$9,695,669,860	4%
4	Tether	USDT	$6,431,242,163	3%
5	Bitcoin Cash	BCH	$4,613,776,379	2%
6	Bitcoin SV	BSV	$3,861,539,716	2%
7	Litecoin	LTC	$3,057,758,053	1%
8	Binance Coin	BNB	$2,651,426,622	1%
9	EOS	EOS	$2,567,439,196	1%
10	Tezos	XTZ	$1,952,676,394	1%
11	Stellar	XLM	$1,468,255,838	1%
				90%

Table 2. Top 11 cryptocurrencies - representing 90% of total cryptocurrency market (data on 6th May 2020)

Data for the cryptocurrencies is collected for the period from April 29th, 2013 to May 1st, 2020. Some cryptocurrencies were released at a later point of time from the initial date of analysis and are included in the daily cap-weighted index at their respective date of release. Figure 1 shows the evolution of the prices for each cryptocurrency. Bitcoin, which reached its highest level of $19 497 in December 2017, represents the highest prices in the market followed by Bitcoin Cash and Ethereum. Further, another period with big rises for these three cryptocurrencies, in particular Bitcoin, is between June and August 2019. Moving away from these levels are Litecoin and Monero which rose to their highest rates in November 2017 and kept levels above $400 and $250 until January 2018. Other cryptocurrencies have prices under $40 during the all the study period.

The daily cap-weighted index calculated and consisting of 2 030 observations are also shown in figure 1, these will be transformed to log returns to be used as the response variable in this study. Figure 2 shows the cap-weighted index returns, hereinafter referred to as returns, and the summary statistics are presented in Table 2.

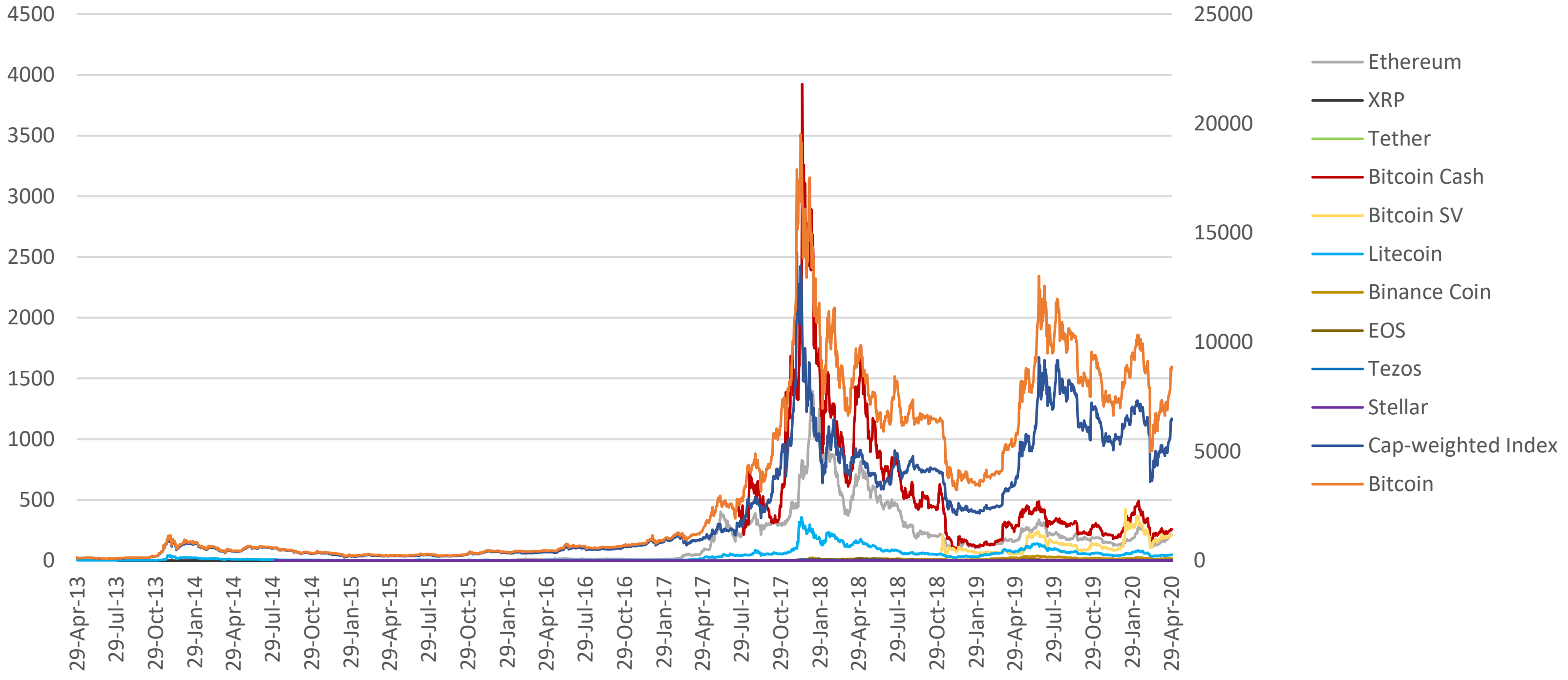


Figure 1. Prices of the top 11 cryptocurrencies and the WMAP. The prices of Bitcoin and the WMAP are presented in the right axis, the rest in the left axis

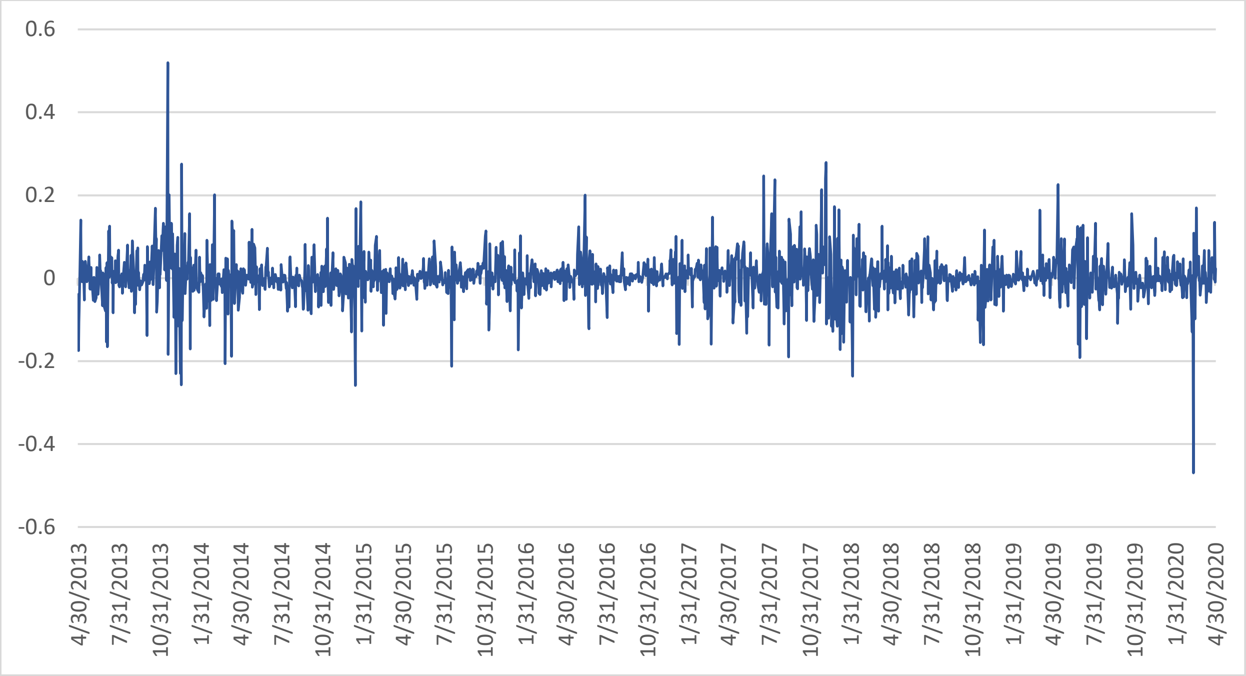


Figure 2. Cap-weighted log returns from April 30th, 2013 to April 30th, 2020

Mean	0.0021
Standard Error	0.0012
Median	0.0011
Standard Deviation	0.0534
Sample Variance	0.0029
Kurtosis	11.8832
Skewness	0.1228
Range	0.9893
Minimum	-0.4696
Maximum	0.5197
Sum	3.8501
Count	1829

Table 3: Descriptive statistics of Cap-weighted log returns

Regarding the predictors, almost all variables that have been suggested in previous literature and other variables of interest are considered, representing cryptocurrency market data, information demand, commodities, financial markets and macroeconomics. The cryptocurrency market data include the market cap weighted values of: open, high, low and volume. A market capitalization one-day-before value is incorporated to evaluate the effects of cryptocurrency supply. Information demand in cryptocurrencies is represented by the google search trends of the term “cryptocurrency”. The commodities included in the study are: gold spot price, gold futures, oil spot price, oil futures, silver spot price, silver futures, gas spot price and gas futures. Regarding financial markets the following indexes are incorporated: Russell 2000, IMOEX, SICOM, FTSE China A50, SHCOMP, HSI, BSE Sensex, NIKKEI, NASDAQ, S&P500, DJI, IBEX 35, FTSE 100, DAX and CAC 40. Additionally, the Trade Weighted U.S. Dollar Index (DWTEXBGS), the exchange rates USD to: GBP, EUR, JPY, CNY, RUB and the VIX, VXD and FFER are examined. Moreover, macroeconomics variables as the Economic Policy Uncertainty Daily Index for US, China and UK are analyzed. Other macroeconomic variables are not included as they are computed in monthly basis. The logarithmic values of all the predictors, except for volume, are used to make them comparable with the response variable. A summary of the 41 predictors, their groups, transformations and sources, is presented in Appendix 1.

Cryptocurrency data can be obtained for the weekends, but other variables do not follow the same frequency, thus weekend cryptocurrency data is removed and when missing data is found the last value is repeated.

4. Methodology

Firstly, stationarity is tested in the returns and model identification performed to evaluate if an ARMA model applies to the returns time series.

Stationarity is tested using an ADF test. In this case, the ADF test specification does not include a time trend and a drift, as the p-values in the lags that minimizes the information criteria indicate that they are not significant in the test equation. The results from the ADF test show that the null hypothesis of the time series containing a unit root is rejected, thus the returns can be considered stationary.

Figure 3 shows the Sample Autocorrelation Function (SACF) and Sample Partial Autocorrelation Function (SPACF). In a non-stationary process the ACF will not convert to zero as the lag length increases presenting shocks that never die away and persist indefinitely in the system (Brooks, 2014). The SACF plot shows that the ACF is under or at the level of the significance, in almost all the lags, with values that approximate zero. This confirms the ADF test outcome that a unit root or non-stationarity is rejected for the returns. Regarding model identification, it cannot be concluded from the ACF and PACF plots since they do not exhibit a simple pattern. Thus, the order of the model is identified applying information criteria and a Box-Jenkins approach for ARMA modeling is used.

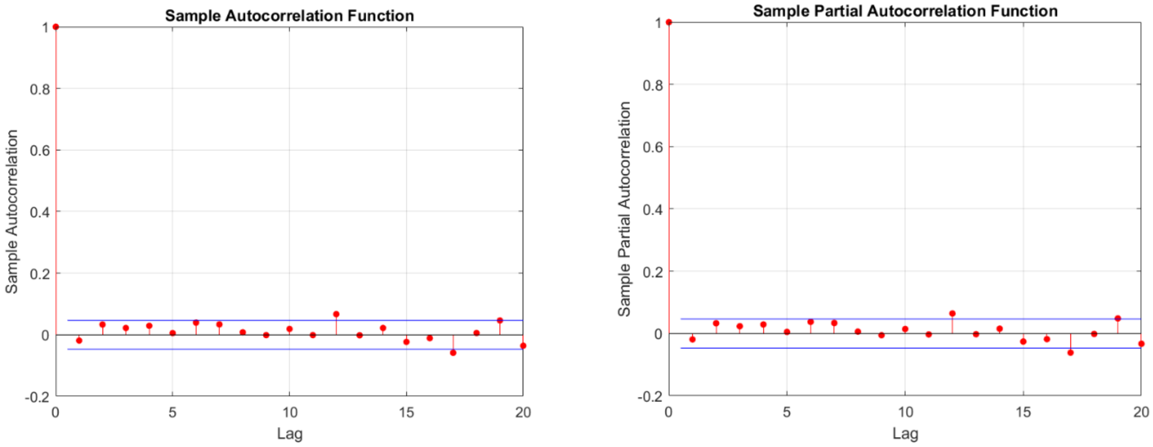


Figure 3: Plot of Sample Autocorrelation Function and Sample Partial Autocorrelation Function

Brooks (2014) describes the Box-Jenkins approach as a practical and pragmatical method which contains three steps: identification, estimation and diagnostic checking.

For the first step, as mentioned, graphical procedures are not appropriate so information criteria is used to identify the model specification. Brooks (2014) explains that the number of parameters chosen should minimize the value of the information criteria, which involves two factors: one that is a function of the Residual Sum of Squares (RSS) and another that is considered a penalty for the loss of degrees of freedom caused by adding extra parameters. The author further emphasizes that adding an extra term will reduce the information criteria only if the decrease in the RSS is larger than the penalty term involved.

Once the ARMA model is specified, the predictors are included to the model. Since Machine Learning methods can handle a large number of inputs, many variables used in previous studies are collected and used as predictors. Stationarity is also tested in all the predictors using ADF test. The first difference is calculated for non-stationary series (Appendix 1) and stationarity is tested again to make sure that all predictors used in this study are stationary time series. The predictors are included in the model to perform the objectives of this study: inference and prediction of the cryptocurrency market returns.

Considering the trade-off between interpretability and prediction accuracy, the process is divided in two stages where the first consists in studying the explanatory power of the predictors (inference) and the second focuses on the prediction power (prediction). Hastie et al. (2013) mention that more restrictive methods, that produce a small range of shapes to estimate the function $\hat{f}$ i.e. linear methods, should be used if inference is the objective. However, more flexible methods, that generate a much wider range of shapes to estimate $\hat{f}$ as non-linear methods, should be used if the purpose is prediction accuracy.

Lasso, using a fitting procedure to estimate the coefficients setting some of them to zero, is more restrictive than a linear regression but also more interpretable as the response is related to a small subset of relevant predictors (Hastie et al. 2013). Thus, for inference a Lasso regression is performed in order to define the relationship between the predictors and the returns and gain insights about the optimal features surrounding the cryptocurrency market returns. The Lasso regression, as shown in equation 2, examine the returns and the explanatory variables on the same day to determine the immediate relationship for the inference purpose.

$$R_t = \alpha + \beta X_t + \varepsilon_t \qquad (2)$$

For prediction, one-day-before data for all the predictors is used, i.e. yesterday's data is examined to predict the daily returns. In this sense, the one-day-before return is included as predictor like shown in equation 3.

$$R_t = \alpha + \delta R_{t-1} + \beta X_{t-1} + \varepsilon_t \qquad (3)$$

Non-linear methods as Bagging, Boosting, Random Forest and Support Vector Machines are hard to interpret but yield more accurate predictions (Hastie et al. 2013). Thus, for prediction all the predictors and the subset of relevant predictors selected by Lasso for the next-day-returns are used in non-linear methods: Regression Trees, Random Forest and Boosting to determine the prediction model that can produce better results for the cryptocurrency market returns i.e. the model with the lowest test Mean Square Error (MSE).

The test MSE is a measure that quantifies the approximation of a predicted value from the estimated function $\hat{f}$, to the true response value, i.e. the true value of the cryptocurrency market return R_t (Hastie et al. 2013), as shown in equation 4.

$$MSE = \frac{1}{n}\sum_{t=1}^{n}\left(R_t - \hat{f}(X_t)\right)^2 \tag{4}$$

The MSE will be small if the predicted value is close to the true response. In that sense, the most accurate method will be the one with lowest test MSE rate. This value will be determined for each model in the prediction section.

5. Results

Following the Box-Jenkins approach an ARMA(0,0) is suggested. This means that the model does not include a moving average or autoregressive lag. Thus, only the predictors are used as explanatory variables in the model. The results are presented regarding inference and prediction.

5.1 Inference

As mentioned, the predictors (except the market cap lag) and the response were analyzed in the same day for the inference purpose. As stated in the previous literature section, according to Hastie et al. (2013), inference refers to estimating a function in order to explain how changes in the input variables affect the response variable, that is, which variables are associated with the response and what is the relationship between the response and the input variables.

The results reveal that Lasso removed 27 predictors, that is, from the 41 predictors only 14 were related to the response. This result was obtained using the value of the tuning parameter λ that minimized the Cross-validation error (see Appendix 2). Table 4 shows the variables selected by Lasso ranked according to the absolute value of the coefficients. The Lasso regression was run several times where some changes in the coefficients were observed, but the ranking and the sign of the coefficients remained the same.

Variables	Coefficients
High	0.6642
Low	0.6640
Open	-0.4696
Market cap lag	-0.1431
DJI	0.1041
Gold spot	0.0474
USDRUB	-0.0263
NASDAQ	0.0242
IMOEX	0.0131
VIX	-0.0017
Gas futures*	-8.30E-04
Gas spot*	-2.94E-04
Oil futures*	1.38E-04
Volume	1.06E-12

Table 4. Variable selection by Lasso
*Excluded in one model in the Lasso regression

Thus, after removing irrelevant variables and unnecessary complexity, a model that is more easily interpreted containing 14 variables is obtained. The variables associated with the response include market data, commodities, stock market indexes, the VIX and the USD-RUB exchange rate.

The largest positive coefficients correspond to the high and low values of the cryptocurrency market, followed by the open and market cap lag with negative coefficients. The stock market indexes Dow Jones, NASDAQ and IMOEX have a positive relationship with the returns, being the most important the Dow Jones Index. The USD-RUB exchange rate and VIX affect the cryptocurrency returns negatively. Regarding commodities, the effect of gold is positive. Other commodities with lower coefficients, implying lower influence in the cryptocurrency returns, are gas spot and futures with a negative effect, and oil futures whit positive effects in the returns. Finally, volume with the lowest coefficient affects positively the cryptocurrency market returns.

5.2 Prediction

As mentioned in the previous sections, prediction uses available inputs X to produce a response that is not available yet so the response variable is the next-day-returns, and in contrast to inference, the exact form of the estimated function is not important as the predictions obtained in each model. Following, the results are presented according to each method after a very short reminder from the literature review.

5.2.1 Lasso

A less flexible method as Lasso does not usually provide accurate predictions. However, removing irrelevant predictors can imply prediction power if the true relationship between the predictors and response is approximately linear (Hastie et al. 2013).

Likewise inference, the model selected by Lasso for prediction implies the value of the tuning parameter λ that minimizes the Cross-validation error (see Appendix 3). The results indicate that five variables are related with prediction of the next-day-return, shown in table 5. In this case, the exchange rate USD to CNY have a very high positive coefficient implying a strong relationship with the next-day-returns. The effect of the stock market index NIKKEI 225 is positive while the effect of the BSE Sensex is negative. Moreover, market cap and gas values affect the next-day cryptocurrency market returns negatively.

Variables	Coefficients
USDCNY	0.726
NIKKEI 225	0.035
BSE	-0.023
Market cap	-2.86E-04
Gas spot	-4.82E-04

Table 5. Variable selection by Lasso

When running the Lasso regression several times, another two models where observed. One model just included the USD-CNY exchange rate, and the second included none of the predictors. Nevertheless, in relation to prediction accuracy, the model with lowest test MSE is the one that included the five variables presented in table 5 with a test MSE of 0.00274.

As a result from variable selection for prediction, a new dataset of predictors is defined as stated in Methodology. Thus, there are two datasets that will be inputs for the following methods, one considering all the predictors, as shown previously in equation 4, and other following the same equation but focused only in the predictors that Lasso found relevant in its prediction model with lowest test MSE: USDCNY, NIKKEI 225, BSE Sensex, market cap and gas spot.

5.2.2. Regression Trees

When the predictors and response variable have a complex and non-linear relationship, Regression trees are more efficient than linear regressions (Hastie et al. 2013).

The results of regression trees, considering the all-predictors dataset as input, show a model with just a root node where the returns lag value of 0.00217 is the splitting point. The observations with a value lower than 0.00217 go to the left with a predicted response of -0.0012, that is the mean value of the observations at that terminal node. On the contrary, the returns with a value higher than 0.00217 go to the right where a prediction of 0.0057 is made. Thus, the shape of the tree is very simple (see figure 4) and no other predictor is considered. The test MSE for this model is 0.00283.

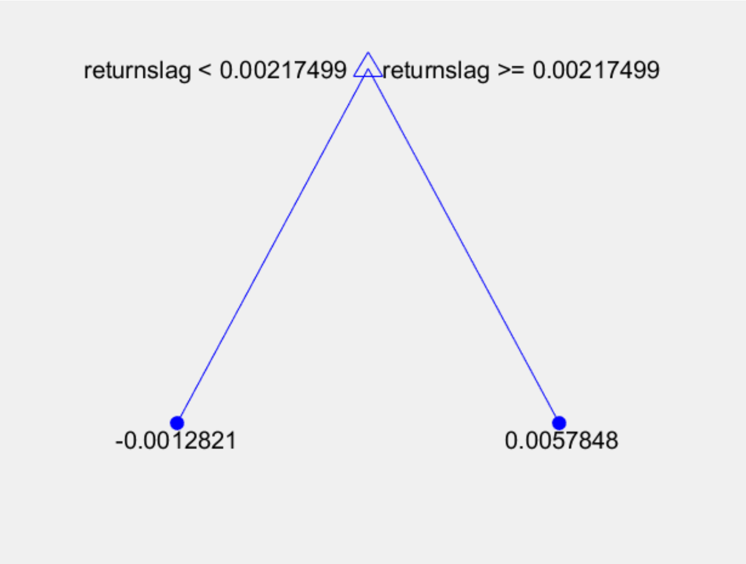


Figure 4. Regression Tree with all-predictors dataset

Similar is the result for the Lasso-predictors dataset (figure 5). A very simple tree with a root node that correspond to the market cap value of -0.00075. The observations with lower values go the left with a predicted response of -0.0010, otherwise to the right where the terminal node predicted value is 0.0048. No other predictor is included in the tree and a test MSE of 0.00284 is obtained.

In both regression tress, the test MSE corresponds to an automated Cross-validation optimization of hyperparameters, i.e. the number of observations in the terminal nodes, that uses Bayesian optimization to minimize the Cross-validation error (see Appendix 4-5).

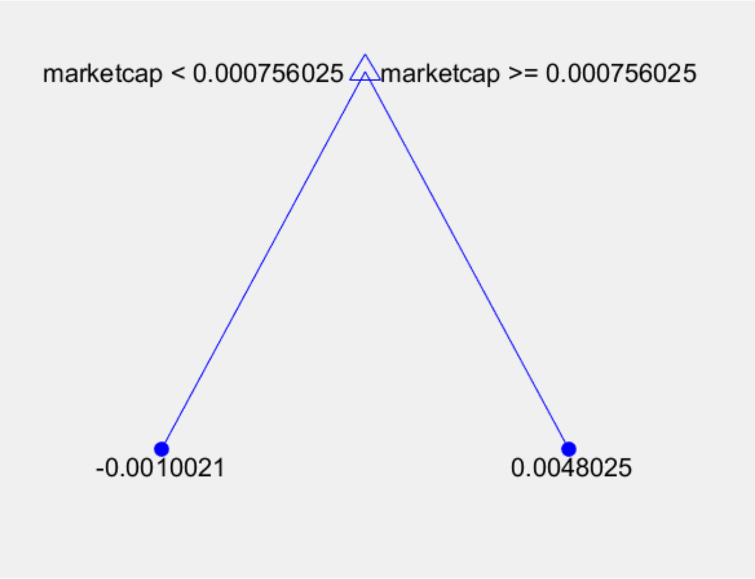


Figure 5. Regression Tree with Lasso-predictors dataset

5.2.3 Random Forest

A more flexible method as Random Forest can increase prediction accuracy since randomness is entered by using bootstrapped predictor samples in the tree building process (Breiman, 2001).

To estimate the results for Random Forest with all-predictors dataset a bootstrapped subset of 14 predictors, randomly and independently chosen, were analyzed to find the best split at each node. Moreover, a minimum value of 5 observations in each leave was considered. The number of trees examined is 100, although, the MSE maintains the same level after 90 trees (Appendix 6). As mentioned in the literature review, the results are insensitive to the amount of predictors in the bootstrapped sample and the number of trees, so including more trees does not generate overfitting (Breiman, 2001). Consequestly, the above mentioned values should not affect the MSE.

Regarding Random Forests' predictors importance, the highest values correspond to market cap, returns lag, USD-CNY, gas futures, DAX and VXD (figure 6). It is remarkable that the USD-CNY exchange rate also appears as an important predictor for the next-day-returns when Random Forest examines all predictors. The test MSE for this model is 0.00103.

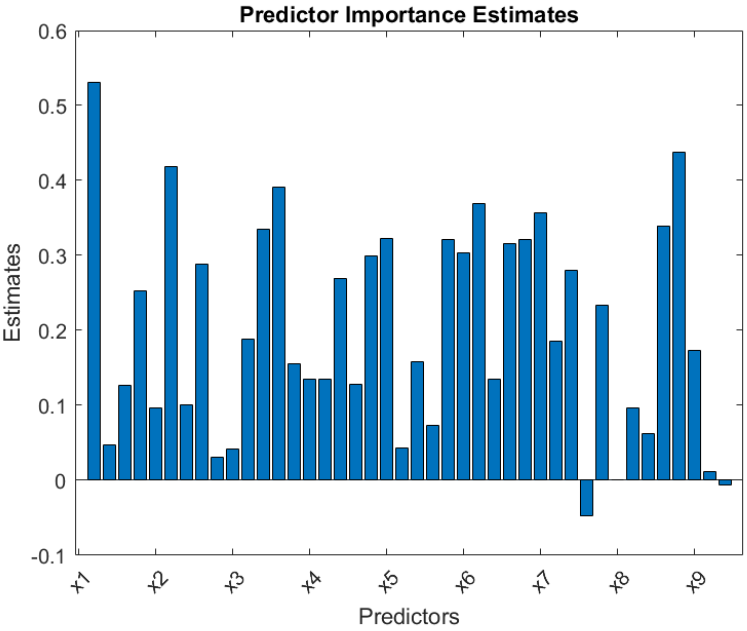


Figure 6. Predictor importance from RF all-predictors dataset
Largest estimates for: Return lag, USDCNY, Market cap, Gas Future, DJI, CAC 40.
From left to right, the first bar is Return lag, then predictors are shown in the same order as in Appendix 1.

Random forest, performed with the predictors selected by Lasso, USD-CNY, NIKKEI, BSE, Market cap and gas, uses a bootstrapped sample of 2 predictors in each node and minimum leaf size of 5 observation. Likewise, 100 trees are evaluated but the MSE remains at the same level after 90 trees (Appendix 7). In this case, the variables have the following order of importance USD-CNY, BSE, Market cap, NIKKEI and gas, and only the stock index NIKKEI has a negative relationship with the returns (Figure 7). For this model, the test MSE is 0.00136.

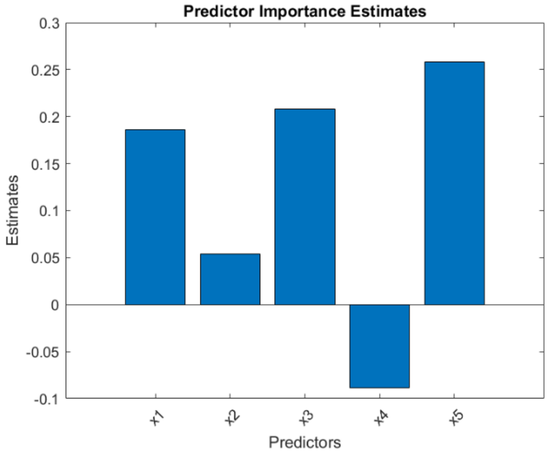


Figure 7. Predictor importance from RF Lasso-predictors dataset.
From left to right: Market cap, gas spot, BSE, NIKKEI, USDCNY

5.2.4 Boosting

Considering Boosting a sequential, forward and stepwise method the focus is on the observations that are poorly modeled (Elith, Leathwick and Hastie, 2008).

The tuning parameters for this method as the learning rate, the number of splits and trees, are also obtained using the automated Cross-validation optimization. As there are many Boosting algorithms, this Bayesian optimization selects the method between Bag and LSBoost, choosing LSBoost for both models.

For the model with all the predictors, the results show that the optimal number of trees is one and the maximum number of splits, the parameter that control the complexity of the trees, is also one. Therefore, it is obtained a tree like the one in regression trees with just one node, the root node (Appendix 7). Moreover, the learning rate which make the process even slower in order to let trees with different shapes to influence the residuals, also corresponds to an optimal value of one. The model obtained with these tuning parameters has a test MSE of 0.00279.

Boosting for the Lasso-predictors dataset gives a very similar outcome, i.e. a tree with just one split (Appendix 8), but in this case 24 trees are fitted and a learning rate of 0.25 computed. As mentioned in the literature review, this implies that the first regression tree minimizes the RSS, then the second tree is fit to the residuals of the first tree, and then these two trees are combined to calculate the residuals that will be used for the third tree and so on for the 24 trees. The test MSE for this model is 0.00272.

5.2.5 Performance comparison

Comparing the test MSE of the different methods and models (table 6), the lowest MSE rate correspond to Random Forest all-predictors model, followed by the same method but having only the Lasso-predictors as inputs. Thus, Random Forest out-of-bag estimates produces a model that can be used to predict cryptocurrency market returns. The next model in prediction accuracy is Boosting with Lasso-predictors, followed by Lasso and Boosting all-predictors model. Regression Trees, both with all-predictor and Lasso-predictors, are not very efficient predicting the cryptocurrency market returns. This result was expected as usually Random Forest and Boosting outperform Regression Trees as indicated in the literature review. However, Lasso outperforming Regression Trees suggests that removing irrelevant variables produced better predictions than the flexibility provided by Regression Trees.

Method	Predictors dataset	MSE
Lasso	Lasso	0.002741
Regression Trees	All	0.002839
Regression Trees	Lasso	0.002843
Random Forest	All	0.001034
Random Forest	Lasso	0.001361
Boosting	All	0.002789
Boosting	Lasso	0.002723

Table 6. Model performance

6. Conclusions

This study analyzed the cryptocurrency market through the cap-weighted index returns of the top eleven cryptocurrencies which represent the 90% of the total market. To study the features surrounding the cryptocurrency market, 41 predictors where selected including variables used in previous studies and other variables of interest, representing cryptocurrency market data, information demand, financial markets, commodities, exchange rates and macroeconomics.

The explanatory power of the predictors was studied using a Lasso regression. It is concluded that the cryptocurrency market variables are the most important determinants of the cryptocurrency market returns. Stock market indexes Dow Jones, NASDAQ, IMOEX have a positive effect in the returns, being the most relevant the Dow Jones Index, and implying that the hedging popularity of the cryptocurrency market has to be carefully considered. Moreover, gold affects the returns positively and the USD-RUB exchange rate negatively. Thus, the cryptocurrency market depends on financial markets, gold and USD-RUB rates but is detached from the Economic Policy Uncertain indexes, Central Bank rates and Google trend for "cryptocurrency". Gas and oil futures seem to have a low effect in the market.

When predicting the next day returns, variable selection by Lasso determined as relevant predictors the USD-CNY rate, NIKKEI, BSE, Market cap and Gas, in particular, the USD-CNY emerged as the most important determinant. Consequently, it seems that Asian markets can provide insights about the next day returns of the cryptocurrency market. Furthermore, the USD-CNY rate was one of the most important predictors in both Random Forest models, confirming the importance of this variable.

Regarding the model with the best prediction accuracy, Random Forest with all predictors presents the lowest test MSE 0.0010 confirming that a more flexible model provides better predictions (as the theoretical perspective indicates) and suggesting a non-linear relationship between the predictors and the cryptocurrency market returns.

Boosting, being a method that learns slowly usually has good performance, however, in this case did not outperform Random Forest. This could be because the tuning parameters for Boosting where chosen using Cross-validation in contrast to Random Forest where the method is insensitive to the tuning parameters. In this sense, using Cross-validation for time series might produce inefficient results. Cross-validation for time series is a current research area. Thus, a future study can apply an alternative to Cross-validation and use other machine learning models as Support Vector Machines and Neural Networks to further analyze the cryptocurrency market.

References

Alahmari, S. A., 2019. Using Machine Learning ARIMA to Predict the Price of Cryptocurrencies. *ISeCure,* vol. 11, no.3, pp. 139-144, Available at: http://www.isecure-journal.com/article_90865.html [Accessed 19 04 2020].

Aiello, L. M., Alessandretti, L., Baronchelli, A. & ElBahrawy, A., 2018. Anticipating Cryptocurrency Prices Using Machine Learning. *Complexity,* vol. 2018, pp. 1-16, Available at: https://www.hindawi.com/journals/complexity/2018/8983590/ [Accessed 19 04 2020].

Arora, S., Bhatia, M., & Mittal, R., 2018. Automated Cryptocurrencies Prices Prediciton Using Machine Learning. *ICTACT Journal on Soft Computing,* vol. 8, no. 4, p. 1758-1761, Available at: http://ictactjournals.in/paper/IJSC_Vol_8_Iss_4_Paper_8_1758_1761.pdf [Accessed 19 04 2020].

Bezkorovainyi, V., Datsenko, N., Derbentsev, V. & Stepanenko, O., 2019. Forecasting Cryptocurrency Prices Time Series Using Machine Learning Approach. s.l., *SHS Web of Conferences*; 2019, vol. 65, p1-7, Available at: http://ceur-ws.org/Vol-2422/paper26.pdf [Accessed 19 04 2020].

Breiman , L., 2001. *Random Forest,* Berkeley: Statistics Department, University of California, Available at: https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf [Accessed 19 04 2020].

Breiman, L., 1996. *Bagging Predictor,* Berkeley: Statistics Department, University of California, Available at: https://www.stat.berkeley.edu/~breiman/bagging.pdf [Accessed 19 04 2020].

Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J., 1984. *Classification and Regression Tress.* s.l.:CRC.

Brookins, C., Rinaldo, A. & Zhao, D., 2019. Cryptocurrency Price Prediction and Trading Strategies Using Support Vector Machines, Available at: https://arxiv.org/abs/1911.11819 [Accessed 19 04 2020].

Brooks, C., 2014. *Introductory Econometrics for Finance.* 3rd ed. New York: Cambridge University Press.

Cheng, E., 2020. Mc Donald's is reportedly part of Chinas's digital currency trial, CNBC, 24 April, Available at: https://www.cnbc.com/2020/04/24/china-digital-currency-mcdonalds-starbucks-part-of-pilot-program.html [Accessed 16 05 2020].

Chowdhury, R., Mahdy, M., Rahman, M. A. & Rahman, M. S., 2020. Predicting and Forecasting the Price of Constituents and Index of Cryptocurrency Using Machine Learning. *Physica A: Statistical Mechanics and its Applications,* DOI: 10.1016/j.physa.2020.124569 [Accessed 19 04 2020].

Cutler, A., Cutler, D. R. & Stevens, J. R., 2008. *Tree-based Methods,* s.l.: Department of Mathematics and Statistics, Utah State University, Available at: https://link.springer.com/chapter/10.1007/978-0-387-69765-9_5 [Accessed 19 04 2020].

Dalalyan, A., Hebiri, M., & Lederer, J., 2017. On the prediction performance of the Lasso. *Bernoulli,* vol.23, no.1, pp. 552-581, Available at: https://arxiv.org/abs/1402.1700 [Accessed 15 05 2020].

Dempere, J.M., 2019. Factors Affecting the Return and Volatility of Major Cryptocurrencies, Sixth HCT Information Technology Trends (ITT), IEEE, pp. 104-109, Available at: https://ieeexplore.ieee.org/document/9075117 [Accessed 06 05 2020].

Elith, J., Hastie, T. & Leathwick, J. R., 2008. A working guide to boosted regression tress. *Journal of Animal Ecology,* vol.7, no.4, pp. 802-813, Available at: https://besjournals.onlinelibrary.wiley.com/doi/full/10.1111/j.1365-2656.2008.01390.x [Accessed 19 04 2020].

Fong, X. R., Fu, F., Li, T. R. & Rizik, Nicholas R., 2019. Sentiment-based Prediction of Alternative Cryptocurrency Price Fluctuations Using Gradient Boosting Tree Model. *Frontiers in Physics,* vol. 7, Available at: https://arxiv.org/abs/1805.00558 [Accessed 19 04 2020].

Gomez-Espinosa, A., Valdés-Aguirra, B. & Valencia, F., 2019. Price Movement Prediction of Cryptocurrencies Using Sentiment Analysis and Machine Learning. *Entropy,* vol. 21, no. 6, p. 589, Available at: https://www.mdpi.com/1099-4300/21/6/589/pdf [Accessed 19 04 2020].

Hastie, T., James, G., Tibshirani, R. & Witten, D., 2013. *An Introduction to Statictical Learning.* New York: Springer.

Panagiotidis, T., Stengos, T., and Vrabosinos, O., 2018. On the determinants of bitcoin returns: A LASSO approach. *Finance Research Letters*, vol. 27, pp. 235-240, Available at: https://www.sciencedirect.com/science/article/abs/pii/S1544612318300023 [Accessed 07 05 2020].

Parashar, N., & Rasiwala, F., 2019. Bitcoin - Asset or Currency? User's Perspective About Cryptocurrencie. *IUP Journal of Management Research,* vol. 18, no. 1, pp. 102–122, Available at: https://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=134816497&site=eds-live&scope=site [Accessed 04 05 2020].

Sun, X., Liu, M. & Sima, Z., 2020. A novel cryptocurrency price trend forecasting model based on LightGBM. *Finance Research Letters*, vol. 32, Available at: https://www.sciencedirect.com/science/article/pii/S1544612318307918 [Accessed 05 05 2020].

Vejačka, M., 2014. Basic Aspects of Cryptocurrencies. *Journal of Economy, Business and Financing*, vol 2, no.2, p. 75 – 83, Available at: https://www.researchgate.net/publication/292586903_Basic_Aspects_of_Cryptocurrencies [Accessed 19 04 2020].

Appendices

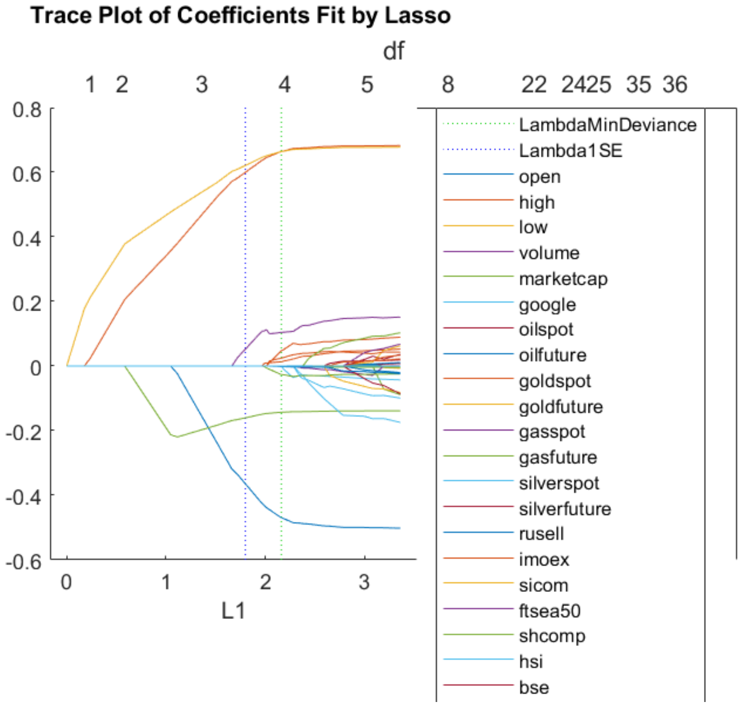
Appendix 1
Predictors: Group, Transformation and Source

Group	Predictor	Transformation	Source
Cryptocurrency market data	Open	Cap-weighted, Log , Diff	CoinMarketCap
	High	Cap-weighted, Log , Diff	CoinMarketCap
	Low	Cap-weighted, Log , Diff	CoinMarketCap
	Volume	Cap-weighted, Diff	CoinMarketCap
	Market Cap	Log, Diff	CoinMarketCap
Information	Google Search	Log	Google Trends
Commodities	Oil spot - DCOILWTICO	Log , Diff	Federal Reserve Bank of St. Louis
	Oil futures - CL1 COMB	Log , Diff	Bloomberg
	Gold spot - XAUUSD CUR	Log , Diff	Bloomberg
	Gold futures - GC1 COMB	Log	Bloomberg
	Gas spot - NGUSHHUB Index	Log	Bloomberg
	Gas futures - NG1 COMB	Log	Bloomberg
	Silver spot - XAG CUR	Log	Bloomberg
	Silver futures - SI1 COMB	Log	Bloomberg
Financial Markets	RUSELL 2000 Index	Log , Diff	Bloomberg
	IMOEX Russia Index	Log , Diff	Bloomberg
	Shenzhen Component Index - SICOM	Log , Diff	Bloomberg
	FSTE China A50 Index	Log , Diff	Bloomberg
	Shanghai Composite Index - SHCOMP	Log , Diff	Bloomberg
	Hang Seng Index - HSI	Log , Diff	Bloomberg
	S&P BSE SENSEX Index	Log , Diff	Bloomberg
	NIKKEI 225	Log , Diff	Bloomberg
	NASDAQ	Log , Diff	Bloomberg
	S&P 500 Index	Log , Diff	Bloomberg
	Dow Jones Industrial Average - DJI	Log , Diff	Bloomberg
	IBEX 35 Index	Log , Diff	Bloomberg
	FTSE 100 Index	Log , Diff	Bloomberg
	Deutsche Boerse AG German - DAX	Log , Diff	Bloomberg
	CAC 40 Index	Log , Diff	Bloomberg
	CBOE Volatility Index - VIX	Log	Bloomberg
	CBOE DJIA Volatility Index - VXD	Log	Bloomberg
	Federal Funds Effective Rate - FFER	Log , Diff	Bloomberg
Exchange rates	Trade Weighted U.S. Dollar - DWTEXBGS	Log , Diff	Federal Reserve Bank of St. Louis
	USDGBP	Log , Diff	Bloomberg
	USDJPY	Log , Diff	Bloomberg
	USDEUR	Log , Diff	Bloomberg
	USDRUB	Log , Diff	Bloomberg
	USDCNY	Log , Diff	Bloomberg
Macroeconomics	Economic Policy Uncertainty Index U.S.	Log	Federal Reserve Bank of St. Louis
	Economic Policy Uncertainty Index China	Log	economicpolicyuncertaintyinchina.weebly.com
	Economic Policy Uncertainty Index UK	Log	www.policyuncertainty.com

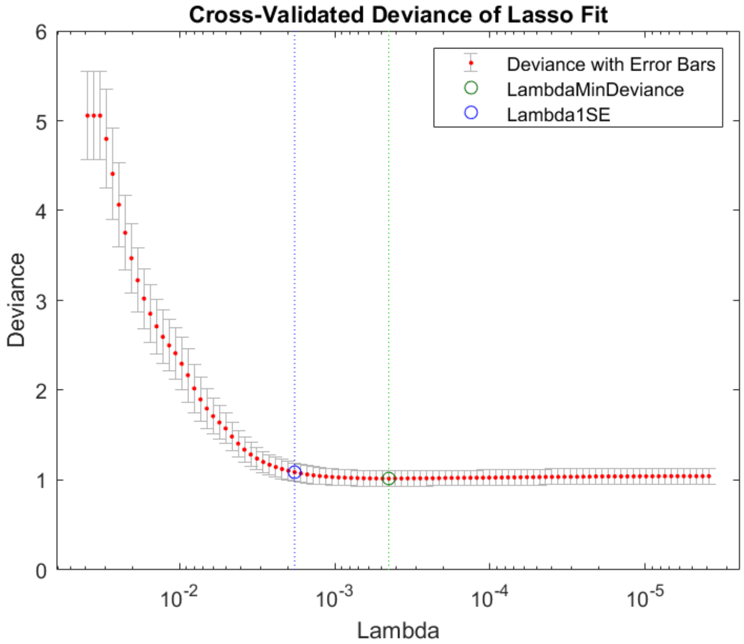
Cap-weighted = Weighted according to Market Capitalization
Log = Natural Logarithm
Diff = First difference to induce stationarity

Appendix 2

Lasso - Variable Selection (Inference)



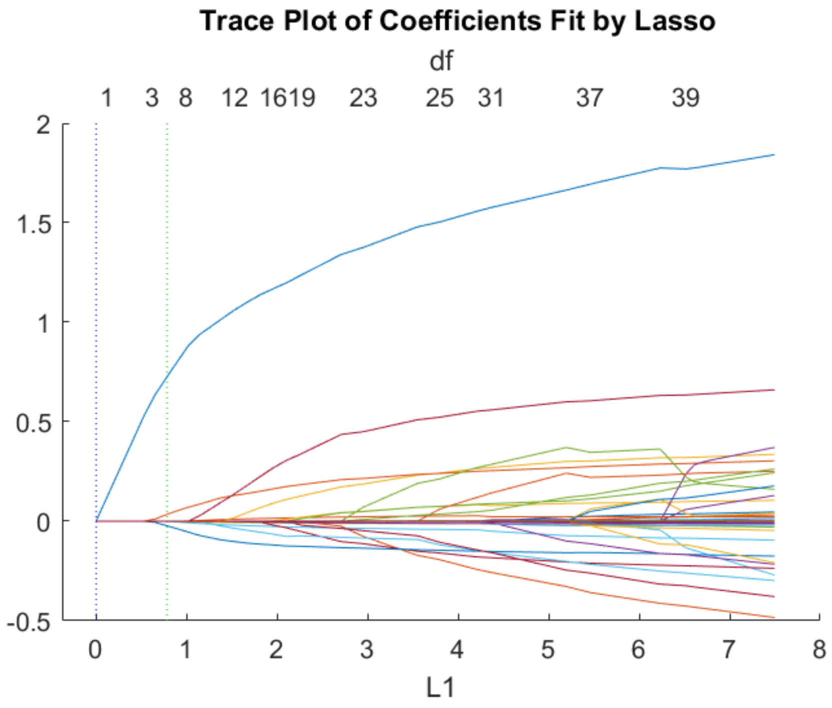
The *x*-axis from the L^1 norm of the coefficients. The y-axis the value of the coefficients. At the top of the *x*-axis the plot contains the degrees of freedom (df), meaning the number of nonzero coefficients: High, Low, Open, Market cap lag, DJI, Gold spot, USD-RUB, NASDAQ, IMOEX, VIX, Gas spot, Gas future, Volume (unfortunately some of them are not shown in the legend of the graph)



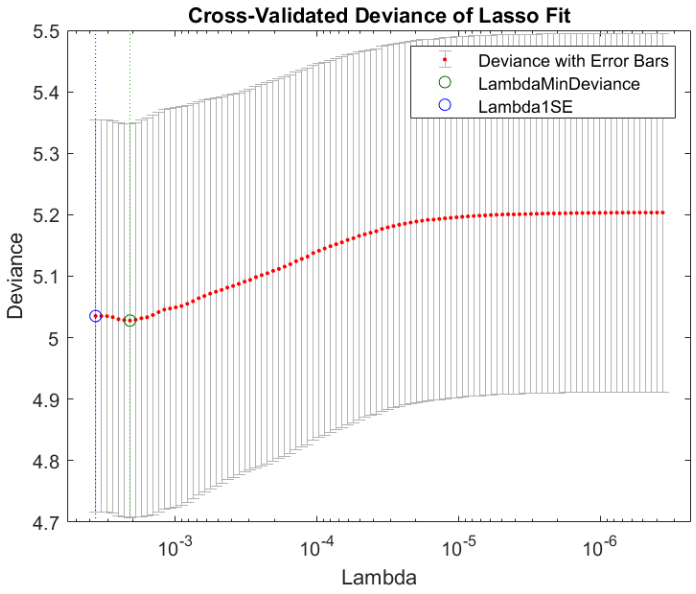
The green circle and dotted line locate the `Lambda` (0.00044) with minimum Cross-validation error.
The blue circle and dotted line locate the point with minimum Cross-validation error plus one standard deviation.

Appendix 3

Lasso - Variable selection (Prediction)



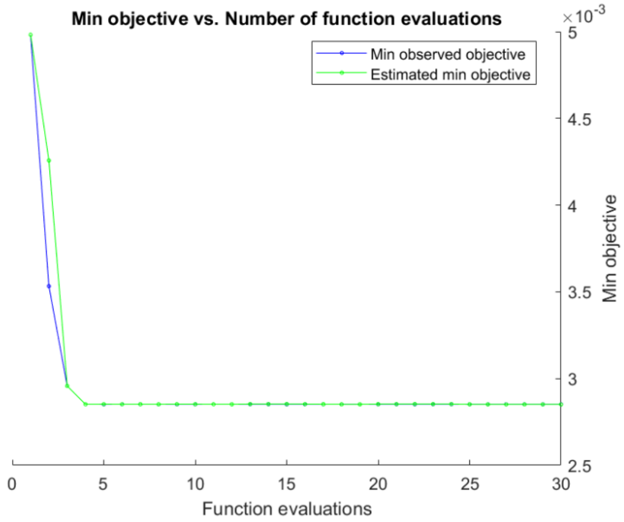
The *x*-axis from the L^1 norm of the coefficients. The y-axis the value of the coefficients. At the top of the *x*-axis the plot contains the degrees of freedom (df), meaning the number of nonzero coefficients: USDCNY, Nikkei, BSE Sensex, Market Cap, Gas spot



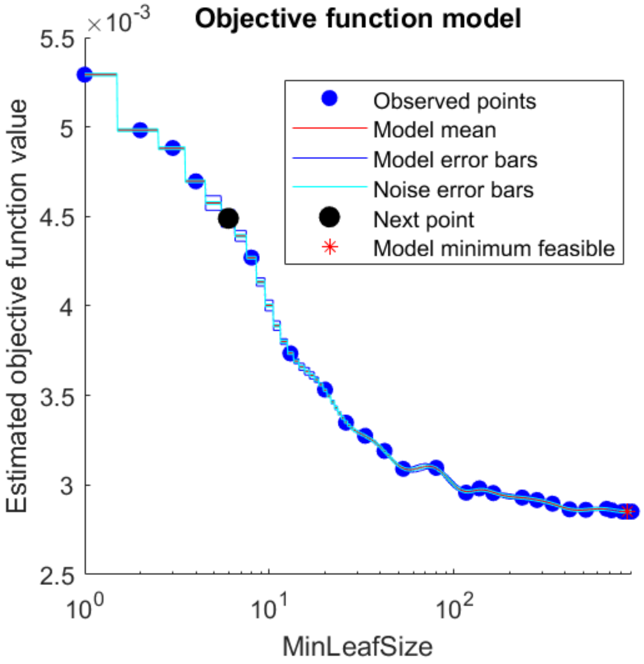
The green circle and dotted line locate the `Lambda` (0.002752) with minimum Cross-validation error
The blue circle and dotted line locate the point with minimum Cross-validation error plus one standard deviation.

Appendix 4

Regression trees - Hyperparameters Optimization all predictors model



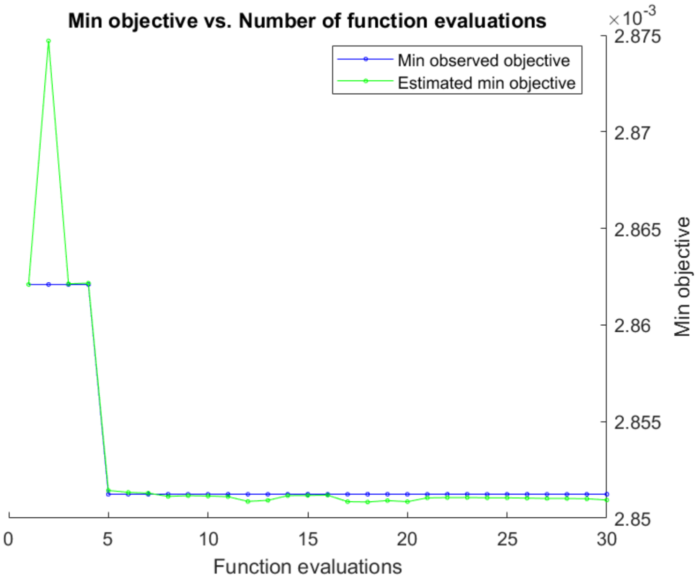
In the x-axis the 30 functions evaluated. The min objective in the y-axis refer to the min MSE
Best observed feasible point:
Observed objective function value = 0.0028518
Estimated objective function value = 0.0028511



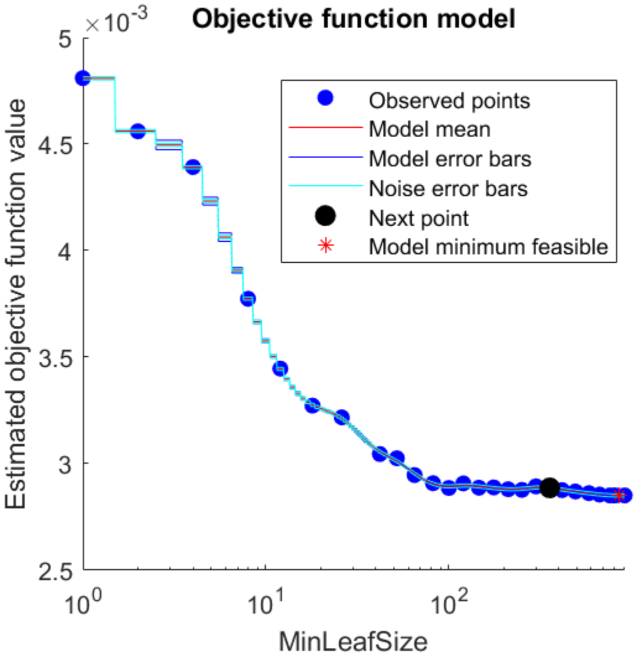
Best observed feasible point:
Min leaf size 880
Best estimated feasible point (according to models):
Min leaf size 880
Estimated objective function value = 0.0028511

Appendix 5

Regression trees - Hyperparameters Optimization Lasso-predictors model



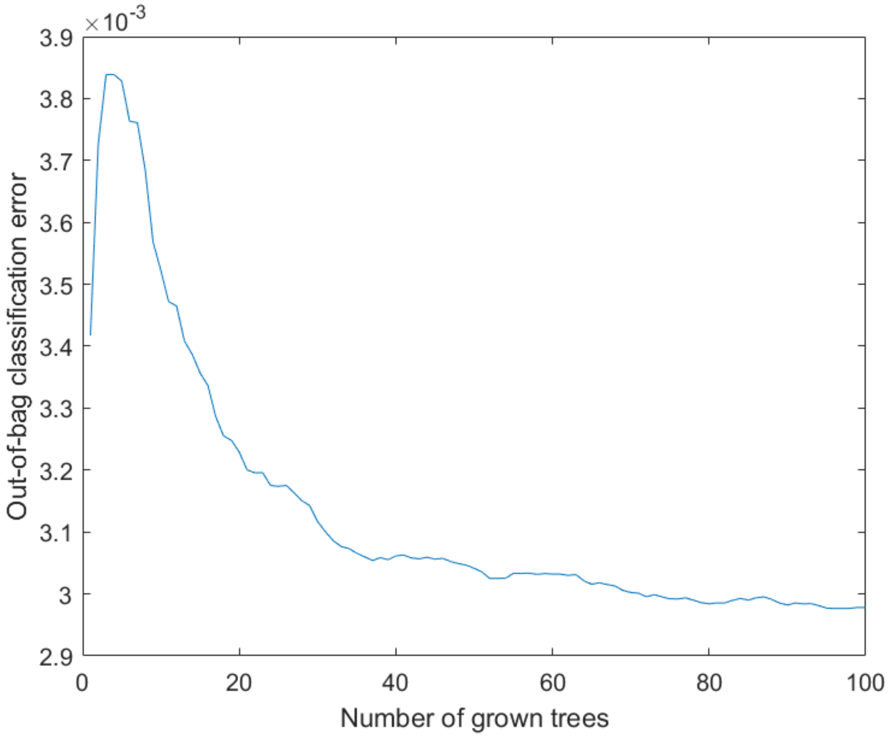
In the x-axis the 30 functions evaluated. The min objective in the y-axis refer to the min MSE
Best observed feasible point:
Observed objective function value = 0.0028512
Estimated objective function value = 0.0028509



Best observed feasible point:
Min leaf size 913
Best estimated feasible point (according to models):
Min leaf size 809
Estimated objective function value = 0.0028509

Appendix 6

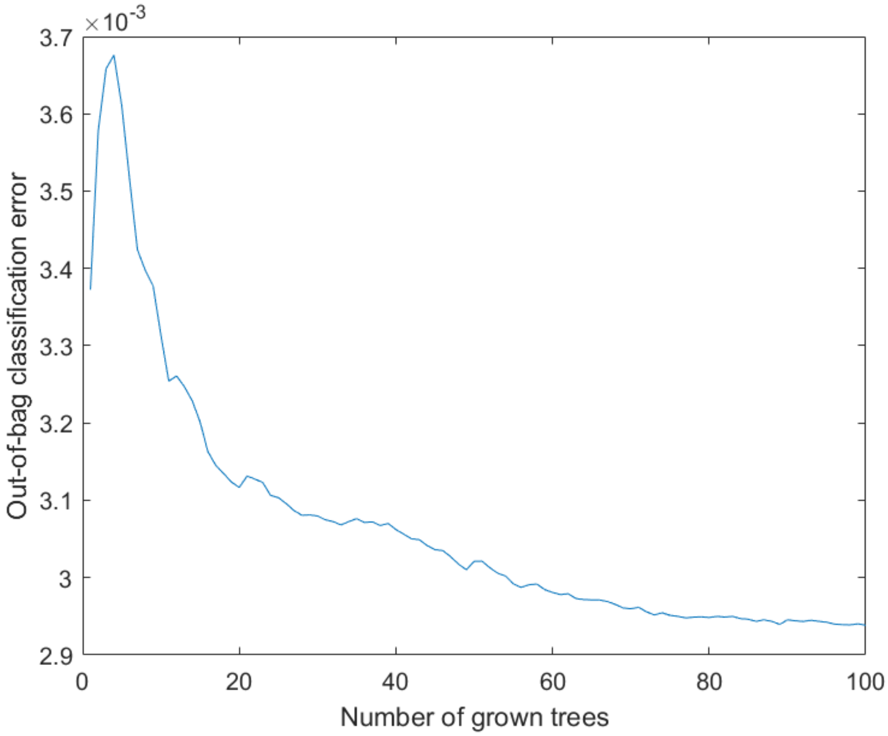
Random Forest - Number of tress all-predictors model



In the x-axis number of trees. In the y-axis the out of-bag classification error (MSE).
The graph shows that the 80 trees are enough to get optimal results.

Appendix 7

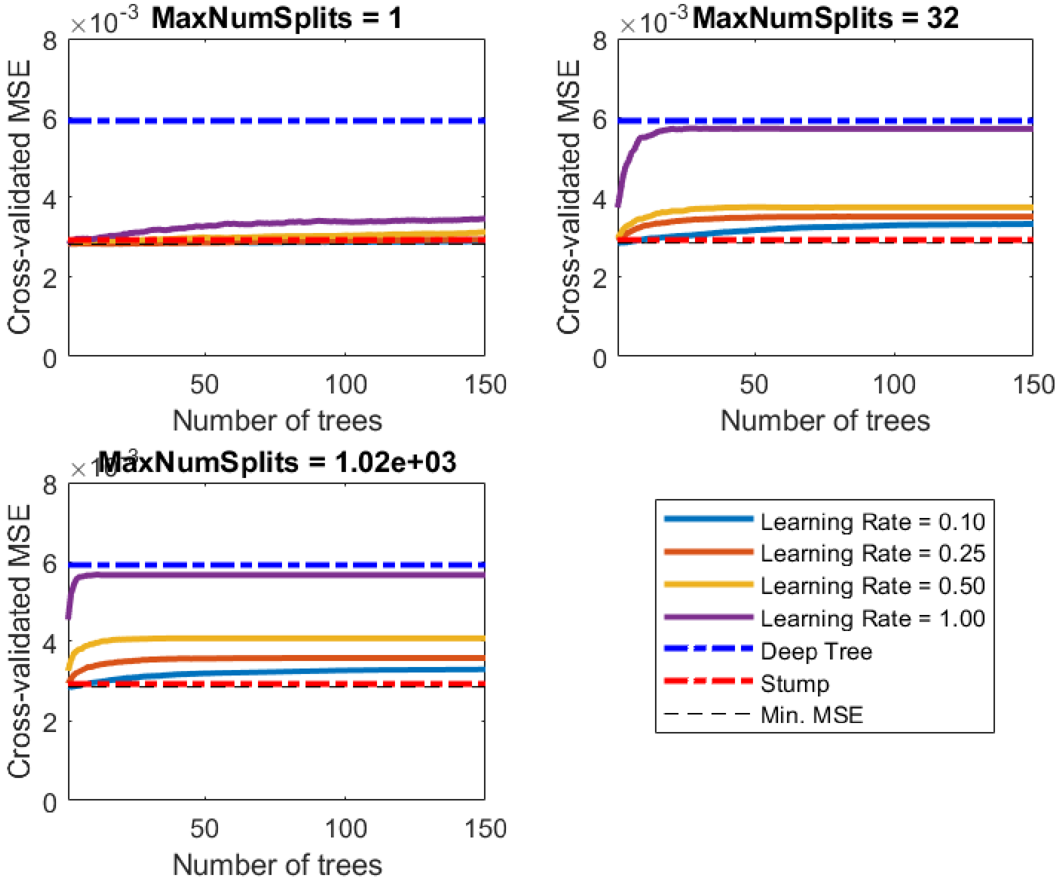
Random Forest: Number of trees Lasso- predictors model



In the x-axis number of trees. In the y-axis the out of-bag classification error (MSE). The graph shows that the 80 trees are enough to get optimal results.

Appendix 8

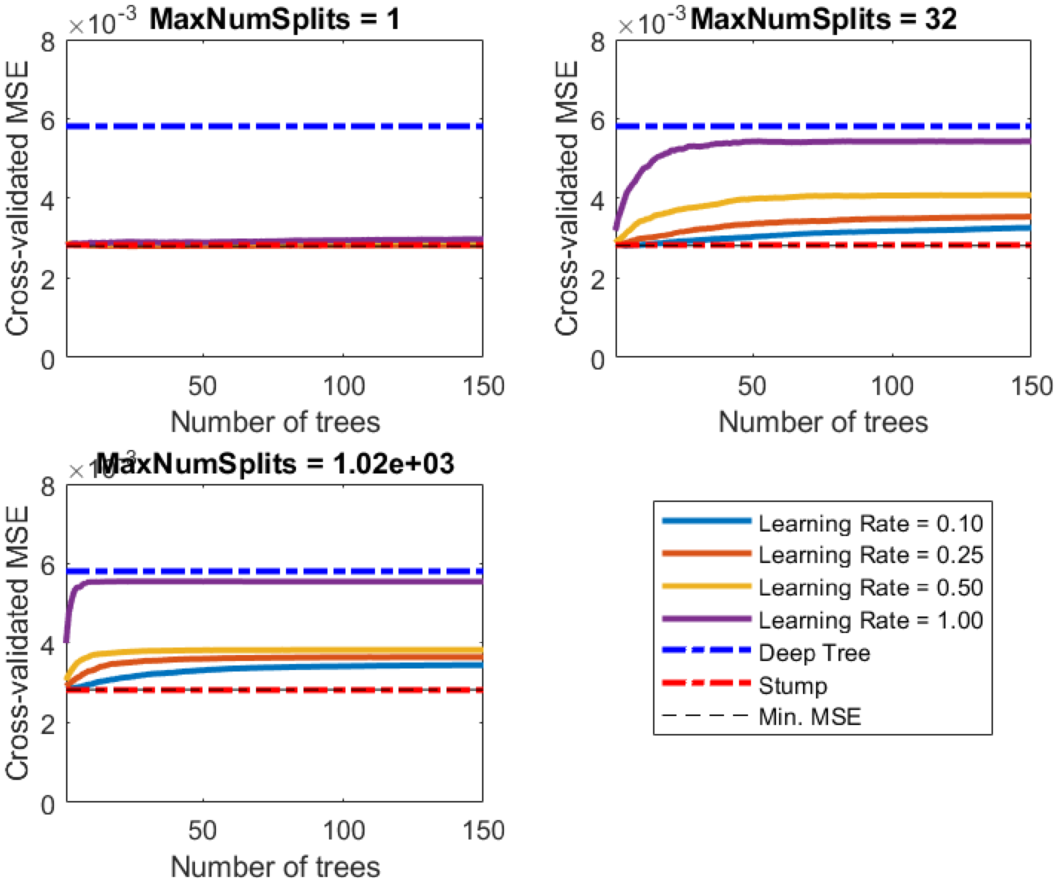
Boosting - Number of splits all-predictors model



In the x-axis the number of trees. In the y-axis the Cross-validated MSE. Cross-validation analysis to find optimal learning rate, deep of the tree and stump minimizing the MSE

Appendix 9

Boosting - Number of splits Lasso-predictors model



In the x-axis the number of trees. In the y-axis the Cross-validated MSE. Cross-validation analysis to find optimal learning rate, deep of the tree and stump minimizing the MSE