BACHELOR'S DEGREE THESIS

Degree in Statistics

Degree in Economics

Title: Social networks & price forecasting: The case of Bitcoins

Author: Adrià Aguiló Thorson

Statistics advisor: Dr. Karina Gibert

Department: Statistics and Operations Research (UPC)

Economics advisor: Dr. Montserrat Guillen

Department: Econometrics, Statistics and Applied Economics

Academic year: 2017 - 2018

UNIVERSITAT DE BARCELONA

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

SUMMARY AND KEY WORDS

SUMMARY

The main conceptual element this thesis orbits around is the idea of using social networks as a data source. First, classical trading theory and current usage of data obtained from social networks is reviewed. Taking all this information into account, a forecasting of the Bitcoin price is performed using both classical methods and machine learning Neural Networks. In order to obtain data from social networks, another complexity layer needs to be added by accessing the sources through APIs and directly web-scrapping the net. The results of all of this complex implementation are given with a strong focus on visualisation using several different techniques. Finally, after a critical discussion a Future Work chapter is introduced, where many possible follow-ups are drawn up.

KEY WORDS

· Bitcoin

· Social Networks

· Tweet scrapping

· Google Trends

· Arima

· Arimax

· Generalized Linear Model

· Artificial Neural Network

XARXES SOCIALS I PREDICCIÓ DE PREUS

El principal element conceptual al voltant del qual gira aquest Treball de Fi de Grau és la idea d'utilitzar les xarxes socials com a font d'informació. D'entrada, s'analitza tant la teoria clàssica d'inversió com l'ús actual de les xarxes socials com a font d'informació. Tenint en compte tot això, es procedeix a modelitzar i predir el preu del Bitcoin mitjançant tant mètodes classics com Xarxes Neuronals Artificials. Per tal d'obtenir dades a partir de xarxes socials, cal afegir una capa de complexitat al treball mitjançant l'access a les fonts a través d'APIs i directament scrapejant les webs. Els resultats obtinguts a partir d'aquesta complexa implementació es mostren en un format explícitament visual utilitzant diferents tècniques. Finalment, després d'una discussió crítica, es procedeix al capítol de Futur del Projecte on es plantegen varies possibles vies de continuació del treball.

PARAULES CLAU

· Bitcoin

· Xarxes Socials

· Tweet scraping

· Tendències de Google

· Arima

· Arimax

· Model Lineal generalitzat

· Xarxes Neuronals Artificials

AMS CLASSIFICATION

· 37M10 Time series analysis

· 62M10 Time series, auto-correlation, regression, etc.

· 62J12 Generalized linear models

· 91D30 Social Networks

· 97R40 Artificial intelligence

· 92B20 Neural Netowkrs, artificial life and related topics

ACKNOWLEDGEMENTS

This work would not have been possible without the support of my two advisors. Dr. Karina Gibert, Statistics Department professor, and Dr. Montserrat Guillen, Director of the Riskcenter, who have been supportive throughout the development of this thesis and who worked actively to provide me with a very valuable academic support to help me pursue this ambitious thesis.

I am also very grateful to Sudalai Rajkumar, a Data Scientist and Kaggle Grandmaster, who provided me Python scripts to perform web scrapping. His help was not only valuable but also inspiring, as selfless actions are scarce but very welcome.

Finally, I would like to say that nobody has been more important to me in the pursuit of this project than the members of my family. Specially, I would like to thank my grandfather, Dr. Ole Thorson, whose guidance is with me in whatever I pursue and has been the ultimate role model.

Table of Contents

1 INTRODUCTION

Last year I started investing in Bitcoin among other cryptocurrencies. Since then, I've been constantly amazed by its volatility. Mainly, because most of the time fluctuations would seem to have no particular reason and therefore I kept wondering, are people getting rich just by pure luck, or are there any patterns and relations that could be modelled?

Day after day, I kept wondering why our civilization is capable of sending humans to the moon but is fairly bad at forecasting economics related variables (1). As a Statistics & Economics student, I had to try (and potentially fail) by myself.

Since the beginning of our times, civilizations have always tried to predict its surroundings. For instance, Babylonians used the appearance of clouds and other factors to predict sudden changes in weather around the seventh century bce (2). Similarly, by 300 bce the Chinese ended up building a calendar where each season was associated with a different type of weather.

All having a thing in common, looking for patterns and establishing relations between observation and results. Currently, that idea hasn't changed much. In fact, the same paper that provided the Chinese and Babybolian history (Fast and Frugal Forecasting (2)) analyses some of their intuitive heuristics for forecasting, reaching the striking conclusion that these naïve methods can be equally good as knowledge-intensive procedures in some specific scenarios.

While forecasting techniques have evolved to a high degree of complexity, economists still struggle to predict changes in the economy. "Random Walk down Wall Street" (3) is a book about stock markets whose main argument is that stocks move in completely random ways. The consequence, is that the average investment fund got a lower return than the market's average. What this means in a straightforward way is that our progress on the forecasting field is very far away from being considered good.

A similar argument is done by an article (4) at Business Insider, where it can be clearly seen that most forecasts from prestigious institutions are almost worse than random chance.

After reading through all these different references, the mildest conclusion that can be reached is that there are serious doubts about how well the main economy indicators can be predicted.

At the end of the day, the economy is the result of every individual action, by each of the citizens of the planet. Therefore, the prediction of the economy could be done by adding up forecasts of each individual person in the world.

Could forecasts be perfect if every single person of the planet "told" what their intentions are? Of course, there would still be some variability sources like accidents and climate catastrophes, but it could definitely be an interesting approach to take.

While asking every person of the planet, or every stock investor, their intentions and feelings might not be feasible nor practical, there might be a backdoor of accessing a fairly good approximation, Social Networks.

On one hand, the usage of Social Networks has been increasing to the point that some networks have over two thousand million users, becoming a population-sized sample and therefore being representative. Specially, if we consider that the fraction of all citizens that invest in stocks is almost overlapped with the fraction of the citizens that are active users on social networks. On the other, in these networks, users, among other actions, post about their political views or their investing intentions, which ends up being very relevant information.

Therefore, through adequate gathering and pre-processing techniques, valuable information could be extracted from Social Networks. Hence, economy or stocks forecasts could increase in accuracy.

This idea is implemented in this thesis with one of the most volatile assets in the world, the Bitcoin, taking into account the debate that has been previously opened through this introduction.

The main objective is to find predictive methods that can properly fit the behaviour of Bitcoin and to consider different information sources that might be able to gather a global sentiment about a stock or an asset to be then applied to Bitcoin. The principal potential source is social networks, but also news outlets or other sources are considered.

This approach could be considered a part on the data-based strategical decision-making process, which has been gaining great importance in the last decade, supported by the Data Science conceptual framework.

The current thesis is structured around several chapters. It begins by analysing the State of the Art (chapter 2), a chapter in which several subjects are taken into account. It covers a wide area: from how social media is being used as a data source by other projects, to classical investment theory.

It then follows with the Methodology (chapter 3) that has been used in this thesis. This is a very important chapter as it contains the whole backbone of the techniques used. It has been structured around the five main branches of techniques used, which are data gathering, preprocessing, exploratory analysis, forecasting and goodness of fit. What is interesting about this structure, is that it follows the same chronology of the whole thesis.

The next chapters are the data related ones, which are Data (chapter 4), Preprocessing (chapter 5) and Exploratory Analysis (chapter 6). As it will be seen, there is a strong focus on visualization used not only as a tool to reach a wider scope but to be able to reach more complex conclusions.

It then proceeds with the main chapter of this thesis, Forecasting (chapter 7), which is where all forecasts are performed. Three forecasting techniques will be used: Arima, Arimax and Artificial Neural Networks. The implementation is done through two different techniques: Rolling Windows and Incremental Windows.

Finally, after a Critical Discussion (chapter 8) the thesis proceeds to the Future Work chapter (chapter 9). This is a very important chapter as the aim of this thesis wasn't to specifically obtain a good forecast of the Bitcoin price, but to give a deep thought on how different data sources can interact with both conventional and new forecasting techniques. Therefore, such a wide topic was not meant to be covered on a thesis, but to open up a critical reflection which I'll try to follow in future work, maybe during my master's degree.

One last relevant information before proceeding is that, in order to keep a coherent threat of thought, an appendix that gathers all non-essential information has been purposedly added after the Bibliography, as it could be a completely different document.

2 STATE OF THE ART

This chapter allows the reader to position themselves among three main knowledge areas which directly impact this thesis. Classical Trading Theory, the current usage of Social Media as a Data Source and Data Science as a conceptual framework.

2.1 Classical Trading theory

This section dives into classical trading theory, to provide a conceptual framework into the forecasting that will be performed in the following chapters. It is done by following the guidance of Technical Analysis of Stock Trends (5).

There are two primary methods used to analyse securities and make investment decisions:

- **Fundamental Analysis:** analysing a company's financial statement to determine the fair value of the business. This approach, when applied to a cryptocurrency, could be equivalent to study the potential future impact of the coin. Then estimate potential prices taking into account the finite number of coins among many other information. To sum up, this is a theoretical approach, that tries mixing information with expertise to estimate future valuation. Usually applied to the long term.
- **Technical Analysis:** statistical analysis of market activity, such as price and volume. It is mainly based on pattern identification, which can then be applied to perform price forecasting.
  It is mainly based on "history repeats itself". For example, as market sentiment shifts from optimism to pessimism, a certain pattern might show up before traders start selling their stock, and therefore cause a decrease on the price.

The current thesis follows the Technical Analysis branch. Which usually takes into account several concepts, which are trends, support and resistance, traded volume and patterns. Studying these concepts might be useful in further stages of variable selection and creation and therefore they will be explained below. In order to illustrate some of the concepts, images of the Bitcoin price from coinmarketcap will be shown.

Main concepts

The **trends** is the general direction towards where a security or market is headed. In cryptocurrencies it indicates where the price is moving towards. Several elements can be identified within a trend.

There are several kind of trends. An uptrend is classified as a series of higher highs and higher lows, while a downtrend consists of lower lows and lower highs. A trend can also be sideway or horizontal.

A trend can also have different lengths. In stock trading, usually the short-term is considered to be less than a month, intermediate-term is set between one and three months and long-term is usually on the year-range.

Usually, **trendlines** are drawn between the lowest lows and the highest highs to show general trend direction. As it will be explained in the following sub-sections, the usage of trendlines can allow assessing if there is a progressive decrease or increase of variance.

Support levels are prices where the price rebounds higher multiple times, whereas resistance levels are where prices rebound lower multiple times. The strength of support and resistance levels are determined by the number of rebounds from the trendline. For instance, the following figure illustrates a support level, around USD 6500, for the Bitcoin price in 2018. It can be clearly seen that the price rebounded many times before increasing again.

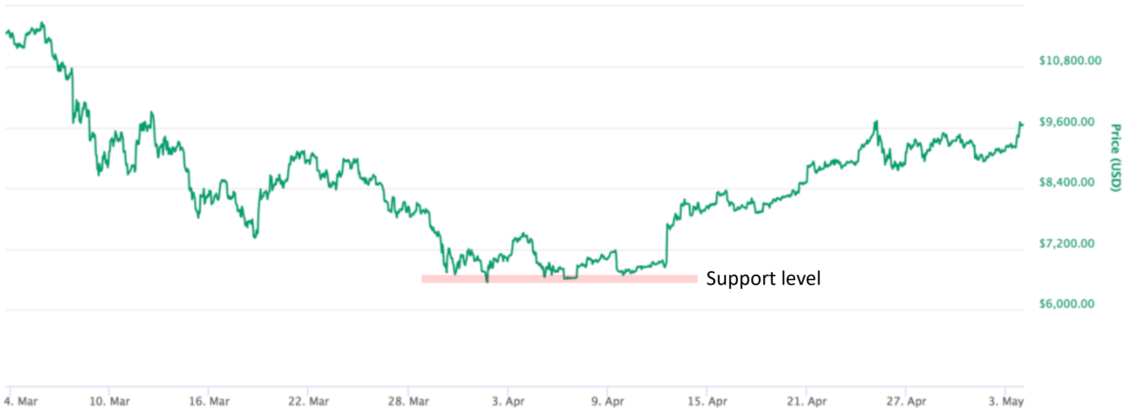


FIGURE 1 – SUPPORT LEVEL - BITCOIN PRICE 03/2018 - 05/2018

Support and resistance levels are psychologically-important levels and therefore a lot of buyers and/or sellers might be willing to trade the stock once it has been crossed. When the trendlines are broken, the market psychology switches and new levels of support and resistance are set. It has to be kept in mind, that sometimes the break is temporary, which is then called a false breakout/down.

Another important concept are **channels**. Which are two trendlines that act as strong areas of support and resistance with the price bouncing around between them until it breaks out beyond one of the two levels, in which case traders can expect a sharp move in the direction of the breakout. When the Channel is broken, the “broken” trend becomes the new support.

Finally, the last main variable **traded volume**, which can be defined as the number of shares that are traded over a given period. In this case, the number of traded Bitcoins, usually daily. The importance of volume can be found in many different scenarios:

· On a broken support/resistance situation, the strength of any given price movement is measured primarily by the volume. Low trading volume might indicate a false breakout/down.

· On the general trend, volume over time can be related to price trends to determine if a stock is gaining or losing momentum. For instance, if Bitcoin price has been trending higher with declining volume, which suggests that the rally may be losing momentum and therefore a price decline could be expected.

· Related to Chart Patterns that will be explained in the following sub-section, if volume is not present alongside these chart patterns, then the resulting trading signal is not as reliable as the case when the signal includes volume.

Patterns

Even though there are a lot of potential patterns on stock or asset trends, only the main will be shown here.

Heads & Shoulders:

A H&S pattern can be identified by locating a peak surrounded by two lower peaks. It usually indicates that there is a potential reversal of the trend. The lows are connected with a trendline, whose value is identified as the key support level. In this case, it can be seen that the H&S pattern does not have the expected result, as price keeps increasing afterwards.

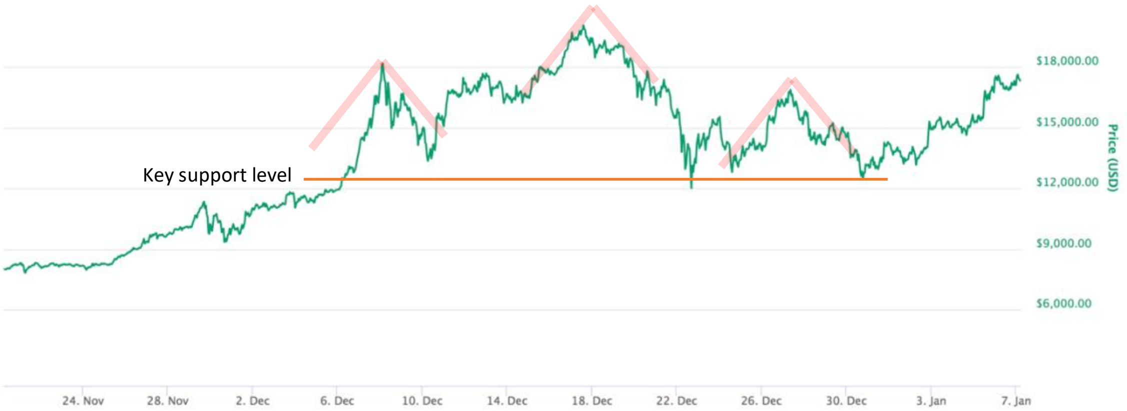


FIGURE 2 - HEADS AND SHOULDERS PATTERN - BITCOINC PRICE 11/2017 - 01/2018

Cup & Handle:

The C&H pattern can be identified by locating a wide U shape, followed by a smaller U shape. It usually happens amid an upward trend, that pauses on the smaller U shape, and then proceeds to keep growing. Therefore, it is a bullish continuation pattern. As it happened in the previous H&S pattern, this Cup & Handle from the Bitcoin price does not seem to follow any specific behaviour after it happened.

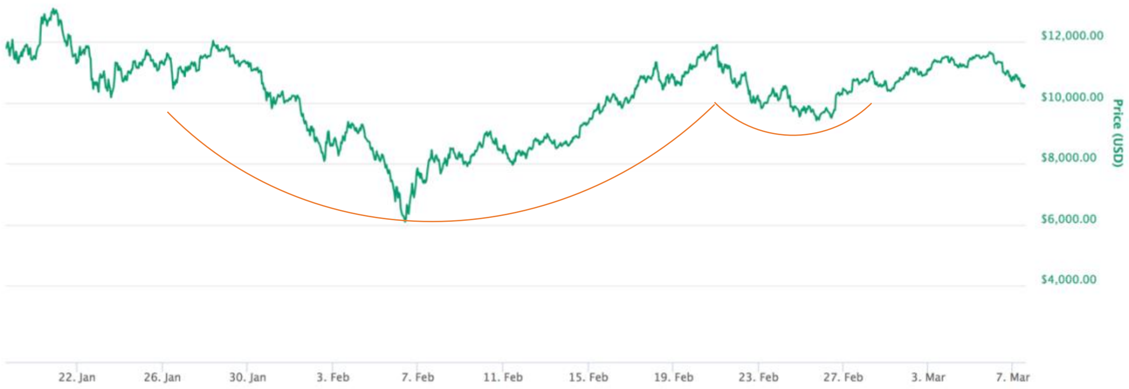


FIGURE 3 - CUP & HANDLE PATTERN - BITCOIN PRICE 01/2018 - 03/2018

<u>Triangles</u>:

Triangles are found when two trendlines converge to each other. If the triangle is asymmetrical, two main situations can be identified. If the flatter trendline is above, a breakout is likely. On the other hand, and as it can be seen in this case, if the flatter trendline is below a breakdown is likely. The lower lows have always similar values, while the highs are slowly decreasing.

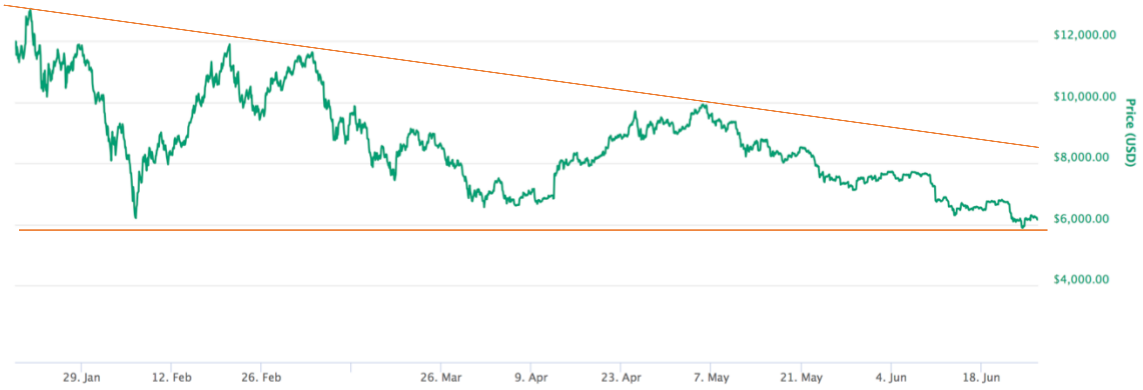


FIGURE 4 - TRIANGLE PATTERN - BITCOIN PRICE 01/2018 - 06/2018

Finally, there are some other relevant patterns such as Flags and Double tops/bottoms that will not be explained in this thesis as they are not essential.

2.2 Social networks as a data source

As people post more and more on social networks, the interest from both the academic community and enterprises to use them as a data source has risen. In this section of the thesis, some interesting examples about the value of information in social networks are presented.

For instance, a Penn State University professor, Conrad Tucker, has recently launched a program that has a clear aim, Predicting Threats to Society Using Data from Social Networks (6). The main ambition of the project is to increase the reliability that the information from these sources provides. In a way, it switches from focusing on the forecasting method to the algorithms that convert data from social network into valuable information. In fact, one of Tucker's quotes is especially relevant in this thesis:

> "If one CEO's tweet can send a stock's price down billions of dollars, that is a huge threat to the company..."

It is especially relevant because it shows a causality chain between social media and assets price. But while relevant people, such as CEO's, seem to have an impact that is easier to distinguish, the difficulty of the task increases when applied to, for example, the whole Twitter community. Then again, the aim is on distinguishing real and fake information, in order to turn the whole social networks' data into relevant inputs for a forecasting model.

Tucker's previously had participated on an on-going research that was based on the idea that social media networks could be used as a real-time sensor, for instance, to predict electricity consumption.

A similar approach is taken by a paper that aims to use social media as a data source for official statistics (7). Specifically, the study is designed to improve the forecast of the Dutch Consumer Confidence Index. To do so, a sentiment index is calculated from Facebook and Twitter public messages.

The conclusion it reaches is that predictions are more precise when using data from social networks. Moreover, the usage of the sentiment index would allow to obtain real time estimates which currently are out of reach due to the survey nature of the data.

Even some thesis have been written about the topic. For instance, engagement prediction is predicted by social networks information in a thesis (8) from the London Imperial College. In this case Facebook is the selected social network and one of the main reached results is that engagement decreases among users with active Facebook friends. The usage of information extracted from social networks is relevant yet again.

The conclusion of this section is double. On one hand there is a wide and growing interest in the topic, both in academic and official environments. On the other, projects on the subject seem to obtain good results when using data obtained from social networks. Therefore, the idea of using these new sources of information for Bitcoin forecasting seem not only adequate, but exciting.

2.3 Data Science

Data Science is the framework in which this thesis is built on, where all the trading theory and social networks ideas are carefully used to obtain relevant forecasts. Therefore, it is interesting to acknowledge that Data Science, as a fairly new field of knowledge, has not had a solid conceptual framework in which to work on.

This is why papers like Environmental Modelling & Software (9) help approach Data Science in an organised and agreeable procedure. Strategical decision making requires a robust implementation that properly handles complex and multi-source data, that allows to build both explanatory and forecasting models in a consensuated way.

Therefore, the referenced paper dives into how classical data analysis methods are frequently insufficient when facing several problems. It also outlines the added value of Data Science, which has been a key element on competition between big corporations.

The main point being that this thesis will follow the path provided by the paper in an effort to standardize the procedure from the data extraction to the conclusions, making the understanding of the procedure easier for any kind of reader.

3 METHODOLOGY

In this chapter the methods used during this thesis are outlined. Specifically, the techniques used to gather and filter data, the type of data analysis and the forecasting models used. In fact, most of the methodology structure is based on the outline provided by A survey on pre-processing techniques: Relevant issues in the context of environmental data mining (10). Specifically, pre-processing, data gathering and filtering, visualisation tools, missing data and data transformations are the sections that have been used.

Almost all the coding has been implemented with R. Therefore, all techniques explained in the following sections of this thesis will assume that an underlying implementation with R happened if the contrary is not stated.

The following figure shows the work diagram this thesis has followed. On the right, there are the main concepts around where the methodology is structured. As it can be seen, there are several data sources and gathering methods. From each, different preprocessing and exploration techniques have been used.

Finally, the dashed light-grey arrows show a possible follow up, which is explained in the Future Work chapter.

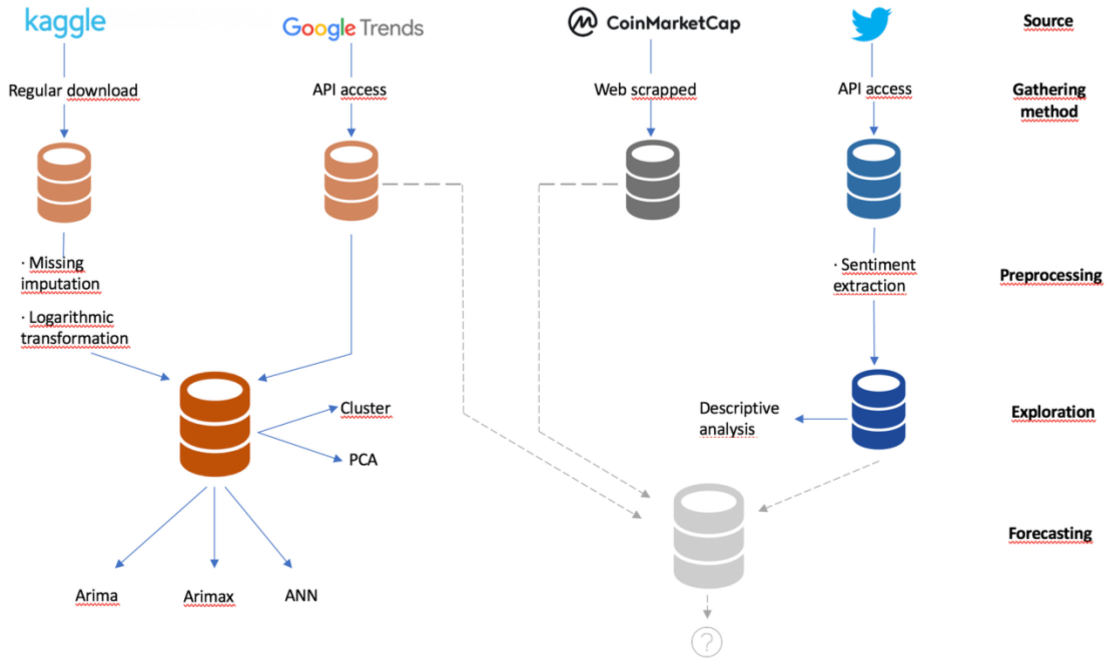


FIGURE 5 – DATA AND TECHNIQUES FLOW CHART

In the following sections, all of the elements from the figure above will be explained. Such explanation has been organised around 5 blocks: Data Gathering Methods, Preprocessing, Data Analysis, Forecasting and Goodness of fit.

3.1 Data Gathering Methods

The data used in this thesis have been gathered using three different methods. Directly downloaded from a website, accessed through an API or directly web-scrapped. The last two will be explained in detail below.

3.1.1 Data Retrieving API

An API, which is the acronym for Application Programming Interface, is an intermediary software that allows two programs to interact with each other. For instance, when a user scrolls through a website, the content they watch is being retrieved by an API from the company's server. In this case, API's are "hidden" below the browser interface.

Some websites, such as Twitter, allow developers to interact directly with some API's in order to, for example, retrieve specific data.

In this thesis, some information has been extracted directly from Twitter servers through their API, which works with the *rtweet* package (11).

3.1.2 Web Scrapping

Web Scrapping or Web crawling is a technique to collect data from websites usually through software that simulates a human surfing the web. It can be performed in a variety of ways and from different programs, but it is usually done with Python when it is done in a Data Science environment.

This is the case of this thesis, where the web scrapping was performed with a Python script. In order to do so, PyCharm[1] was the environment selected to implement the code.

3.2 Preprocessing

As it has been said before, the whole methodology chapter is mainly based on the guidelines provided by "A survey on pre-processing techniques". Not only it provides a theoretical framework but specifical steps to carry out pre-processing, which is why this paper is mentioned again in this section.

[1] Available online at: https://www.jetbrains.com/pycharm/

3.2.1 Missing Values

Missing values is a very sensitive issue faced by data scientists as when treated incorrectly can significative affect the conclusions derived from the data. Two typical errors faced by missing values is removing rows or substituting the column's mean.

Mean-substitution is a bad solution, as it can significatively decrease the sample's variance. This can lead to a wrong acceptance or rejection of statistical hypothesis among other problems.

Removing rows that contain missing values can create bias problems. This can happen if it is not a random phenomenon, but an specific sub-population is the one that creates the missing data. It can be even a bigger problem when working with time series data, where discontinuing the trend have many effects.

In order to avoid all of this techniques that might create relevant errors in the analysis, an adequate substitution technique had two be selected out of two main options. Univariate interpolation or a Multivariate Imputation algorithm. Both of which are explained below.

3.2.1.1 Univariate Interpolation

Applied to time series data, this technique consists on estimating the missing value from the past and following values of the same variable.

Let's assume there is a missing value of a variable *x* at period t. There could be several ways of estimating x_t.

One way could be using its immediate temporal neighbours, x_{t-1} and x_{t+1}. For example, it could be an average between x_{t-1} and x_{t+1}.

Another way could be making a regression taking into account, x_{t-i} and x_{t+i}, where *i* would set how many temporal neighbours are taken into account. Then, the estimated x_t value would be obtained from the regression $x_{t+i} = \alpha + \beta(t+i) + u_{t+i}$ setting *i* to 0.

While Univariate Interpolation can be a very useful tool for time series, taking into account other variable's behaviour can be also adequate and maybe even better. Therefore, in the presence of a multivariate dataset, Multivariate Imputation techniques should at least be considered.

3.2.1.2 Multivariate Imputation

While the literature for missing values imputation is abundant, it is not always clear which techniques should be used. Amid huge datasets and demanding timelines, sometimes the criterium for choosing a technique seems to be computational time.

Luckily, and as it will be seen in the following chapters, this thesis' dataset had a very low number of missing data and in only one variable. As a consequence, a computationally demanding algorithm could be chosen. In this case, Predictive Mean Matching (PMM from now on) was chosen.

3.2.1.3 Predictive Mean Matching

PMM, is an interesting algorithm in the sense that it does not generate values, but it builds a metric that allows pairing cases with missing data to similar cases with data present.

The algorithm functioning is explained step by step (12), assuming a scenario where only one variable (x) has missing data and there is a set of variables without missing values (z):

i. PMM begins making a subset of the dataset, excluding all rows without data on x. Then a linear regression is estimated of x on z. Obtaining a set of coefficients b.

ii. In order to add enough variability in the imputed values, a new set of coefficients b* is generated by making a random draw from the posterior predictive distribution of b.

iii. Using b*, an x value is calculated for all rows, including the ones without missing values.

iv. For each row with a missing value on x, locate a set of rows whose predicted value from b* is close to its real value.

v. Among those cases, one is randomly selected. Its observed x value is then imputed to the case that had the missing value. Therefore, it is a real value in the sample that is being imputed.

vi. Steps from ii to v are repeated until there are no more missing values.

In its R implementation through the *mice* package (13), a parameter *m* can be modified. This parameter indicates the number of multiple imputations, which means how many cases (*m*) should be in each match set (step iv). In this thesis *m* has been set to 5, which is R’s default. What setting *m* to 5 means, is that the algorithm matches each row with missing data with the 5 cases that have the closest predicted value. Then chooses one randomly.

3.2.2 Multigranularity

Granularity can be defined as the level of detail in which data are represented. Some real-life applications require dealing with data at different granularities. Applied to time series, high granularity could mean data cases every second, while low granularity data cases every day or year.

When working with different sources of information on a time-series environment, multigranularity can arise. Which is the problem of having data collected at different time frequency. Usually, it is solved by two ways:

i. The low granularity variables are repeated to match the length of the higher granularity data.
ii. The high granularity data are summarized to fit into the low granularity scope.

As it will be seen in the exploration of the variables, this current thesis works from three different databases, all of which have different granularities.

3.2.3 Merging Datasets

The merging of the different data sources has been done by the *sqldf* package (14), which allows to use SQL language inside R's code.

The merge has been done taking into account the multigranularity present in the used data sources, repeating low granularity data & averaging or summarizing higher granularity data in order to obtain an homogeneous data base to properly work on.

3.2.4 Logarithms vs Growth rate

In economic theory, natural logarithms are used when modelling growth rates due to its ease of calculation and versatility. In the following proof it will be seen that this is an adequate procedure for low growth rates.

Growth rate "g":

$$\frac{x_{t+1} - x_t}{x_t} \rightarrow x_{t+1} = (1+g)x_t$$

If we take n periods:

$$x_{t+n} = (1+g)^n x_t$$

When calculating average growth rate:

$$y_{t+n} = y_t(1+g)^n \rightarrow (1+g)^n = \frac{y_{t+n}}{y_t} \rightarrow 1+g = (\frac{y_{t+n}}{y_t})^{1/n}$$

Calculation with logarithms in the following page.

Average growth rate:

$$y_{t+n} = y_t(1+g)^n \rightarrow ln(y_{t+n}) = ln(y_t) + nln(1+g)$$

For small growth rates:

$ln(1+g) \approx g$. For instance, ln(1.05) = 0.0488.

The higher the growth rate, the worse the approximation. For instance, ln(1.8) = 0.5878.

This reasoning will be empirically checked in the data analysis chapter of this thesis to guarantee the right transformation is applied.

3.2.5 Sentiment Analysis

The term of Sentiment Analysis is used to describe the usage of several techniques in order to identify subjective information form a data source. When applied to texts, the goal is to identify the connotation of a text, which can be called its sentiment.

As with many Machine Learning algorithms, a supervised learning approach can be taken if the texts are rated. And therefore the algorithm will be trained to predict the score from the words and possibly the relationship between them.

On the other hand, if there are no ratings for the texts, the application of ML algorithms becomes harder and usually more basic approaches are taken. One possible way is to use a library which contain all existing words in a determinate language and has assigned a rating or a feeling to each of them.

This simple approach can be implemented in R with the *get_sentiments* function from the *tidytext* package (15). There are several ways of classifying the feelings. In this thesis, *nrc* and *afinn* sentiment libraries, called lexicon, have been used.

The **NRC** lexicon has 8 possible feelings "anger", "anticipation", "disgust", "fear", "joy", "sadness", "surprise", or "trust". Each word of the library can have more than one feeling associated. For instance, the word *abandon* has “fear”, “negative” and “sadness” feelings associated. And *ability* has only one associated feeling, which is “positive”. Finally, a word like *fraud* has two associated feelings, which are “anger” and “negative”. The figure in the following page shows the frequency of each feeling in the nrc classification.

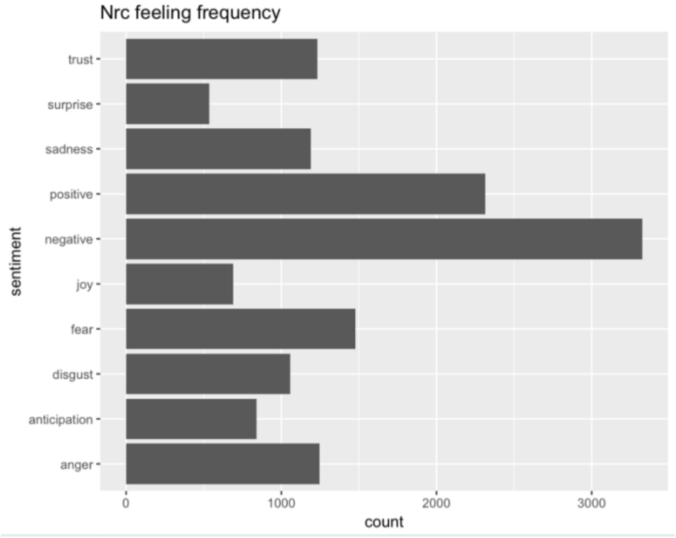


FIGURE 6 - NRC FEELING FREQUENCY

The **afinn** lexicon gives each word a score, which ranges from -5 (being the worst feeling possible) to 5. In this case, each word has only one associated score. Taking the same words as before, *abandon* has an associated score of -2. The word *ability* has a score of 2, and the word *fraud* has a score of -4. The following figure shows the distribution of scores of all words on afinn.

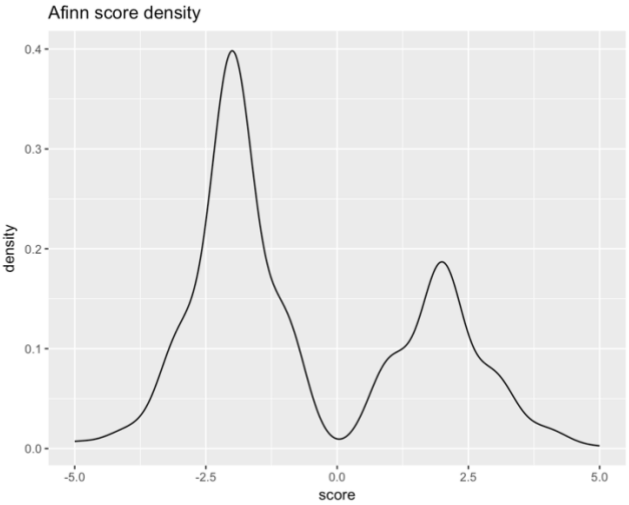


FIGURE 7 - AFINN SCORE DENSITY

Before proceeding to the next stage, it is important to highlight that the limitations of using these libraries are wide and should be used carefully. Specially, because interactions between words will not be taken into account.

3.2.6 Tweet Mining

Once Sentiment Analysis can be calculated for each word, this thesis suggests several ways of calculating the overall Sentiment of each tweet. In this thesis, it has been done through two approaches.

Numerical approach

The first way is adding up all the scores of the words of each tweet (using the numerical scores from afinn). Tweets of an overall score of over 1 will be classified as positive and under -1 as negative. The remaining tweets will be treated as neutral.

Or overall score is calculated for each tweet, and kept as numerical. Therefore, the higher value, the more positive a tweet is and viceversa.

Categorical approach

Each word has one or more associated feelings (using the categories from nrc). Then, for each tweet, all associated feelings from all word of the tweet are counted. Then, for each tweet only the highest frequency sentiment is kept.

3.3 Data analysis

3.3.1 Clustering & Profiling

On a Data Science approach, Cluster Analysis is understood as a technique that identifies groups of cases that are similar among them. Usually, clustering algorithms have a double objective: minimize variability intra-group and maximize variability between groups.

Clustering can be performed with several techniques, depending on the dataset and the aim of the project. Usually, all the techniques have to define to parameters. 1 - The distance between observations, which shows how similar are cases. 2- The kind of cluster. 3 – The aggregation criterion.

3.3.1.1 Distance Metric

In order to calculate the distance between observations, the package used has been *cluster* (16). It contains a function called *daisy* which calculates a Dissimilarity Matrix. Which obtains, as stated in the package, “all the pairwise dissimilarities (distances) between observations in the data set”.

As all variables are numeric, there is a wide range of options to choose from regarding the distance metric, Euclidean and Manhattan might be two of the most popular ones.

The Euclidean distance is defined in R's *daisy* function as "root sum-of-squares of difference", which can be algebraically expressed as $d = \sqrt[2]{\sum_{i=1}^{n}(x_i - y_i)^2}$.

The Manhattan distance metric is also defined in R's package, this time as "sum of absolute differences". Which can be algebraically expressed as $d = \sqrt[2]{\sum_{i=1}^{n}|x_i - y_i|}$, where *n* is the number of observations.

Theoretically, it could be said that Manhattan distances are more robust to outliers and should be used in high variability data.

3.3.1.2 Types of Clustering

Clustering techniques can be classified in two main subgroups.

Soft Clustering: a probability of belonging to a cluster is assigned to every data point.

Hard Clustering: each unit is classified in a single cluster. A specific case of this is hierarchical clustering, which is implemented with R's *hclust* function from the *stats* package (17). It forms hierarchical groups of mutually exclusive subsets and is usually implemented using Ward's method.

3.3.1.3 Ward's Aggregation Criterium

Finally, once the distances between all cases have been calculated and the type of clustering has been chosen, the data points have to be aggregated into the clusters.

The selected function, *hclust*, allows to calculate an agglomerative hierarchical cluster through Ward's aggregation criterium (18).

In each iteration, the pair of clusters to merge is decided by how much does the objective function decrease. As the objective is to obtain clusters with minimum intracluster variance, the element most similar to the others on the cluster will be the one added.

Once all the possible mergings have been calculated, the number of clusters is decided conceptually by observing the hierarchical cluster plot. Usually this is done by "cutting" the longest branch, which will result in the most different clusters between them.

3.3.2 Dimensionality reduction

There are several techniques to reduce dimensionality. In this case, and as all the variables are numerical, Principal Component Analysis has been chosen.

PCA (19) is one of the key techniques of explicatory multivariate analysis. It allows to obtain a plane (or subspace) that retains as much information as possible from the original hiperspace. Therefore, multivariate and non-linear relations can be identified in a straightforward way. Which would be impossible using other exploratory techniques.

Parting from an only numerical and centered dataset, transformations are calculated to obtain its eigenvalues and eigenvectors through diagonalization. Each eigenvalue has its correspondent eigenvector, and it indicates the percentage of variability gathered in each principal component (the associated eigen vector provides the direction of this principal component). Therefore, once ordered decreasingly, the first two eigenvalues will have the eigenvectors that will create the most informational two-dimensional subspace. This subspace will be the one chosen to project all data rows, in order to get an overall glimpse of how the data behaves.

3.4 Forecasting

The current thesis has chosen not to forecast with just one forecasting methods, but with several. This has been thought this way in order to obtain a wider possible interpretation. The selected models are a classical Arima, an Arimax and finally an Artificial Neural Network. It is true that ANN are not designed to make time-series forecasting and this problem will be discussed latter on in this thesis.

As the aim was to make short term forecasting, a Rolling Window approach has been selected which has allowed to perform hundreds of predictions from each day of the train dataset.

This approach has been implemented in two different ways, a Rolling Window (RW) and an Incremental Window (IW), both of which are explained below.

3.4.1 Rolling Window

Forecasting with a Rolling Window (20) approach allows to assess the stability of the model over time. This can be evaluated taking to ideas into account: 1 – If coefficients are time invariant 2 – Errors are time invariant, both in average and in variance.

For each rolling window subsample, a model is estimated, and then 1 to 10 steps ahead forecasts are estimated. Therefore, in the following scheme h=1:10. As it will be seen in the forecasting chapter of this thesis, 250 rolling windows have been used, and within each rolling window 10 forecasts have been estimated.

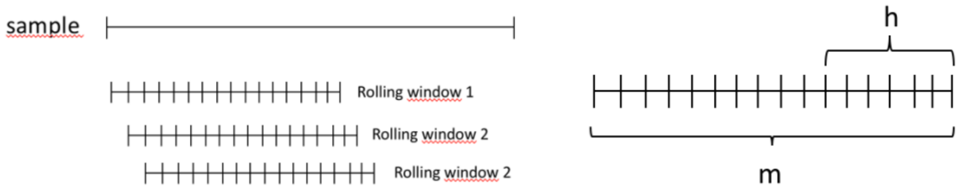


FIGURE 8 - ROLLING WINDOW SCHEME

3.4.2 Incremental Window

All of the concepts introduced in the Rolling Arima chapter are still valid for the Incremental one. The only difference is that in this case the model will have infinite memory. This means that every iteration our train dataset has one more observation than the iteration before.

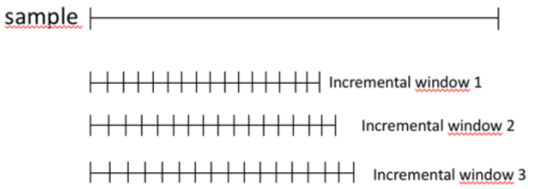


FIGURE 9 - INCREMENTAL WINDOW SCHEME

The expected impact of using an Incremental Window, is that the forecasts should adapt more slowly to slope changes than the Rolling Window implementation. This assumption will be discussed again in the model comparison section (7.4).

3.4.3 Arima

Arima models (21) which are Autoregressive Integrated Moving Average, allow to estimate forecasts based merely on a series of the variable that is being forecasted. To do so, it first calculates how many differentiations should be performed. And then it is calculated how many lags of the variable and the errors should be taken into account.

The form of an ARIMA(p,D,q) usually is as shown in the following expression:

$$\Delta^D y_t = c + \phi_1 \Delta^D y_{t-1} + \ldots + \phi_p \Delta^D y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}$$

where $\Delta^D y_t$ denotes a Dth differenced time series of y_t, and ε_t is an uncorrelated innovation process with mean zero. Differentiation is defined as:

$$\Delta y_t = y_t - y_{t-1}$$

The p,D,q structure can be calculated through a conceptual approach or through the *auto.arima* function from the *forecast* package (22). In order to keep the methodology chapter brief, the steps to decide which is the optimal structure will be directly explained in the Forecasting chapter.

The Arima(p,D,q) structure will be calculated with historical data once, and will be kept through all the Rolling and Incremental Windows. Therefore, only the parameters will be re-estimated in each iteration. The two main causes that lead to avoiding the recalculation of the structure are:

1- The auto.arima function has a high computational cost.
2- Even if it had not, the goodness of fit is higher if (p,D,q) is selected conceptually in this case.

3.4.4 Arimax

Arimax Models are Autoregressive Integrated Moving Average with Exploratory Variables. As the name suggests, they add exploratory variables to an Arima Model such as:

$$\mathit{\Delta}^D y_t = c + \beta X_{t-1} \; + \phi_1 \mathit{\Delta}^D y_{t\,\_\,1} + \cdots + \phi_p \mathit{\Delta}^D y_{t\,\_\,p} + \varepsilon_t + \theta_1 \varepsilon_{t\,\_\,1} + \cdots + \theta_q \varepsilon_{t\,\_\,q}$$

Where X are all other variables.

In fact, the current thesis has added lags also to X. And therefore the notation would be slightly more complex. The way the lags have been added is explained below.

3.4.5 Rolling Window implementation

Before proceeding any further, it is necessary to highlight that an Arima model that is being implemented through a Rolling Window has two options, recalculating in each window both the (p,D,q) structure and the parameters, or only the parameters.

In this thesis, the former option has been selected, again, due to computational constraints. As a consequence, both in the Incremental RW and the regular RW, have their structure calculated with the historical data.

Finally, in the RW implementation section it is necessary to explain the usage of the two RW with each forecasting technique.

Arima and Arimax will use both Incremental and regular Rolling Windows while ANN will be used only with an incremental rolling window. This is because, as it will be explained in the ANN section, they need large amounts of data to work properly.

3.4.6 Lag creation & variable selection

In order to create lags for all variables at once, two functions have been used altogether: *lag* from the *stats* package and mutate from the *dplyr* package (23).

After several lags are created for every variable, a stepwise procedure in a Generalised Linear Model has been applied in order to select the most relevant lags for each variable and if some variables should be directly ignored. This first variable filtering allows to reduce future computational time in the forecasting models and also filters potential noise sources.

It is important to highlight that the glm model has not been used to make forecasts, but just to select the adequate number of lags. The stepwise procedure, implemented through the *step* function in the R's *stats* package (17), tries all possible combinations of variables in order to obtain the minimum Akaike Information Criterion.

3.4.7 Artificial Neural Networks

ANN are brain-inspired computing systems that specially resemble synapses, which is the structure that allows a neuron to pass an electrical or chemical signal to another neuron. ANNs are built by adding layers of connected nodes. All inner layers are called hidden layers, while the first is the input layer and the last is the output layer. The structure and the way it functions will be explained thoroughly below with information extracted from a book written by MIT professors called Deep Learning (24).

While training the neural network, the algorithms aim is to approximate a function *f(x)* to the function defined by the available data *f*(x).* The model can be represented in an non-cyclical graph that describes the transformation to each observation until it is transformed into the response variable.

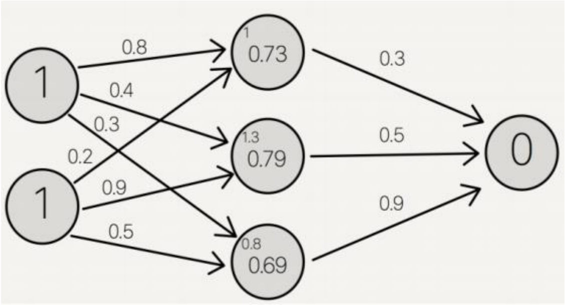


FIGURE 10 - EXAMPLE OF POTENTIAL VALUES THAT COULD BE OBTAINED THROUGH THE LAYERS OF A NEURAL NETWORK. IMAGE OBTAINED FROM (24).

Let's assume there is a response variable and a set X of exogenous variables. At the left of the figure, there is the input layer. Each of the units represents a variable from X. Therefore, in this case X would have only two variables.

On the right, there is the output layer, which represents the response variable. The values obtained in this layer, will be the estimated value of the response variable for each row.

As it been said before, all layers between the input and the output layers are named hidden layers. This is because the values of this layers are not from observable events, but mere transformations of the input layer.

Each unit of a hidden layer represents an aggregation of information given by the input data (or the previous hidden layer if there is more than one). This aggregation is the result of applying a non-linear activation function to the linear combination of the resulting values of the previous layer with a set of associated weights. This way, each units allow the model to capture interactions between variables and therefore, the more units, the higher number of interactions that can be captured.

The number of layers of the neural network is what defines the depth of the model. This concept is where the term Deep Learning comes from.

Finally, before proceeding to explaining the details of the ANN used in this thesis, a global scheme of a multilayer neural network is shown.

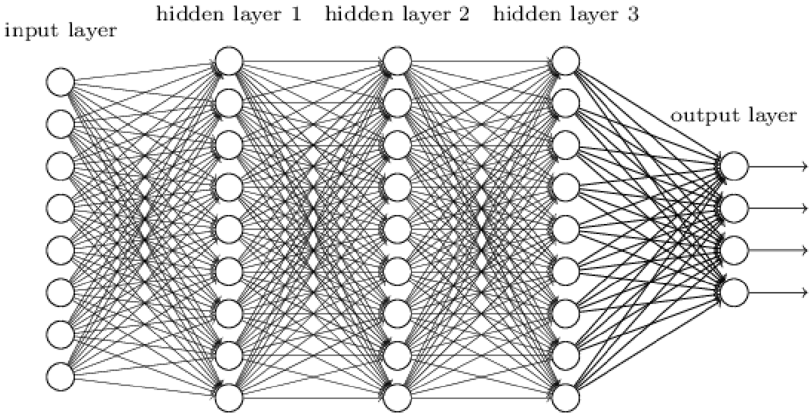


FIGURE 11 - NEURAL NETWORK WITH THREE HIDDEN LAYERS. IMAGE OBTAINED FROM (24).

In this thesis, neural networks will be trained using the train function from the *caret* package (25). It allows to train a wide range of machine learning algorithms. Moreover, it also gives the option to tune hyper-parameters through the *expand.grid* function.

In this thesis, the "nnet" method was selected, which fits a single hidden layer neural network. Two hyper-parameters can be tuned, size and decay. Size represents the number of units in the hidden layer and decay is a regularisation parameter to avoid over-fitting. The specifics about the final hyper-parameter selection will be explained in the specific section on the Forecasting chapter.

3.5 Goodness of fit

Two simple measures will be used to numerically assess the goodness of fit for each model and forecasting scope.

On one hand, the Mean of the Absolute Error will be calculated. Where p_t is the value to be predicted and $\widehat{p_t}$ its prediction, n is the number of observations.

$$MAE(\hat{p}) = \frac{1}{n}\sum\nolimits_{t=1}^{n} |\widehat{p_t} - p_t|$$

On the other, a variability measure will be calculated over the absolute errors. Calculated as it is shown below:

$$V\big(MAE(\hat{p})\big) = \sum\nolimits_{t=1}^{n} \big(|\widehat{p_t} - p_t| - MAE(\hat{p})\big)^2$$

More importantly, graphical techniques will be used to analyse how does the goodness of fit change over time and how it distributes through the different forecasting scopes.

Some of the main graphical tools used will be the display of errors through time by scope, density of errors by scope, cumulative error and parameter stability.

4 DATA

4.1 Sources

The data used in this thesis has been gathered from three sources. Kaggle, Google Trends and Twitter. Both the sources and the gathering methods are explained in detail in the following sections.

4.1.1 Kaggle

Kaggle[2] is a platform for Data Science competitions. It takes an open source approach where everybody can upload their codes. Being a very valuable source of knowledge for data scientists.

Moreover, users can also upload data bases. This leads to talk about Sudalai Rajkumar, a Data Scientist and one of Kaggle's main contributors. Mr. Rajkumar scrapped data from a cryptocurrencies exchange, coinmarketcap.com[3], and uploaded it to the community of Kaggle.

The dataset which had Bitcoin's related variables was exactly what was needed for this thesis. The only problem was that to continue this project once the thesis had been handed, more recent data would be needed to merge the data with other recent variables that will be explained in the following chapters.

To solve this problem, I contacted Mr. Rajkumar and he provided a Python code, so I could perform directly the scrapping from coinmarketcap.com. Thanks to this, while the data used in the current thesis is the one regularly downloaded from Kaggle, the project will have the necessary tools to keep moving forward.

4.1.2 Google Trends

Google Trends[4] is a website based on Google Search. It allows to obtain the relative number of searches out of the total search-volume of the topics introduced. Same searches in different languages are identified as the same, and therefore it provides the interests across the globe. Finally, it allows to select determinate regions and time period. The abscissas axis represents time, while the ordinate axis shows the relative interests of the topics. The highest number of searches in a week is set to 100, and all other values are escalated accordingly.

The results can be both downloaded directly as a csv file, or accessed through its API. When using R, the API can be accessed through *gtrendsR* package (26). In this thesis, the data was accessed directly from the website.

[2] Available online at: https://www.kaggle.com (30)

[3] Available online at: https://www.coinmarketcap.com (31)

[4] Available online at: https://trends.google.es/trends/?geo=ES (32)

Finally, it is necessary to point out that if the relative search interest is below 1, Google just labels it as “<1”. In the current thesis, the cases where this happened, a value was imputed randomly following a Normal Distribution with mean 0.5 and standard deviation 0.1.

4.1.3 Twitter

Finally, the last data source has been obtained by scrapping tweets from Twitter, one of the main social networks according to Statista (27).

Let’s begin by analysing Twitter. The network allows its users to send short and plain messages (280 characters). Texts are typically joined by hashtags (#), these hashtags work as labels that allows users to find topics that they are interested to.

By default, the messages are public. Other main networks such as Facebook and Instagram are more media-sharing oriented. And therefore, privacy is more important. On the other hand, Twitter has become a kind of a debate forum, where users usually engage on politics and economics debates.

The combination of the lack of privacy and the user’s tendency to use it to express its political beliefs leads to the argument of Twitter being the best, or at least a very good, network to use for understanding how society thinks about every subject.

Once Twitter had been chosen, a decision was taken which involved gathering all tweets that contained the hashtag #Bitcoin. The problem: Twitter does not make historical data available to the public nor its users.

As I researched deeper into ways of accessing historical data from Twitter, it became apparent that there were only three main ways:

1 – Paying some companies that offer historical tweets.

2- Developing a very advanced data crawling which could be against Twitter’s Terms of Service[5].

3- Access the last days tweets through Twitter’s API during a period of time and build week by week a tweets database.

The third option was chosen, and from the 20th of March until the handling of this thesis, all tweets that contained the #Bitcoin were scrapped and saved. As it has been explained in the methodology chapter, the tweets were accessed through Twitter’s API, using R’s package *rtweet* (11).

[5] Available online at: https://g.twimg.com/policies/TheTwitterUserAgreement_1.pdf

4.2 Databases

4.2.1 Kaggle Database

Range: **2010-02-22, 2018-02-19**. Therefore **2920** rows. With **24** variables. All monetary units are in United States Dollars, and therefore the nomenclature will be ignored from now on.

Variables	NAs	Definition
Date	0	Date
price	0	Daily price
total_bitcoins	0	Total Bitcoin supply
market_cap	0	Total market value of Bitcoin in dollars
trade_vol	21	Total quantity of Bitcoin traded
block_size	0	MegaBytes of data store in a block
avg_b_s	0	Average MegaBytes of data store in a block
n_orphan	0	Number of orphaned blocks
n_trans_per_block	0	Average number of transactions per block
med_confirm_time	0	Median confirmation time
hash_rate	0	Hash calculations per second
difficulty	0	Difficulty of the hash calculation
miners_revenue	0	Miners revenue
trans_fees	0	Transaction fees
cost_per_trans_pgt	0	Cost per transaction as a percentage of the price.
cost_per_trans	0	Cost per transaction
n_unique_addresses	0	Number of unique addresses
n_trans	0	Total count of daily transactions
n_trans_total	0	Accumulated number of transactions
n_trans_excl_poplr	0	Total number of transactions excluding longest chains
n_trans_excl_long	0	Number of transactions of short chains (under 100)
output_vol	0	Output volume
est_trans_vol	0	Estimated transaction volume for the following day
est_trans_vol_usd	0	Estimated transaction volume for the following day in USD

TABLE 1 - BITCOIN DATABASE VARIABLE SUMMARY

A second version of this database is scrapped for the Future Work chapter, which then will range from **2010-02-22** to **2018-06-22**.

4.2.2 Google Trends Database

Range: **2013-04-13, 2018-03-31**. Therefore **260** rows. With **3** variables.

Variables	NAs	Definition
Date	0	Date
bitcoin_good	0	Number of bitcoin_good searches in Google
bitcoin_bubble	0	Number of bitcoin_bubble searches in Google

TABLE 2 - GOOGLE TRENDS DATABASE VARIABLE SUMMARY

A second version is obtained for the Future Work chapter, which then will range from **2010-02-22** to **2018-06-22**.

4.2.3 Twitter Database

Range: **2018-05-01, 2018-06-22.** Only **3** variables and a tweet per row, which results in **over a million** rows.

Variables	NAs	Definition
Text	0	Up to 280 characters from the tweet text
Date	0	Date of the publication of the tweet
Retweets	0	Number of retweets each tweet has

TABLE 3 - TWITTER DATABASE VARIABLE SUMMARY

5 PREPROCESSING

The preprocessing of the data used in this thesis is structured around four main blocks. First, a logarithmic transformation is applied to most of the variables. Then, missing values are imputed and lags to all variables are calculated. Finally, data from the different sources is merged.

5.1 Logarithmic Transformation

As it has been explained in the Logarithm's proof at the methodology chapter, logarithmic transformation is an adequate approximation of the growth rate when this are not very high. In the following plot, the density of day to day growth (as a percentage of the previous period price) is shown. As it can be seen, the variability is not very high, and therefore the usage of logarithms is suitable in the Bitcoin price.

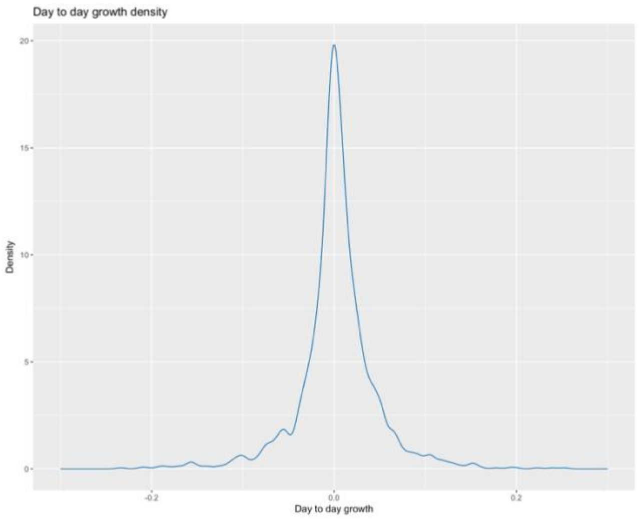


FIGURE 12 - DAY TO DAY BITCOIN PRICE GROWTH DENSITY – FROM 2013 TO 2018

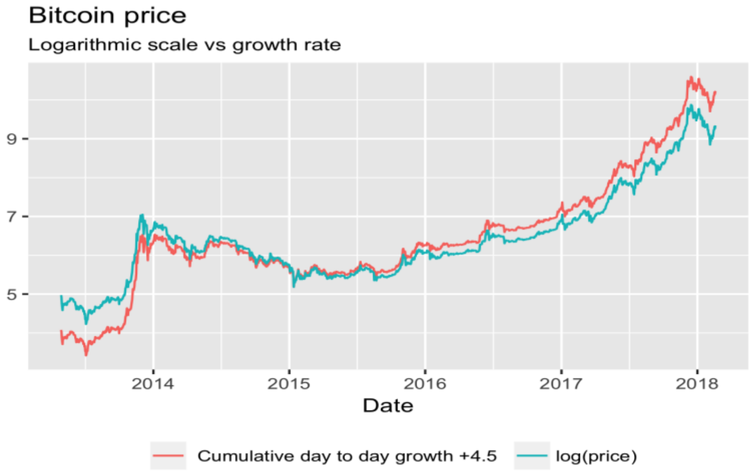


FIGURE 13 - CUMULATIVE GROWHT RATE AND LOGARITHMICALLY TRANSFORMED BITCOIN PRICE

Figure 13 shows that the logarithm of the price has an almost identical behaviour than the cumulative day to day growth, that has a constant added in order to make the two series overlap.

Once it has been seen that applying a logarithmic transformation to the price is suitable conceptually, it is interesting to highlight that it has a clear non-linear behaviour (as it will be seen in the exploratory chapter). Therefore, this transformation is not only suitable but adequate, as many of the techniques used are based on linearity assumptions.

Finally, before proceeding further into the preprocessing, it is necessary to explain that this same transformation will be applied to almost all variables, as they all exhibit a very similar behaviour. From this point on, all variables will be referred with their regular name, even though they will have been logarithmically transformed.

5.2 Missing Data Imputation

Luckily, the data had only missing values in one variable, which is *traded volume,* and of the more than 1500 rows, only 15 have missing values. As it has been explained in the methodology chapter, a PMM imputation is performed.

As it can be seen in the following plot, the density of the imputed values is fairly similar to the predicted one, and as a consequence the imputation can be considered to be correct.

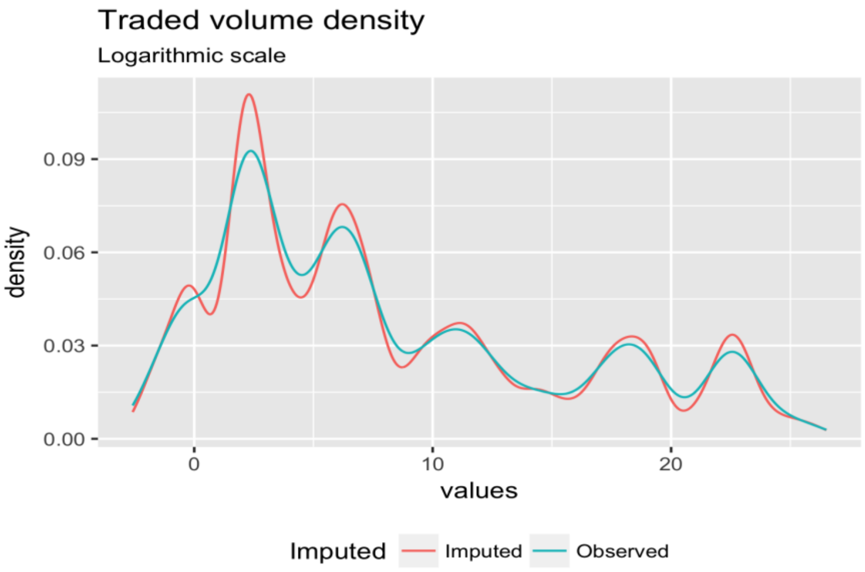


FIGURE 14 - IMPUTED AND OBSERVED LOGARITHMICALLY TRANSFORMED TRADED VOLUME DENSITY

5.3 Variable creation & lags

A new variable is defined: day to day growth which will represent the day to day price growth and will be calculated as the following formula shows. The way the variable is defined force the first observation to be removed.

$$dtdg = \frac{Marketprice_{t+1} - Marketprice_t}{Marketprice_t}$$

Another variable is created, g_diff, which shows the difference of bubble and good searches about Bitcoin in Google Trends.

$$g_{diff} = Goodsearches - Bubble\ searches$$

Some more variables could be defined following the criteria defined at the Classical Trading Theory chapter (2.1), but as it is seen there, in several examples applied to Bitcoin the patterns do not work as expected and therefore a purely causal approach will be taken.

Some variables are lagged in order to obtain the necessary database for the Arimax and ANN. The number of lags have been decided conceptually from the Arima's autoregressive and moving average analysis, included in the Appendix. It concluded that 1,2 and 7 day lags of the price were adequate. This thesis makes the assumption that it is possible that the exogenous variables affect the price with the same lag structure.

Therefore, 1,2 and 7 lags are created for all variables, resulting in a 78 variables dataset. Then, a stepwise mechanism is implemented as explained in the methodology chapter to decide which variables are included, and with how many lags. The following are the results:

Variables with one lag: difficulty, price, traded volume, block size.

Variables with two lags: difficulty, day to day growth (dtdg), price, block size, Bitcoin bubble searches.

Variables with seven lags: number of transactions, transaction fees.

5.4 Data Merging

As it has been previously said, until the Future Work chapter the tweets' sentiment will not be merged into the main dataset as there is not enough data. Therefore, the main dataset (which contains daily data) is only merged with the Google Trends data (which contains weekly data). Taking into account the multigranularity theoretical framework tackled in the Methodology chapter, it is decided that as the price is the variable being forecasted, it should be kept in the same granularity level as it was collected. Therefore, the trends data, which is weekly, is repeated so every day of the week has the same value.

6 EXPLORATORY ANALYSIS

The exploratory analysis of the current thesis has been kept minimal, in order to keep the focus on the forecasting. At the end of the appendix, all variables can be found plotted against the Date.

First, some key plots are shown in order to provide some context to the reader. Then, a Clustering and Principal Components Analysis are shown together, in order to make some specific arguments. Finally, a profiling of the Clusters is made to obtain several conclusions.

6.1 Relevant plots

Let us begin with a plot of the Bitcoin price through time, which will show the time-series being forecasted in this thesis, before and after it is transformed to logarithms.

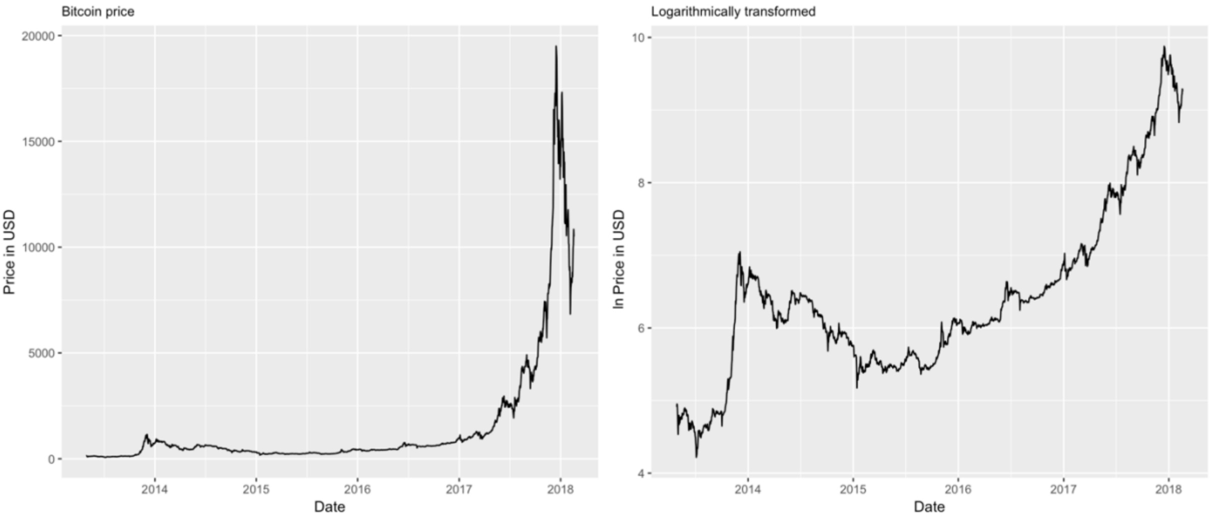


FIGURE 15 - BITCOIN PRICE TIME-SERIES BEFORE AND AFTER LOGARITHMIC TRANSFORMATION

Using the *corrplot* package (28), a plot that shows all correlations between each pair of variables is calculated. As it was expected, most variables have a very high positive correlation. The only exceptions are: the cost per transfer, which has decreased over time. The day to day growth of the logarithm of the price, with appears to be uncorrelated to all other variables. Finally, the information taken from Google does not seem to be correlated.

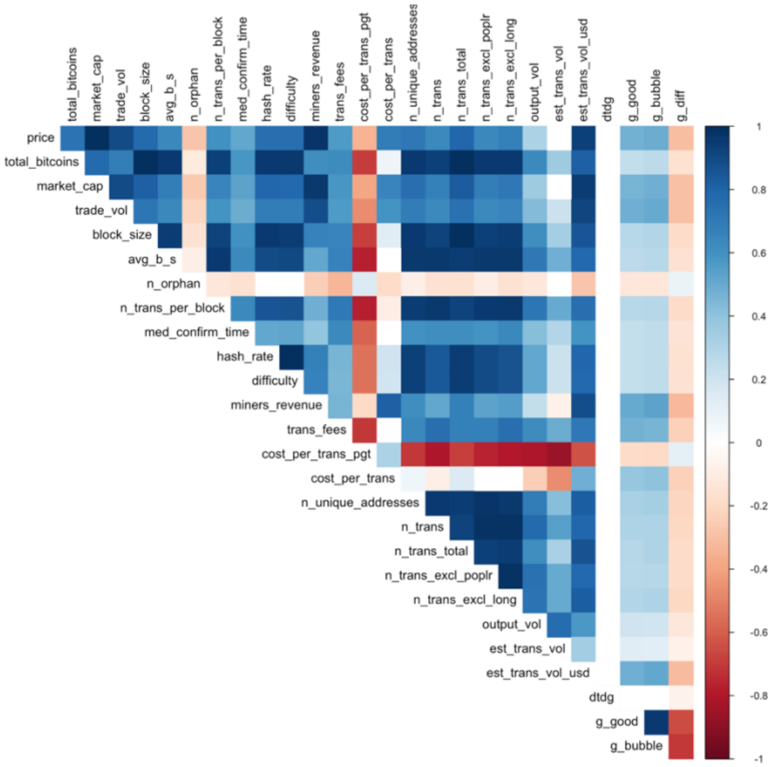


FIGURE 16 - CORRELATION BETWEEN VARIABLES IN THE DATASET

The following plot displays the weekly searches of “Bitcoin good” and “Bitcoin bubble” on Google. As it can be seen, Bitcoin good is usually more searched, but on determinate periods the Bitcoin bubble search rises rapidly, being the most searched. Overall, it can be said that the two peaks are on the dates of higher price growth.

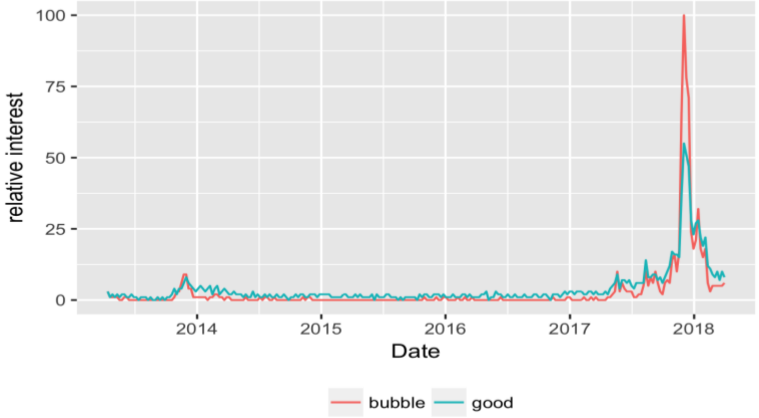


FIGURE 17 - BITCOIN BUBBLE AND BITCOIN GOOD SEARCHES IN GOOGLE TRENDS OVER TIME

In order to dive deeper into the dataset, bivariate plots are not enough. Therefore, the exploratory analysis follows with a Principal Component Analysis and a Clustering technique.

6.2 Principal Components Analysis & Clustering

The first plot to consider in order to analyse the PCA, is the cumulative explained variance by each of the principal components. As it can be seen, the first factorial plane gathers over 75% of the total hyperspace variance. As it is high enough, only the first factorial plane will be analysed in this section.

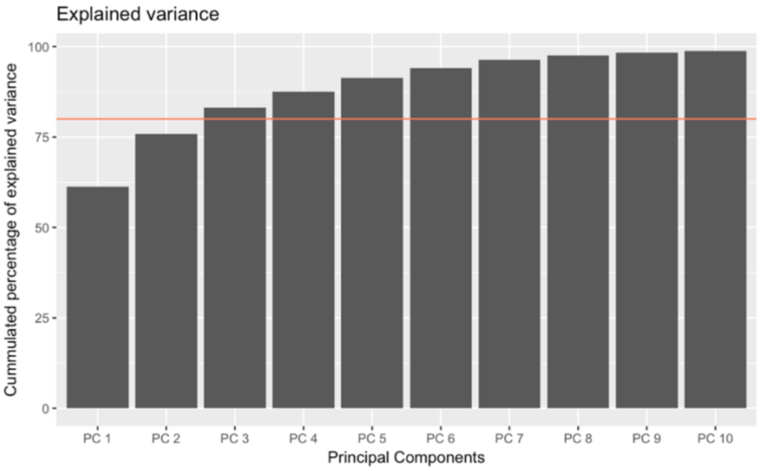


FIGURE 18 - CUMULATED EXPLAINED VARIANCE THROUGH THE MAIN PRINCIPAL COMPONENTS

The following plot shows variable representation over the first factorial plane. As it can be seen there is a large set of variables, all correlated between them and uncorrelated to the price. The correlated set of variables are all volume-related in the sense they are counts, for instance: number of transactions per block, block size, n of unique addresses, n of Bitcoins. Of course, as the system success has increased so have all of these variables, which are apparently not correlated to price.

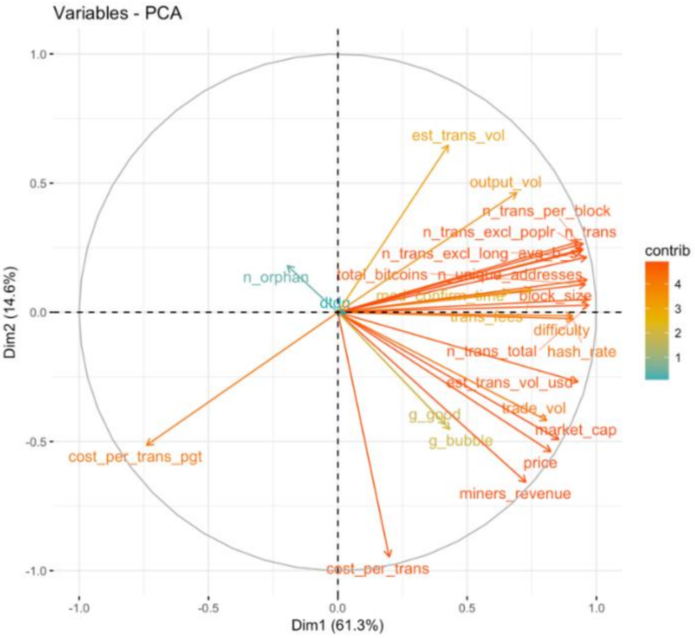


FIGURE 19 - VARIABLES REPRESENTED ON THE FIRST FACTORIAL PLANE

Once the first factorial plane has been analysed, the exploration proceeds into the Clustering.

In the following dendrogram it can be seen that the most obvious cut would have been selecting 2 groups. The two-group cluster representation has been added to the appendix. As the goal was to obtain very detail and constrained groups, 5 of them were selected.

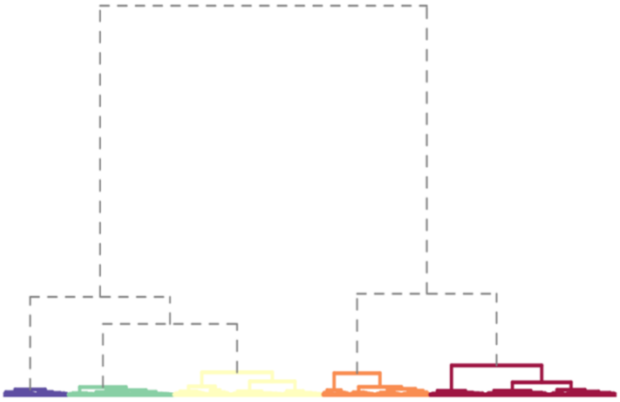


FIGURE 20 - 5 GROUP DENDOGRAM

From the following figure, very interesting ideas can be extracted. First of all, there is a clear change of behaviour of the variables through time. In fact, it seems that every there is a change approximately every 300 to 450 days. This length will be used to calculate the length of the rolling window, which will be set to 365 days.

On the right, the clusters have been painted onto the first factorial plane of the PCA. The fact that there is almost no overlapping between clusters seems to show that the factorial plane is a good representation of the hyperspace.

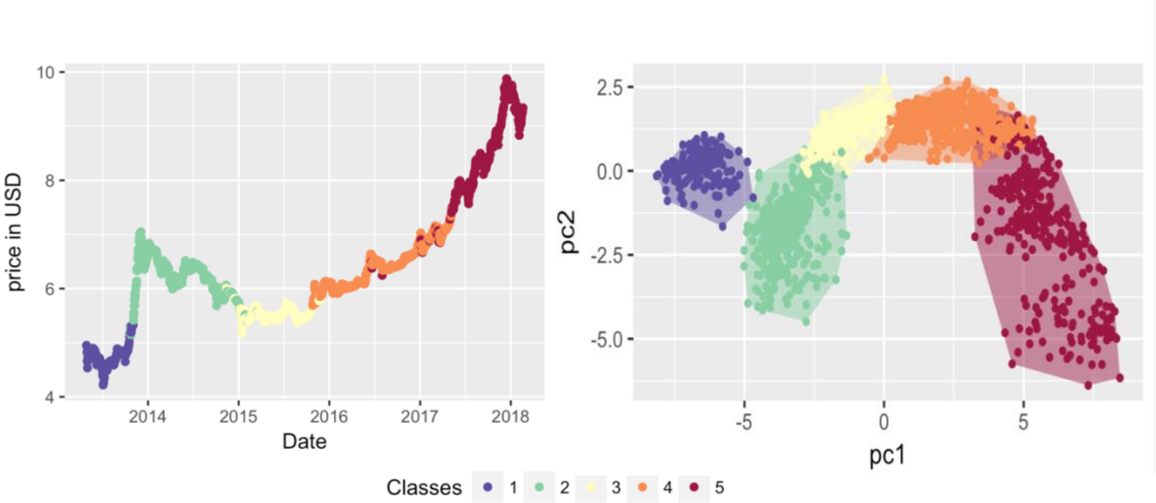


FIGURE 21 - CLUSTERS REPRESENTED OVER THE BITCOIN PRICE AND OVER THE FIRST FACTORIAL PLANE FROM HE PCA

Cluster profiling

The following figure shows, in a very condensed way, the whole profiling of all clusters. Each variable is escalated and plotted through polar coordinates. While the escalation does not allow to identify cluster's mean on each variable, it allows to identify its hierarchy.

It can be clearly seen that on most variables the hierarchy is always the same. The fifth cluster, which contain the most recent observations, have in most of the variables the highest value. The less recent data the cluster includes, the lower the value the cluster gets in most variables.

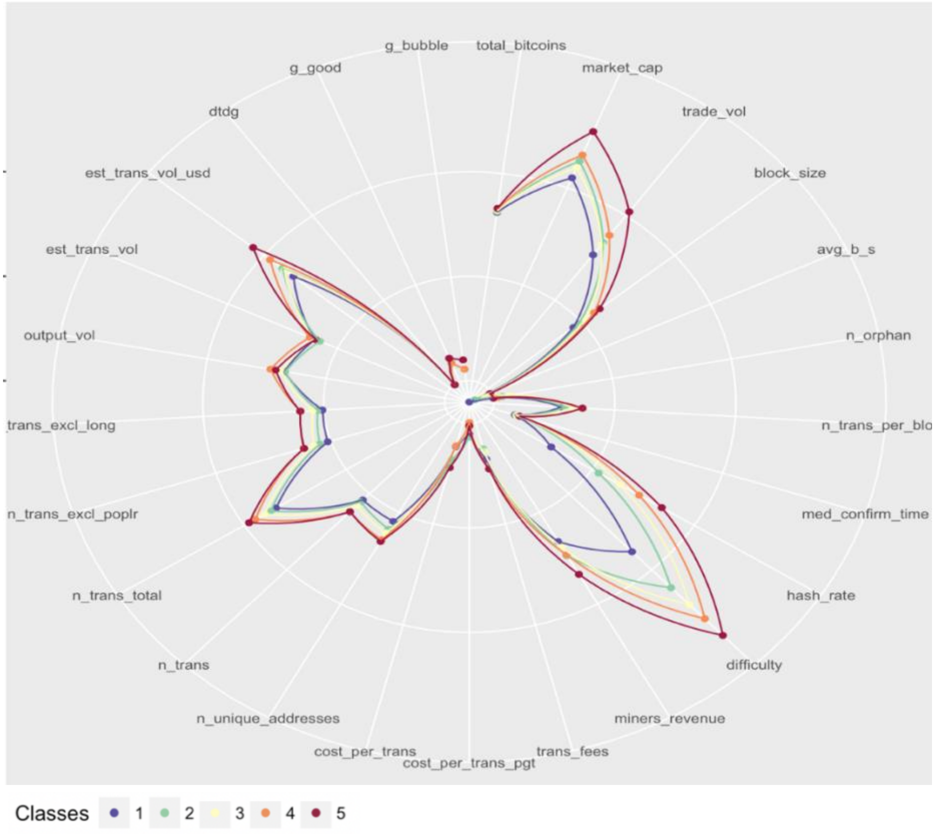


FIGURE 22 - CLUSTER PROFILING

What this mainly means is that while the price has been changing over time repeatedly reaching new values, the same has happened to other variables. The result, is that even though there is no time variable included in the cluster, the resulting groups are time ordered.

Once all these conclusions have been reached, and a deeper understanding of the database is achieved this thesis follows into the main chapter, Forecasting.

7 FORECASTING

The forecasted variable is the Bitcoin's price. As it has been explained in the methodology chapter, three main methods have been used to make the predictions. Arima, Arimax and Artificial Neural Networks (ANN).

As the available data includes a very wide range of time there will not be forecasts of the whole series due to computational time constraints. In the following plot, it can be seen that the selected range goes from 2017 to mid-February. Also, besides the price, the monthly and weekly average is shown, as it will be used in some of the forecasts to remove the "noise".

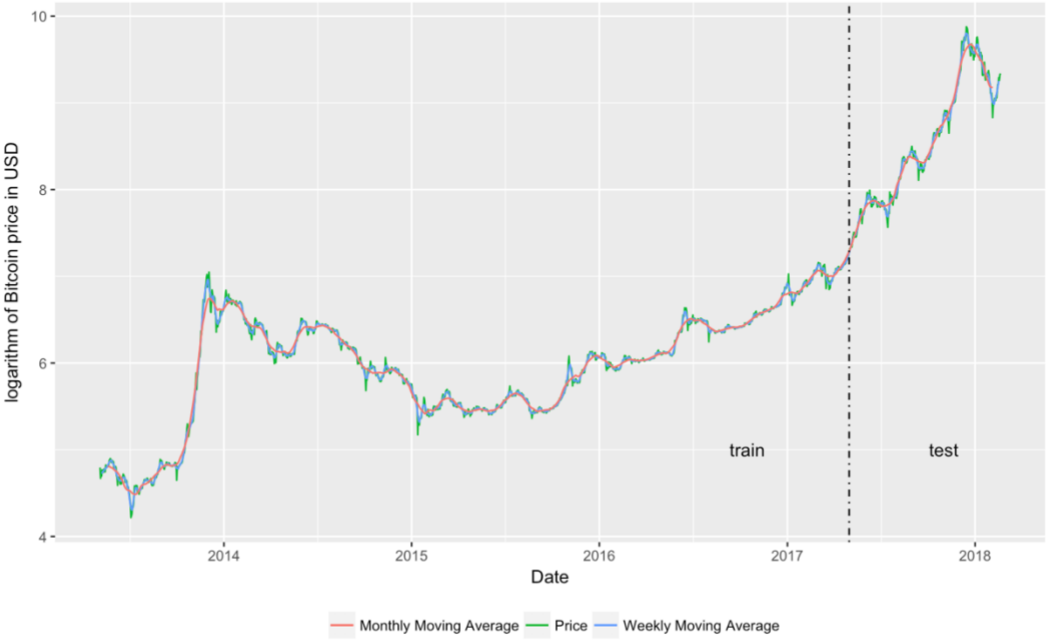


FIGURE 23 - TRAIN AND TEST DIVISION OVER THE FORECASTED VARIABLE

Before proceeding to each of the three methods, it has to be stated that each forecasting methods will make forecasts from 251 different days, from the 12th of June of 2017 to the 19th of February of 2018. From each day, a 1 to 10 scope forecasts will be performed. Resulting in a total 2410 forecasts from each method. For each of the scopes, several visualizations will be shown, as well as a summary table at the end of the section comparing both numerically and visually all models.

7.1 ARIMA

In the following sub-sections it will be assessed how has the Arima forecast performed, mainly through a visual approach in both, Incremental and Rolling Windows.

7.1.1 Structure Selection

In order to perform the selection of the (p,D,q) Arima and Arimax structure, a double approach has been taken. A theoretical one, based on residuals behaviour analysis, and an automated one, based on AIC optimization through the *auto.arima* function, from the *forecast* package (22).

In order to keep this thesis clutter free, the detailed explanation of the selection of the (p,D,q) Arima and Arimax structure has been included in the Appendix.

The conclusion of the procedure is that the best possible structure is (2,1,7). This structure's meaning can be simplified by stating that the price, with a first-order differentiation, at determinate period, depends on the price on the two last periods and the errors from the last seven periods.

As it has been explained on the Methodology chapter in the Arima sub-section (3.4.3), the structure is calculated once with the train dataset. It will be kept through all the forecasting Windows, only re-estimating the parameters.

7.1.2 Incremental Window

The following plot displays all forecasts through time. As it can be seen, it is obvious that the infinite memory of the parameter calculation causes the forecasts to adapt slowly to a slope change. As a consequence, when the series is relatively stable the forecasts are fairly good. Finally, it is clear that longer scope forecasts are clearly worse, which was expectable.

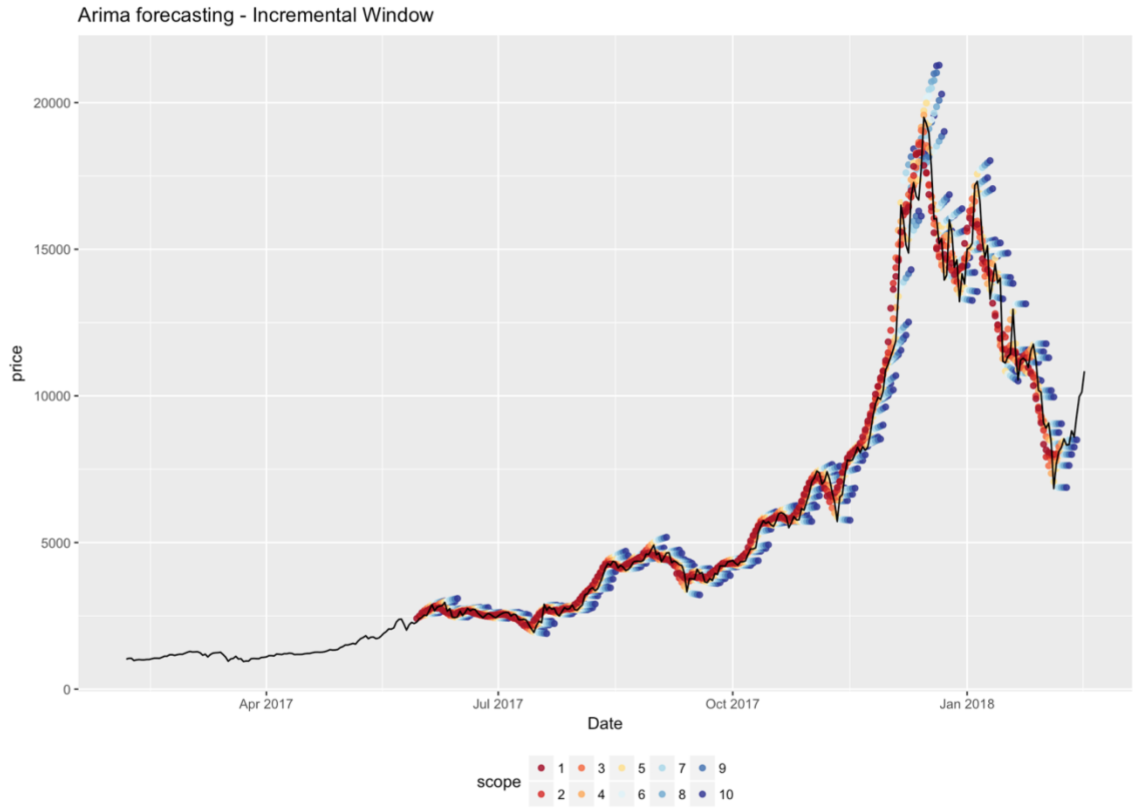


FIGURE 24 – ARIMA FORECASTS THROUGH TIME BY SCOPE - INCREMENTAL WINDOW

Graphical error assessment

On the following plot the errors through time are displayed. It can be seen that most of the time errors oscillate around 0, which is a baseline requirement. In contrast, a requirement that is not met is that variability is not constant, but increases over time both on the low and high scopes.

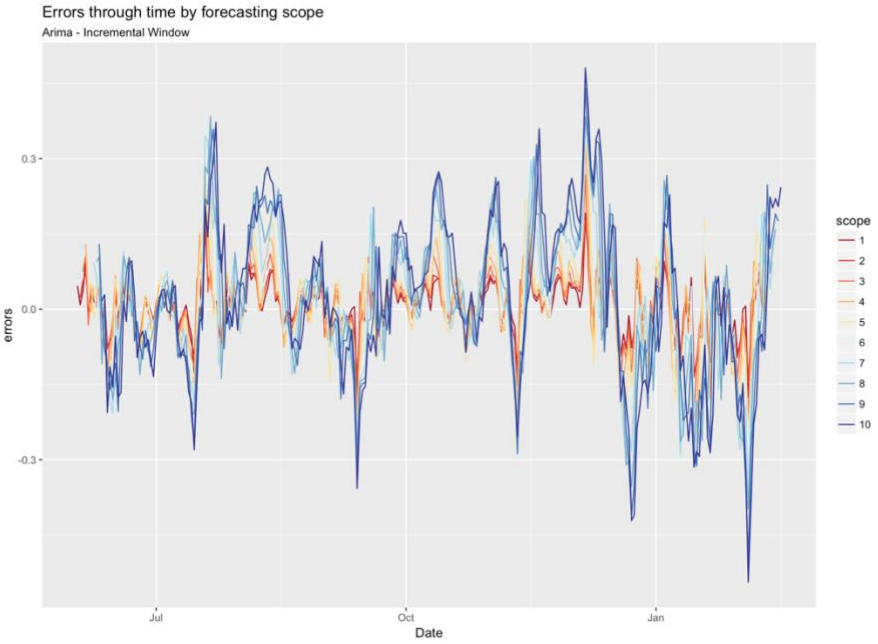


FIGURE 25 – ARIMA FORECASTING ERRORS THROUGH TIME BY SCOPE - INCREMENTAL WINDOW

As it was expected from the errors through time plot, error distributions by scope are centered on 0. The higher the scope, the higher the variability.

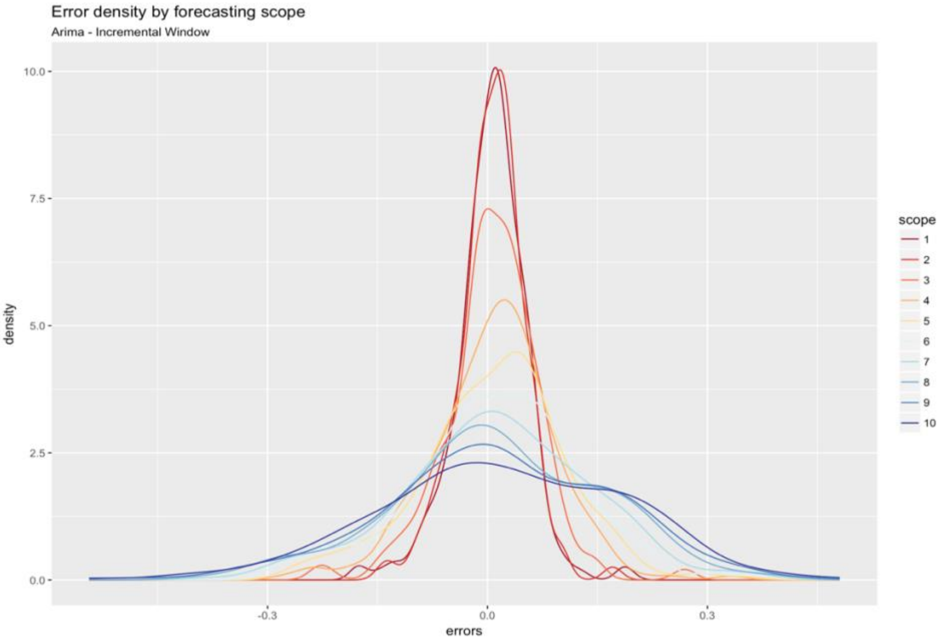


FIGURE 26 - ARIMA ERROR DENSITY BY FORECASTING SCOPE - INCREMENTAL WINDOW

Finally, on the Incremental Window Arima Forecasting sub-section, the parameter stability is checked by displaying the coefficient of the parameters on each model estimation. As it can be seen, the model presents a surprising variability on the coefficients, which should be looked at carefully if the model were to be implemented and used in stock trading. It is surprising due to the nature of Incremental Windows, which have infinite memory, and therefore coefficients are expected to be almost invariant through time.

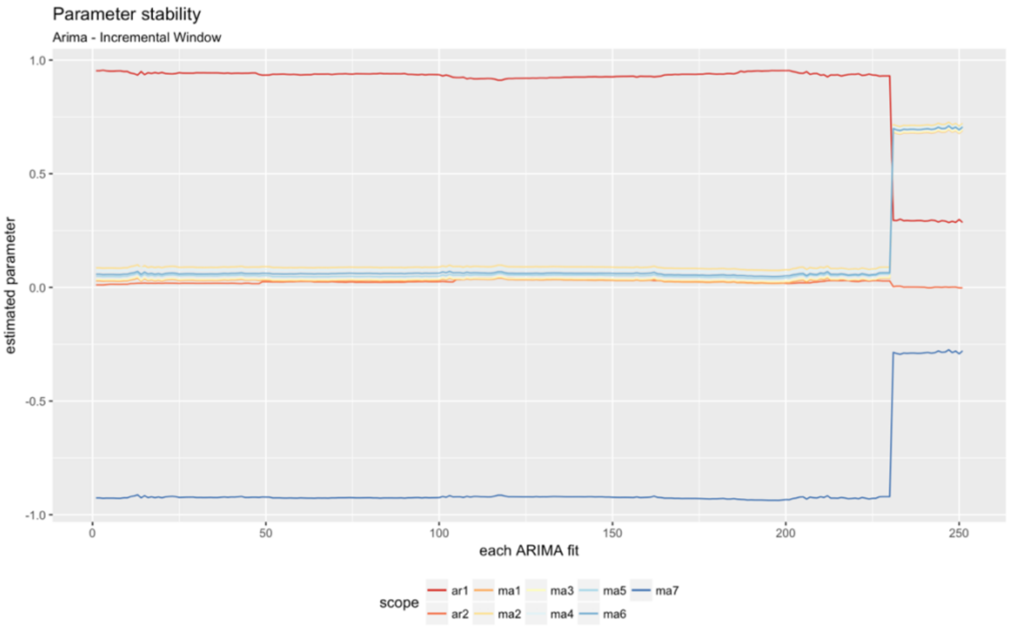


FIGURE 27 - ARIMA PARAMETER STABILITY BY MODEL - INCREMENTAL WINDOW

7.1.3 Rolling Window

The following plot displays all forecasts through time. As it can be seen, the fact of using a Rolling Window does not seem to increase the speed of adapting to a slope change. A potential cause could be the selection of a wider than optimal Window.

As it happened with the Incremental Window, it is clear that longer scope forecasts are clearly worse, which was expectable.

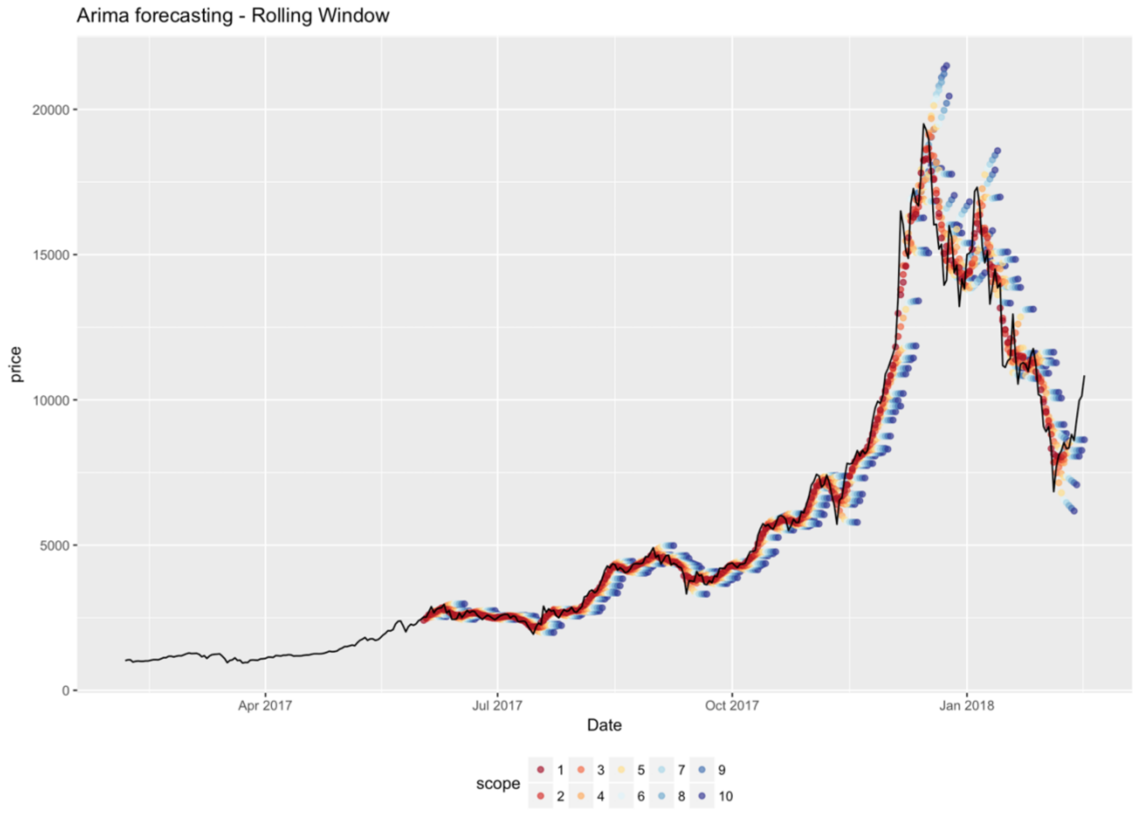


FIGURE 28 - ARIMA FORECAST THROUGH TIME BY SCOPE - ROLLING WINDOW

Graphical error assessment

On the following plot the errors through time are displayed. It can be seen that most of the time errors oscillate around 0, which is a baseline requirement. Again, as it happened in the Incremental window, a requirement that is not met is that variability is not constant, but increases over time both on the low and high scopes.

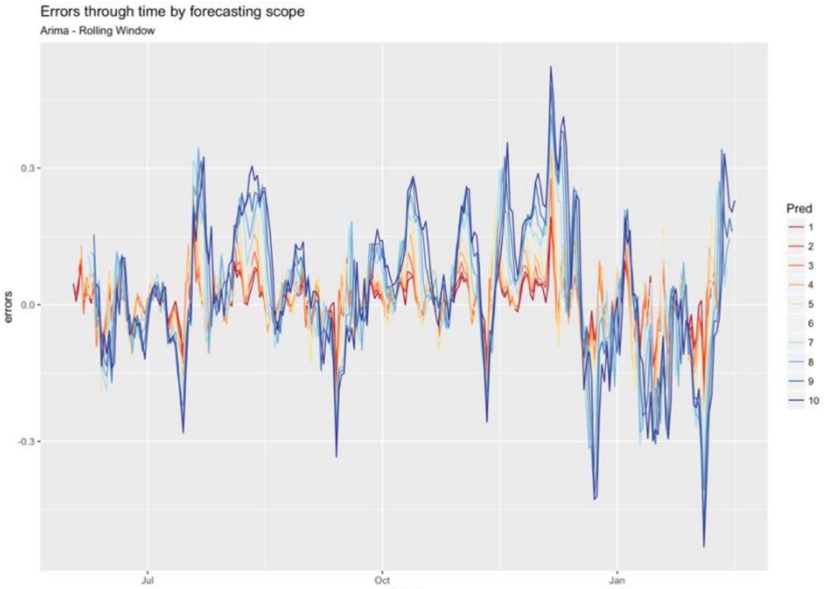


FIGURE 29 - ARIMA ERRORS THROUGH TIME BY SCOPE - ROLLING WINDOW

As it was expected from the errors through time plot, error distributions by scope are centered on 0. The higher the scope, the higher the variability.

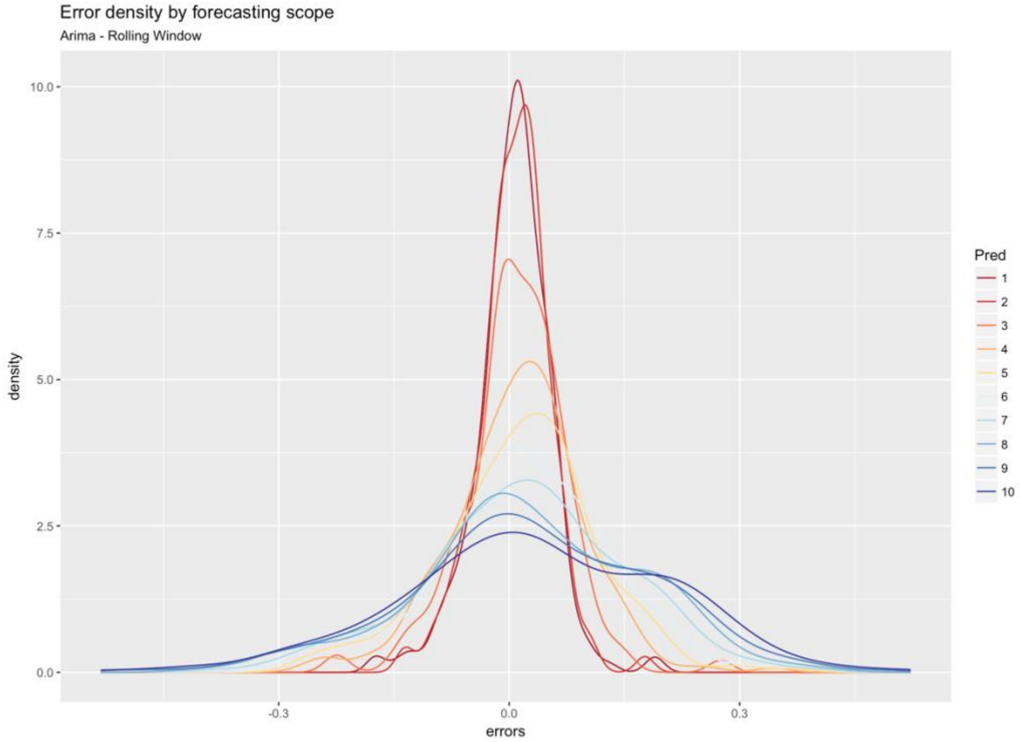


FIGURE 30 - ARIMA ERROR DENSITY BY SCOPE - ROLLING WINDOW

Finally on the Rolling Window Arima Forecasting sub-section, the parameter stability is checked by displaying the coefficient of the parameters on each model estimation. As it can be seen, the behavior is different from the one on the Incremental Window. It is clear then even though there is still variability, the coefficients appear to less stable as it was expected.

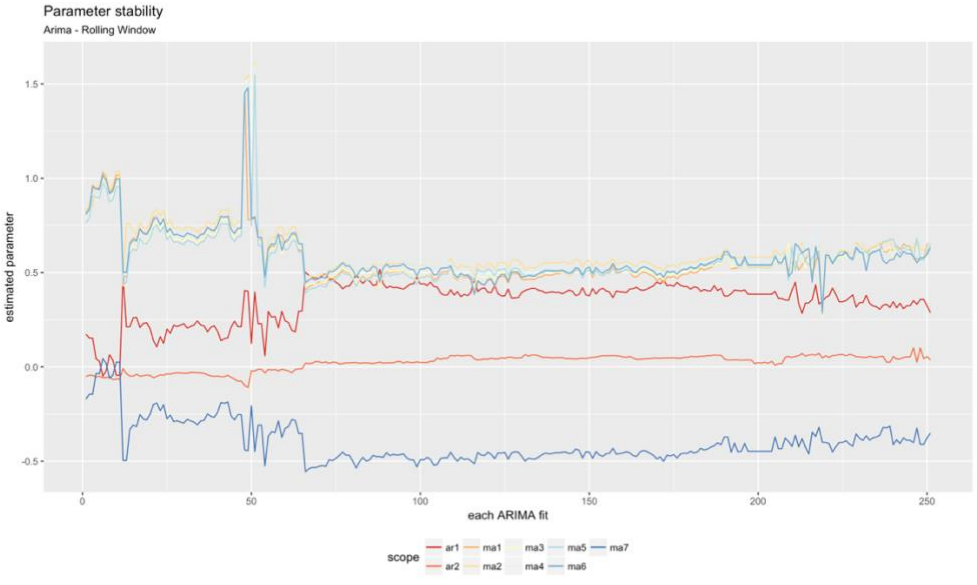


FIGURE 31 - ARIMA PARAMETER STABILITY BY MODEL - ROLLING WINDOW

7.2 ARIMAX

In the following sub-sections it will be assessed how has the Arimax forecast performed, mainly through a visual approach in both Incremental and Rolling Windows.

Before proceeding, some information about the model has to be given. It uses the same (2,1,7) structure calculated in the Arima, and adds several exogenous variables.

The included variables are showed on the Preprocessing chapter in the Variable Creation & lags section (5.3).

7.2.1 Incremental Window

The following plot shows the forecasts through time of the Arimax Model under an Incremental Window. In contrast to the Arima models, it can be seen that higher scopes don't go as far away from the real series. Also, lower scopes seem to follow adequately the series.

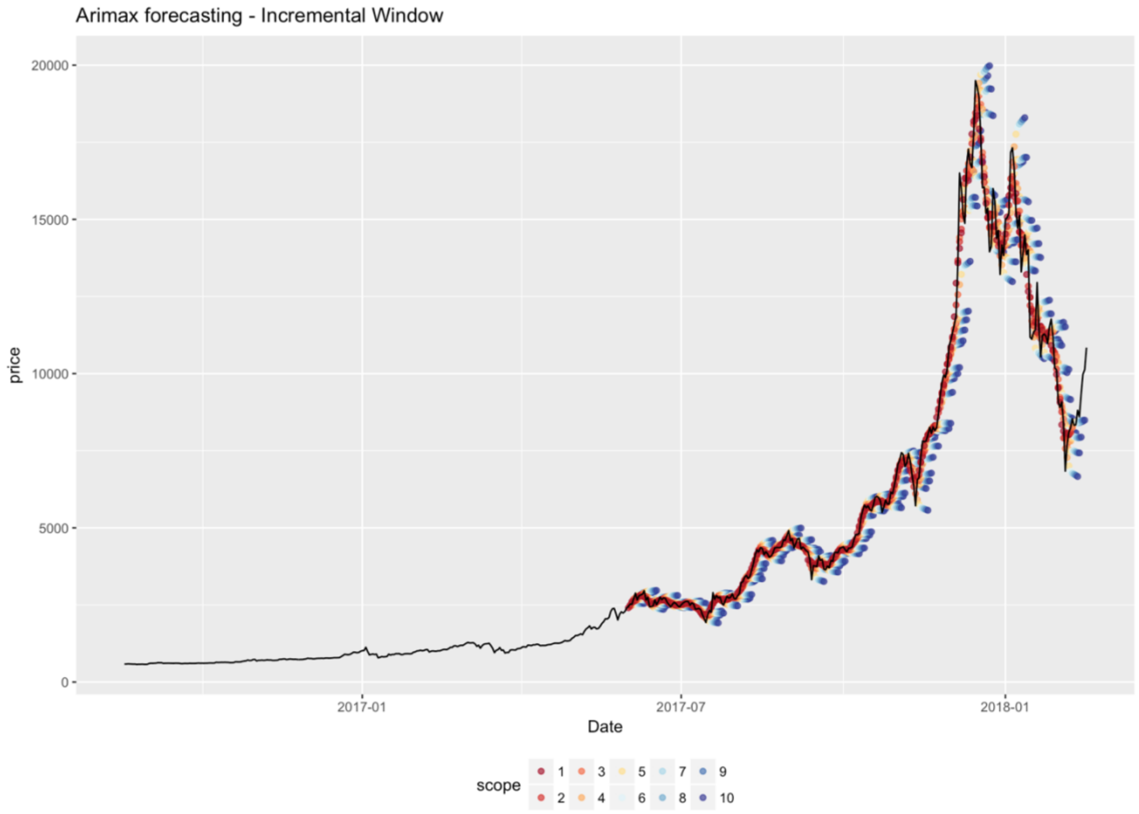


FIGURE 32 - ARIMAX FORECASTING THROUGH TIME BY SCOPE - INCREMENTAL WINDOW

Graphical error assessment

As it happened in the Arima models, errors through time seem to have an increasing variance but oscillate around 0, which was to be expected in an adequate model.

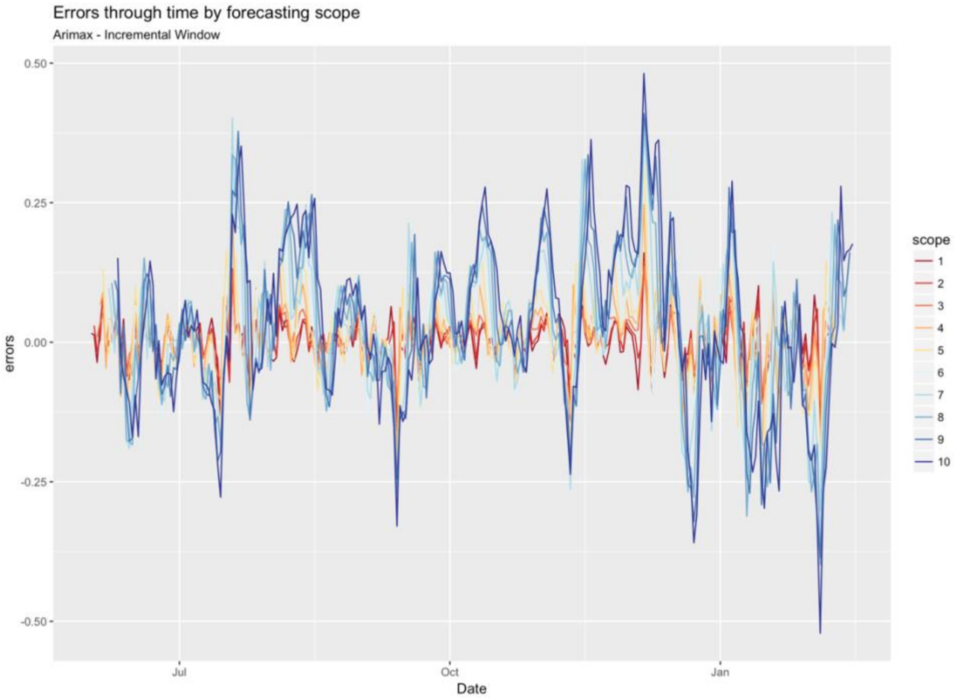


FIGURE 33 - ARIMAX ERRORS THROUGH TIME BY SCOPE - INCREMENTAL WINDOW

Again, error densities are centered on 0. Surprisingly, the 3 and 2 days scope seem to have a better forecast than the 1-day forecast. This can be seen as the density distribution is higher.

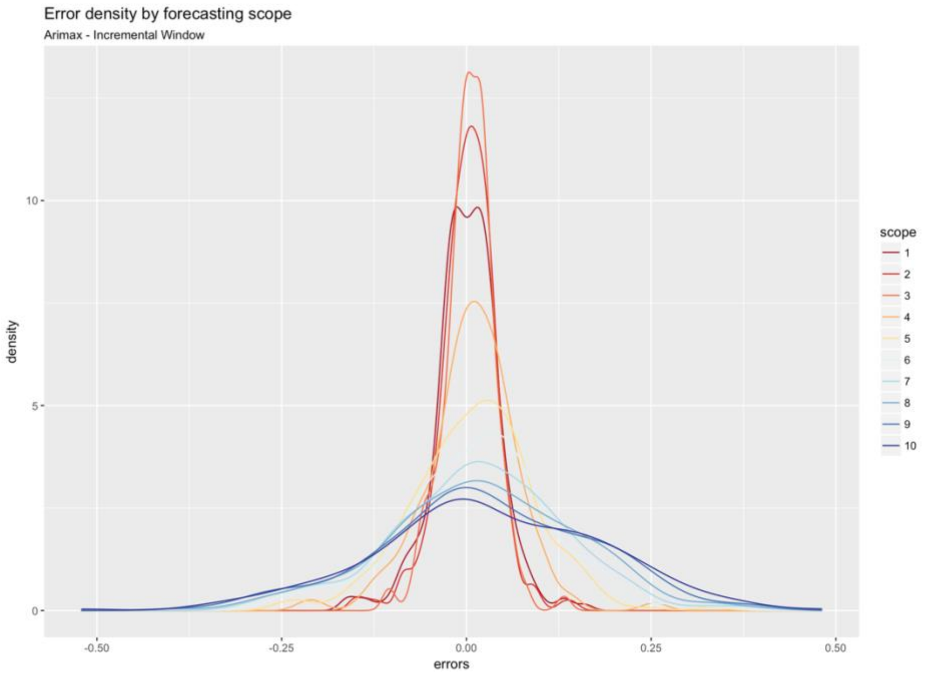


FIGURE 34 - ARIMAX ERROR DENSITY BY SCOPE - INCREMENTAL WINDOW

Finally, it is very interesting to see how the stability of the AR and MA parameters is achieved, and in exchange, the block size (lagged once and twice) parameters are volatile. A potential explanation could be that the variability of the parameters is absorbed by the exogenous variables, allowing the parameters of the ARMA part to be stable.

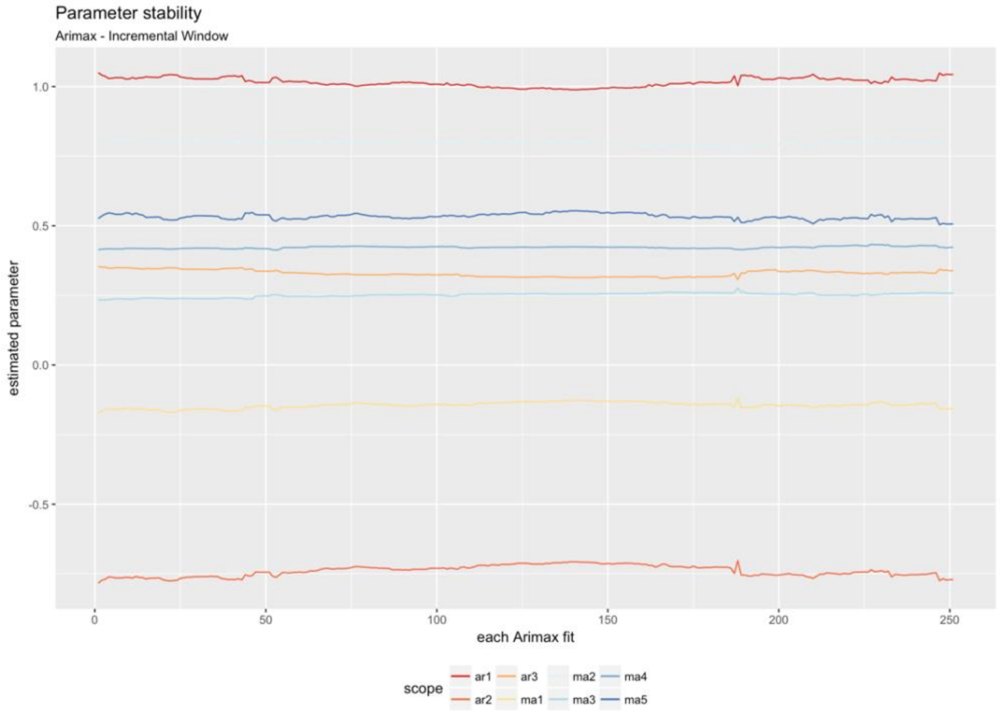


FIGURE 35 - ARIMAX ARMA PARAMETER STABILITY - INCREMENTAL WINDOW

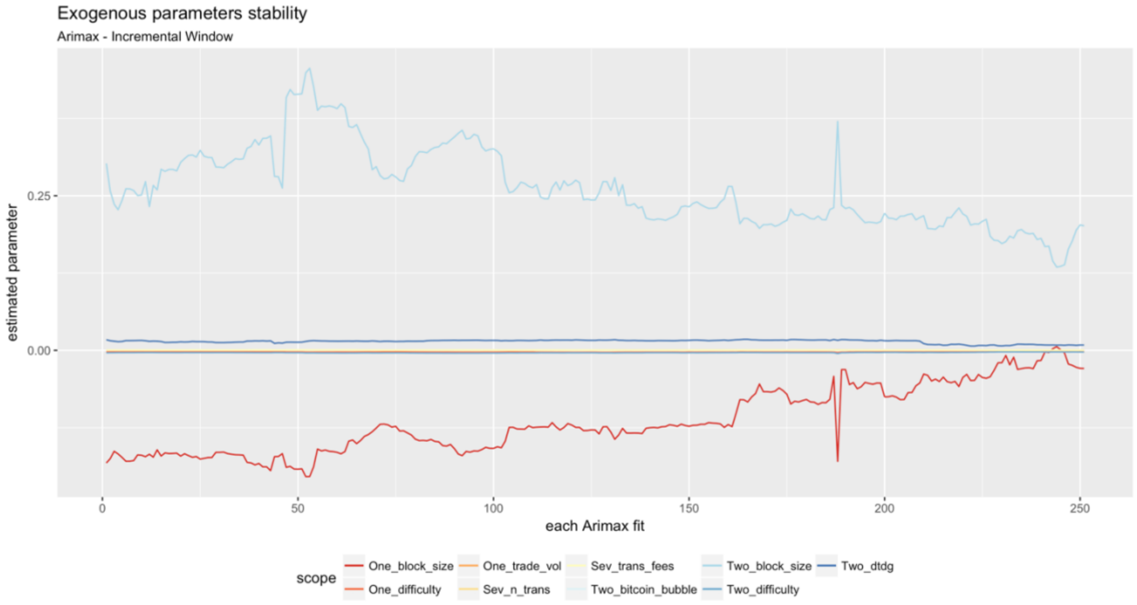


FIGURE 36 - ARIMAX EXOGENOUS PARAMETERS STABILITY - INCREMENTAL WINDOW

7.2.2 Rolling Window

The following plot shows the forecasts through time of the Arimax Model under a Rolling Window. It presents an unexpected behaviour on the higher scopes.

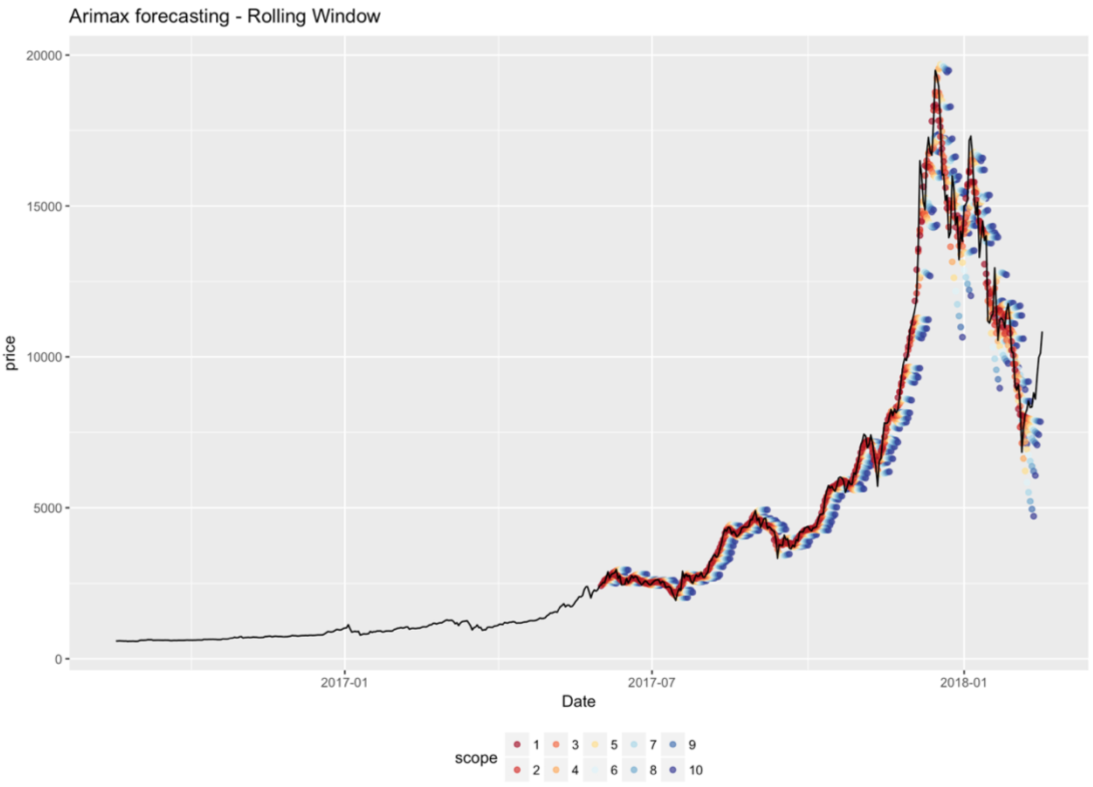


FIGURE 37 - ARIMAX FORECASTS THROUGH TIME BY SCOPE - ROLLING WINDOW

Graphical error assessment

The unexpected behaviour that was previously observed is seen again in the errors through time plot, as the variability increases sharply on the last forecasts.

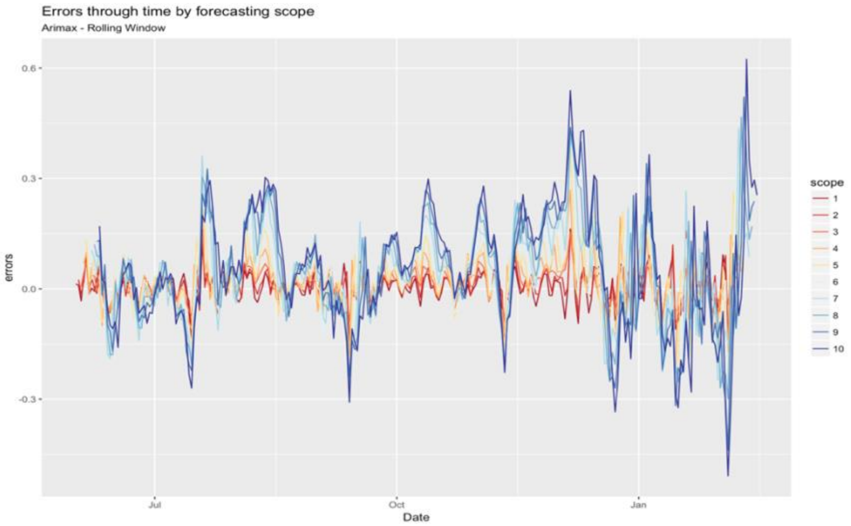


FIGURE 38 - ERRORS THROUGHT TIME BY SCOPE – ROLLING WINDOW

In the error density by prediction scope plot, it can be seen that 1 to 3 day forecasts seem to be correct, while all other scopes get higher variability errors. Again, the 1 day scope is not the best one, which is remarkable.

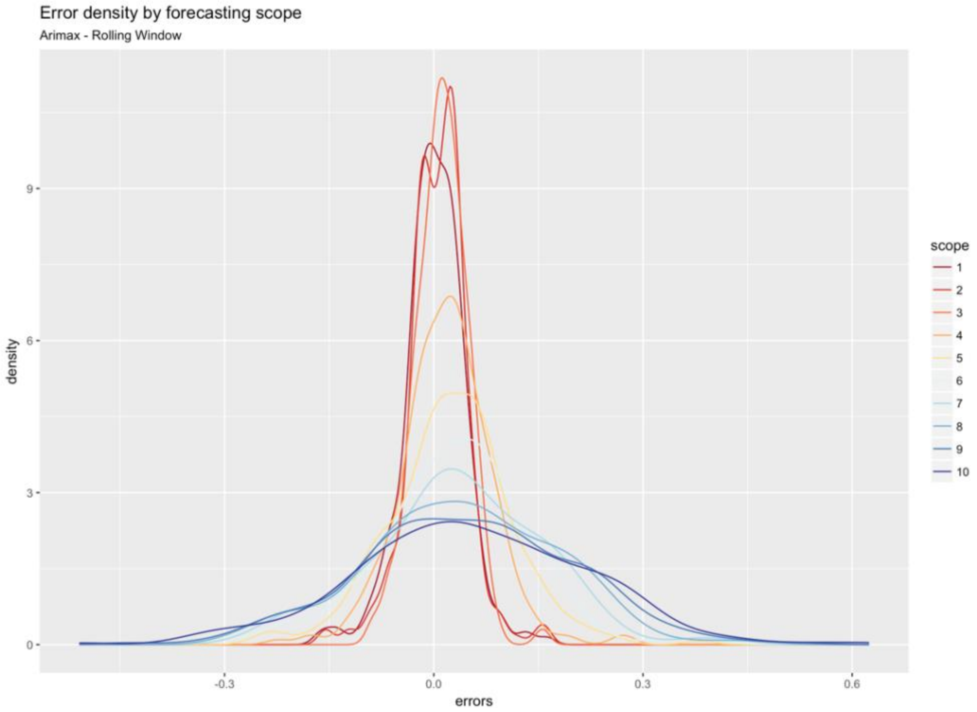


FIGURE 39 - ARIMAX ERROR DENSITY BY SCOPE - ROLLING WINDOW

In contrast of the Incremental Window, here the Arimax parameters seem to keep the stability it has in the IW implementation but appear to be slightly more variable.

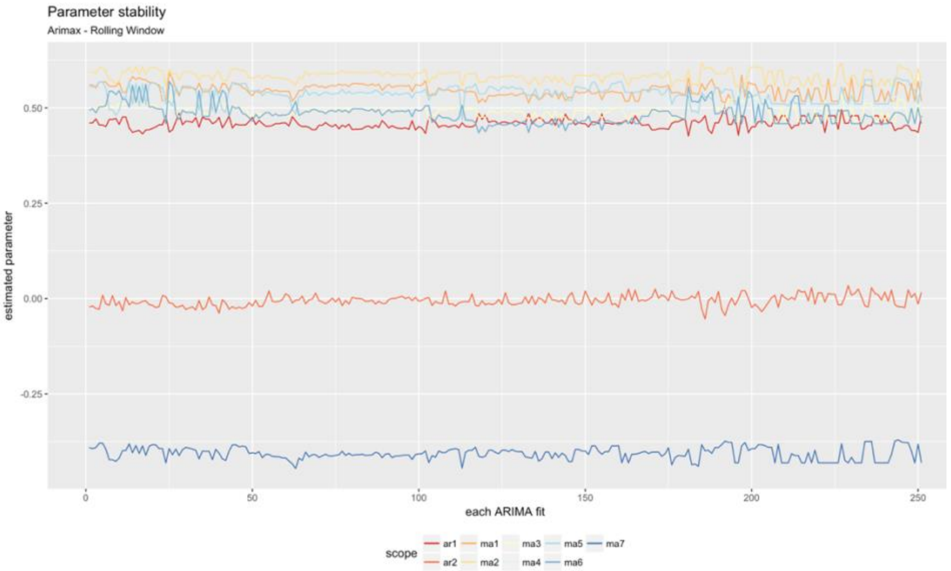


FIGURE 40 - ARIMAX PARAMETER STABILITY - ROLLING WINDOW

As it can be seen in the following plot the one and two-day lags of the variable Block size seem to gather all the coefficient value, while all of the other exogenous variables coefficients are around 0.

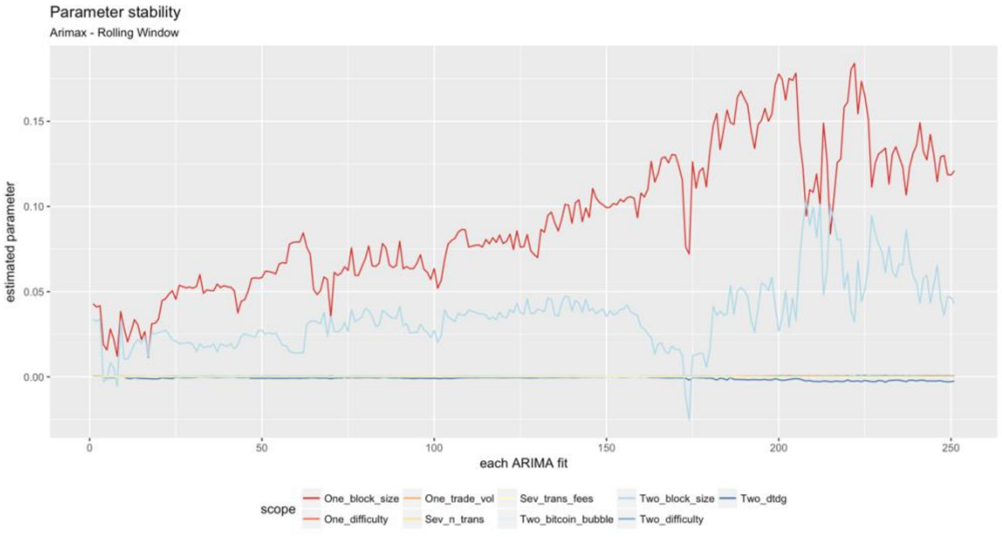


FIGURE 41 - ARIMAX PARAMETER STABILITY - ROLLING WINDOW

7.3 Artificial Neural Network

As it is done with the Arima models, the neural network is trained once with historical data to set the optimal hyper-parameters. Then, in each Rolling or Incremental Window only the parameters will be re-estimated.

This training with historical data is done with a 10 fold cross-validation, 1 to 10 potential units in the hidden layer and a decay value ranging from 0.1 to 0.5. The selected performance metric is the root mean squared error. The final selected values are 8 units in the hidden layer and a decay value of 0.3.

As it has been explained in the methodology chapter (3.4.5) the ANN implementation will be only under an Incremental Window approach.

In the forecasting through time plot, it can be seen that the overall appearance is significatively different than the Arima's. It is clearly seen that there is no time-series behaviour in the forecasts. This is probably a consequence of using a causal model which originary goal is not to make future predictions with a tight structure as in the Arima framework.

Also, there are some obvious outliers, that will not be taken into account when calculating error densities and variance as if this algorithm where to be implemented it would have had some minimum and maximum tolerance.

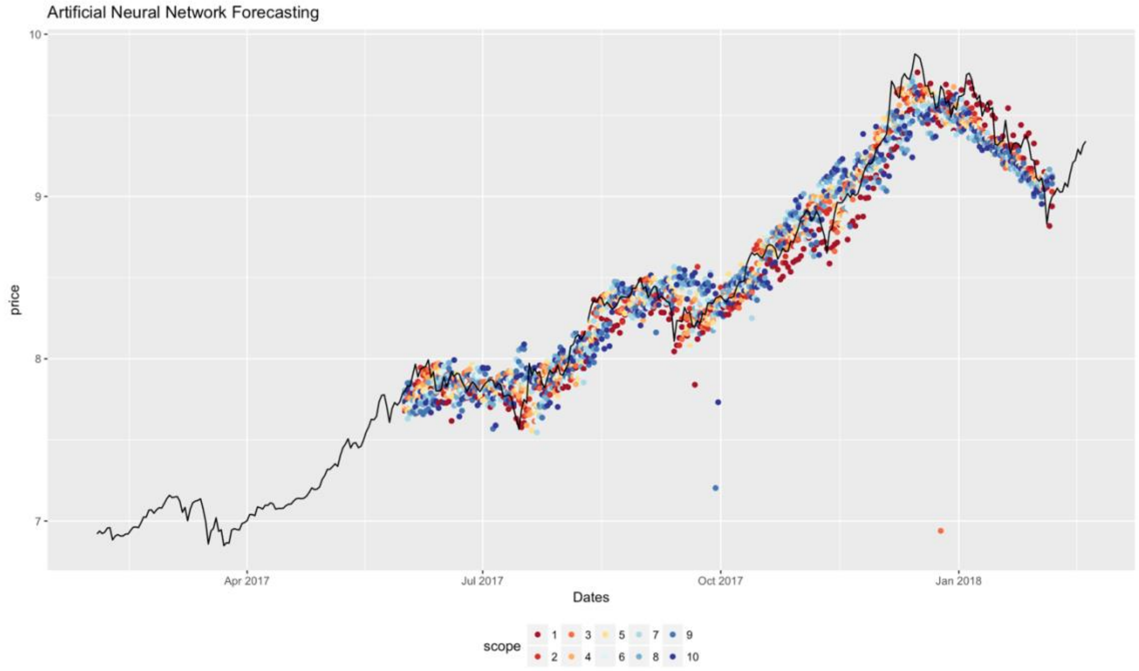


FIGURE 42 - ANN FORECASTS BY SCOPE - INCREMENTAL WINDOW

As it could be expected from the forecasting plot, it is seen that errors from the ANN have a wide variance and are not necessarily centred in 0. In fact, each of the scope errors has a different centre.

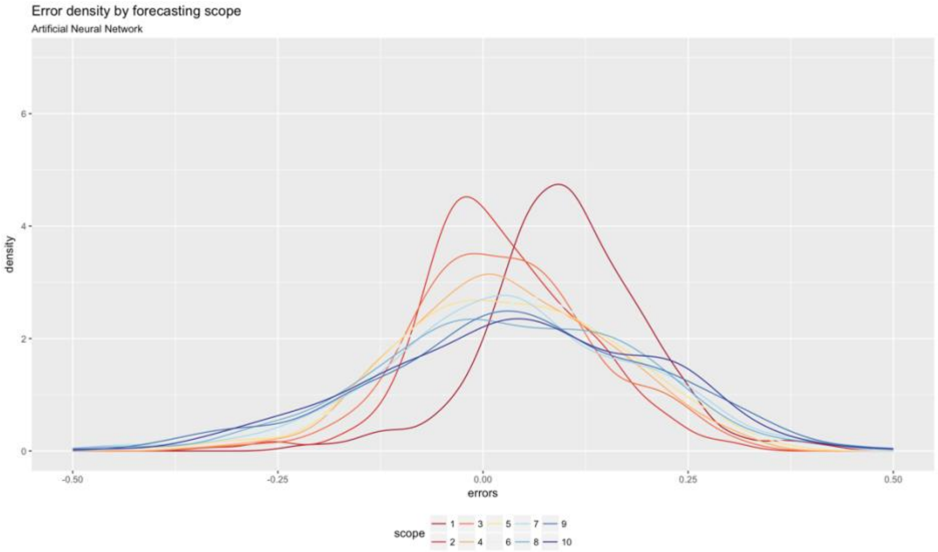
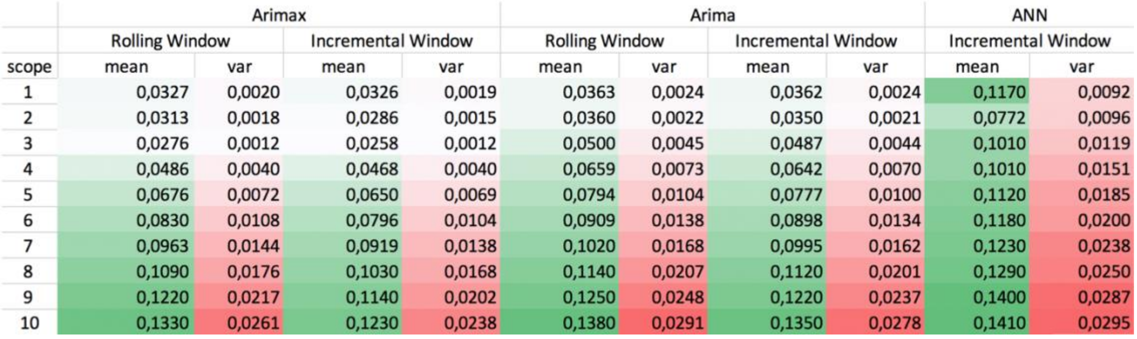


FIGURE 43 - ANN ERROR DENSITY BY SCOPE - INCREMENTAL WINDOW

7.4 Model Comparison

Numerical error and variance assessment and comparison

As it can be seen, all models end up having a very similar error range. What could be said is that the Arima with an Incremental Window is the model with a lower variance throughout all scopes. The Arimax with an Incremental Window is the lowest error model, again, through all scopes.



	Arimax				Arima				ANN	
	Rolling Window		Incremental Window		Rolling Window		Incremental Window		Incremental Window	
scope	mean	var	mean	var	mean	var	mean	var	mean	var
1	0,0327	0,0020	0,0326	0,0019	0,0363	0,0024	0,0362	0,0024	0,1170	0,0092
2	0,0313	0,0018	0,0286	0,0015	0,0360	0,0022	0,0350	0,0021	0,0772	0,0096
3	0,0276	0,0012	0,0258	0,0012	0,0500	0,0045	0,0487	0,0044	0,1010	0,0119
4	0,0486	0,0040	0,0468	0,0040	0,0659	0,0073	0,0642	0,0070	0,1010	0,0151
5	0,0676	0,0072	0,0650	0,0069	0,0794	0,0104	0,0777	0,0100	0,1120	0,0185
6	0,0830	0,0108	0,0796	0,0104	0,0909	0,0138	0,0898	0,0134	0,1180	0,0200
7	0,0963	0,0144	0,0919	0,0138	0,1020	0,0168	0,0995	0,0162	0,1230	0,0238
8	0,1090	0,0176	0,1030	0,0168	0,1140	0,0207	0,1120	0,0201	0,1290	0,0250
9	0,1220	0,0217	0,1140	0,0202	0,1250	0,0248	0,1220	0,0237	0,1400	0,0287
10	0,1330	0,0261	0,1230	0,0238	0,1380	0,0291	0,1350	0,0278	0,1410	0,0295

TABLE 4 - AVERAGE ABSOLUTE ERROR AND VARIANCE FOR ALL FORECASTING METHODS BY SCOPE

Graphical error and variance assessment and comparison

Displaying the table results in a graph some more conclusions can be reached. For instance, all models seem to converge to a similar error the higher the scope is. Therefore, the main difference on accuracy is found when methods are forecasting on lower scopes.

It is also important to highlight that the forecasting methods seem to have a lot more impact than the Window method (Rolling or Incremental).

It can also be said that the hierarchy remains constant through all scopes. This means that if models were ranked from best to worst on the first and last scope, the order would be the same on both rankings.

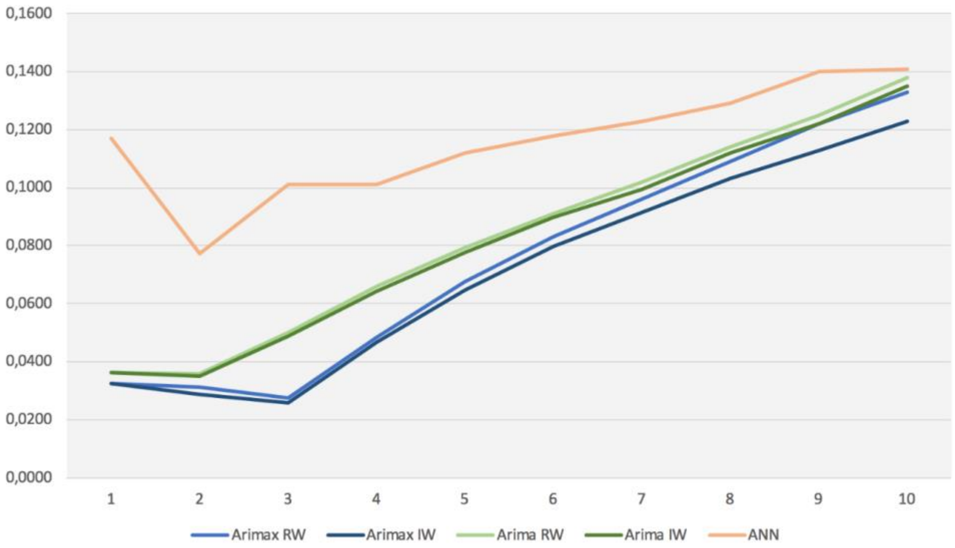


FIGURE 44 - AVERAGE ABSOLUTE ERROR BY FORECASTING METHOD AND SCOPE

When assessing the variance, it can be seen that most conclusions that were reached on the error assessment are valid as well: Arimax is the best model in terms of variance and model hierarchy is kept through scopes. The only difference may be that the convergence is not so clear as it was in the last plot.

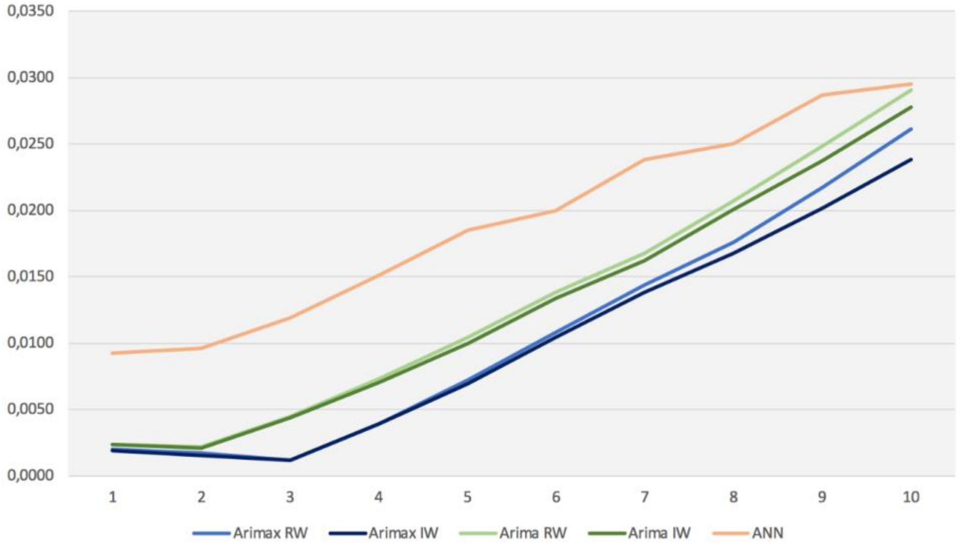


FIGURE 45 - ERROR VARIANCE BY FORECASTING METHOD BY SCOPE

Once it has been seen that Incremental Windows seem to have a better forecast for both Arima and Arimax models, only those are kept with the ANN, and shown in the next plot.

A trend of 1 day forecasts from each of the IW methods is displayed. As it can be seen, the ANN repeatedly misestimates the price, while Arima and Arimax seem to follow the trend adequately.

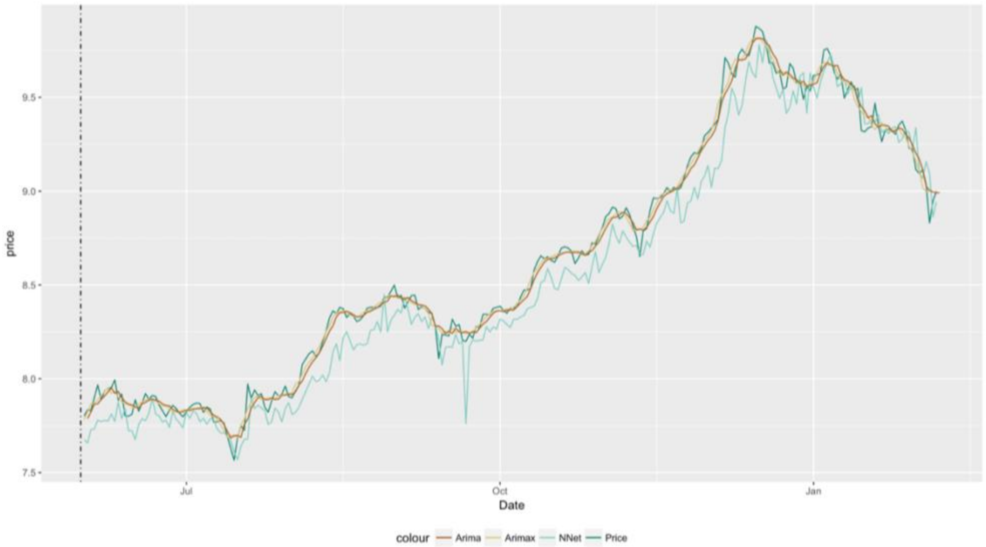


FIGURE 46 - 1 DAY FORECASTS - INCREMENTAL WINDOW METHODS

Finally, as it has been clearly seen that the ANN is not suitable, only IW Arima and Arimax are left to compare. Therefore, a side by side error density is shown, where it can be clearly seen then 2 and 3 day scope of the Arimax model have a better behaviour.

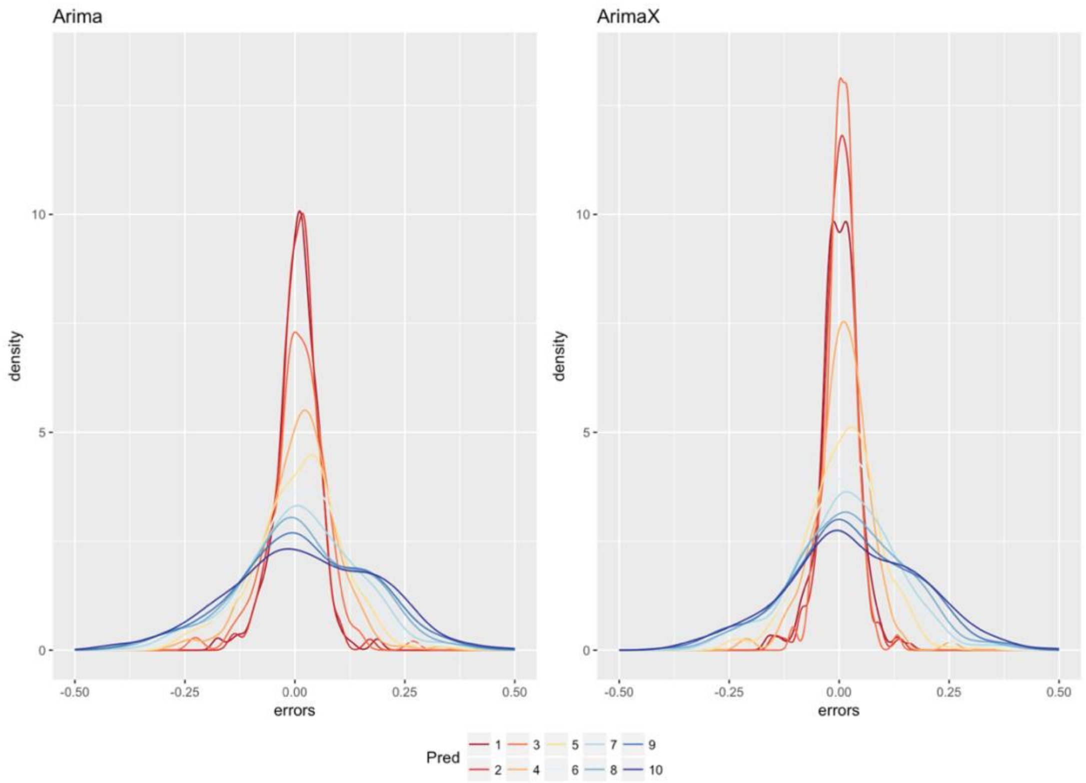


FIGURE 47 - DENSITY COMPARISON BETWEEN ARIMA AND ARIMAX - INCREMENTAL WINDOW

Finally, once the model selection has been done, there is still one last topic to discuss. Which is the assessment of the impact of the variable Bitcoin Bubble on the Causal models. As a consequence, both models are compiled without the variable. In order to keep the extension of this thesis under a reasonable range the table that contains the data from all models will be placed at the appendix. Here, the average absolute error comparison chart is shown.

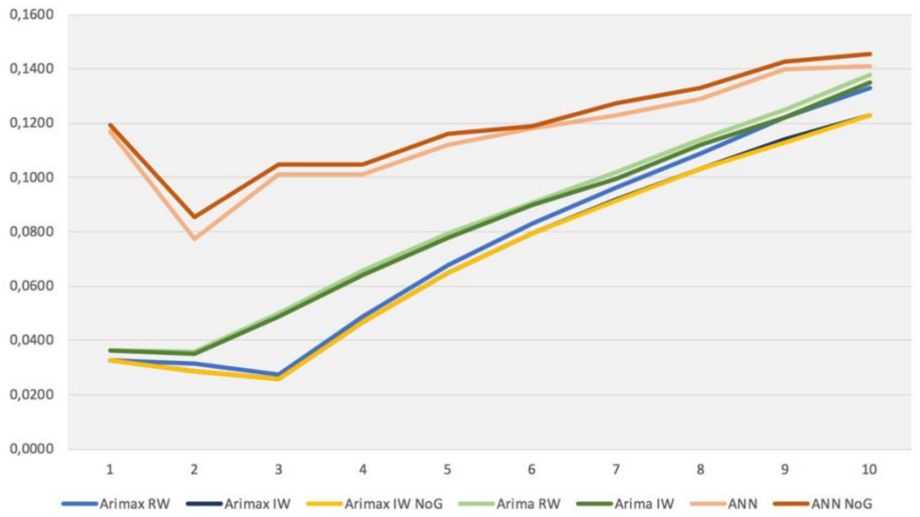


FIGURE 48 - AVERAGE ABSOLUTE ERROR BY FORECASTING METHOD AND SCOPE

As it can be seen, while in the Arimax the removal of the variable has virtually no effect, in the ANN the error is slightly higher through all scopes when the variable is removed. One potential explanation is that even though the ANN is worse as its not build as a time-series forecasting model, it is better at finding causal relations and therefore the removal of a relevant variable changes the outcome significatively.

Regarding the lack of effect to the Arimax errors, it could be expected due to the coefficient value was close to 0 as it can be seen in the parameter stability in section (7.2.1).

On the other hand, as the change in variance is very slight in both cases the plot is added to the appendix as it does not add relevant information. What can be said about error variance is that the removal of the variable does not seem to have a specially relevant effect on the lower scope predictions, while some divergence can be observed on the higher scopes.

8 CRITICAL REFLECTION

This thesis began with two interlaced goals, forecasting the price of Bitcoin & the study of social networks and how they can impact forecasting. The fact that it was a part of a wider scope project made it interesting as it gave a sense of purpose.

As the research kept going, it became certain that the study of social networks as a data source was a very interesting topic but also a very wide one, many different sources and techniques could be used. Finally, the first selected unordinary data source was Google Trends which seemed to be a useful tool to gather overall interest across the world on a certain topic. The following selected data source was Twitter, but due to the slow nature of real time tweet storing it had to be added to the Future Work chapter, as it will be seen below.

Once the main dataset had been built, which contained information from two different sources, the forecasting could begin.

Among many available variables (more than 75 once all of them had been lagged) only a few were automatically selected through a stepwise technique. All of them, but specially the one and two days lags, could have been conceptually selected which gave coherence to the obtained results. The only exception was the seven days lag that was introduced to the transfers volume variables, which might have an unknown weekly behaviour.

What should be specially highlighted is that, from the Google Trends information, the variable Bitcoin Bubble was selected with a two days lag, already indicating its potential effect.

Interestingly, the effect of the variable seemed to be different depending on the model. While the variable was not remarkably impactful in the Arimax model as the errors remained virtually equal when removed, it did make an impact in the Artificial Neural Network.

Also, it should be highlighted that the usage of different models had a greater impact than the inclusion of variables. This conclusion is drawn from the fact that both the ANN and the Arimax had the same variables but got very different results.

The implementation of the Rolling or Incremental Window seemed to have none to little impact. This could potentially change using different width rolling windows, but this path of forecast improving did not seem adequate.

Overall the usage of this source of information, related to extracting sentiment or interest from across all individuals, seems a promising forecasting tool, that should be taken into account specially as more and more population get access to the internet.

Finally, this thesis proceeds to the Future work chapter, the last one, where a final discussion takes place and a sneak peak of how could tweet sentiment be incorporated is given.

9 FUTURE WORK

As this thesis was conceived from the beginning to be an ongoing research rather than an A-to-Z project, the Future Work chapter is one of the most relevant, as it is where many possible paths to follow up are shown. They represent different directions in which the project could keep evolving.

Therefore, many ideas will be shown below. They can be grouped in several groups: Forecast related, Implementation Related, Tweet related

9.1 Forecasting

The main future work should be the usage of Recurrent Neural Networks (RNN) instead of regular ones. This is because ANNs are not designed to estimate forecasts, but to find causal relations. On the other hand, RNNs are specifically designed to make time-series forecasting, and therefore it should be the immediate next step of this thesis if it were to be continued.

Also, computational cost analysis has been one of the main constrains of this thesis because rolling windows are costly, and many models have been calculated. Therefore, it would be interesting to evaluate the forecasting error taking into account the computing time. Specially, this would be interesting if it were to be used as a real time investment advisor.

Finally, two broad ideas that would be interesting to keep working on are trying other forecasting methods and other error assessment techniques. Still, this might not be very important as regarding the errors, the differences between models are wide, and therefore a very precise error assessment does not seem necessary. Regarding other forecasting methods, it is clear that the RNN can already be a good add-up and probably adding more methods would over-diversify the project, taking the focus out of the important points.

9.2 Implementation

The code built during this thesis, would allow to perform real-time data gathering from all sources, Twitter and Google Trends from their API's and Python-scraped for the main database. Therefore, a new approach considering the data as a data stream could be performed, allowing it to be used in real time investment decisions.

This could lead to an investment assistance platform, potentially built on Shiny (29). It would allow to retrieve data from all sources in real time and provide investing advice. Given that the forecast models consistently achieved good forecasts.

Also regarding the implementation, other methods of variable selection could be used as it is clear that the glm-stepwise approach has been used due to it's simplicity and ease of use. More complex techniques could be used, specially with the Neural Networks, while taking into account computational efficiency.

9.3 Social Media

Perform the basic sentiment analysis described in the methodology over a tweet database. In fact, this first idea has already been executed, and some insights will be shown at the end of this chapter.

One of the main proposals on which to keep working on is to build a more sophisticated Sentiment Analysis, for instance, using a financial library or using more complex techniques that take into account interactions between words.

9.4 Tweet Sentiment Analysis

As it has been said throughout this thesis, Twitter can be a very valuable source of information. Unfortunately, not enough Twitter data were available when this thesis was handed, and as a consequence it could not be included in the forecasting techniques. The following paragraphs explain how this thesis would keep working on this area:

More tweets should be collected, until reaching a critical mass that could be used in forecasting. Also, a very interesting add-up could be the possibility to identify several groups of tweets in order to filter them. For instance, advertisements of cryptocurrencies exchanges would likely be grouped together and filtered out as the information they provide is unlikely to be related with Bitcoin prices. This could be done through a hierarchical clustering technique.

Finally, even if the data range was not enough to include in the Forecasting and therefore pointless to include in the central chapters of the thesis, some sentiment analysis of the available tweets has been done. In a way, to exemplify that this project is indeed ongoing. Therefore, the following plots give a glimpse of what could be done.

The available tweets have been gathered from 1st of April of 2018 to 22nd of May. Over one million tweets were collected, which end up being one million when repeated tweets are removed.

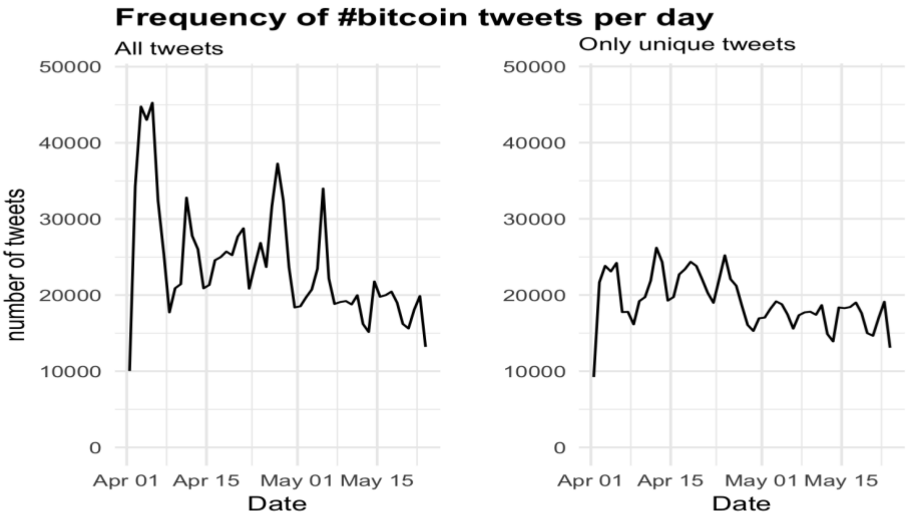


FIGURE 49 - #BITCOIN TWEET VOLUME THROUGH TIME - BEFORE AND AFTER FILTERING

The following, are the ordered word frequency from all tweets. As it can be seen, a lot of words do not provide information, and therefore are filtered. As it can be seen, a common topic word is *cybersecurity*, which could indicate people is worried about security on cryptocurrencies.

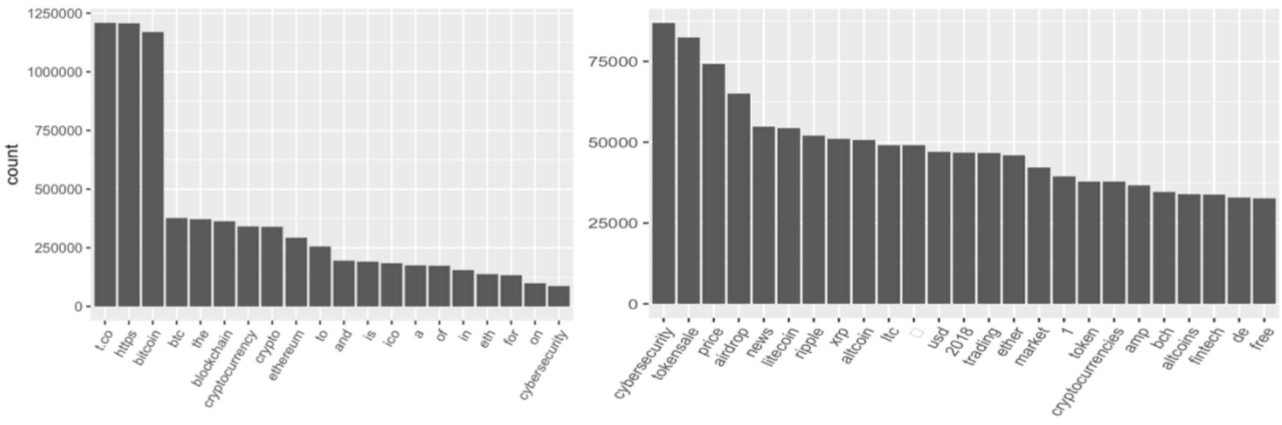


FIGURE 50 - WORD FREQUENCY IN TWEETS ANALYSIS

The following plot illustrates the sentiment of the tweets which has been obtained as explained in the methodology chapter. As it can be seen, most of the tweets appear to have a neutral sentiment. Also, positive tweets appear to be consistently over negative tweets.

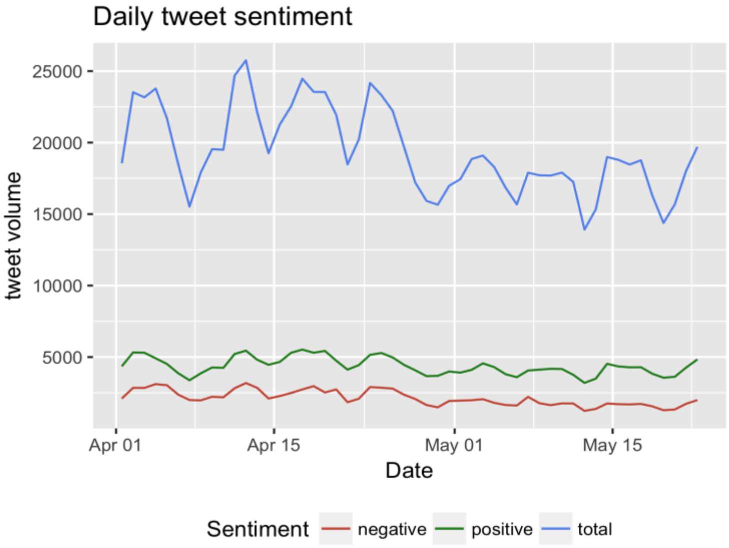


FIGURE 51 - TWEET VOLUME BY BINARY SENTIMENT

In order to get more detailed information about the sentiment, the "total" group is substracted and the plot is done again. Now something else can be observed, which is a slight divergence between both series, potentially indicating a future price recovery if tweet sentiment ends up being related to price behaviour.

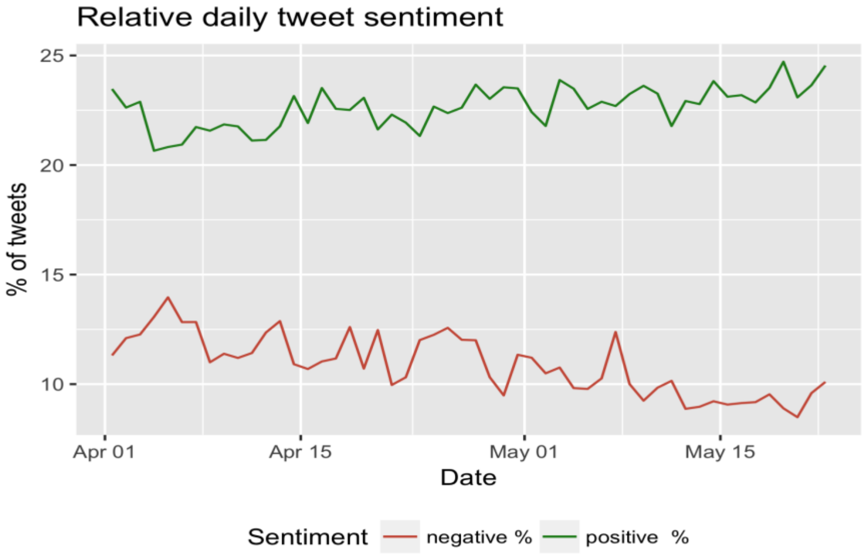


FIGURE 52 - PERCENTAGE OF TWEETS (POSITIVE OR NEGATIVE) RESPECT THE TOTAL

Once the binary sentiment has been shown, which has been extracted from the nrc library in a numeric format and then converted to a binary variable, the next step is to dive into a multi-categorical sentiment approach.

Several facts can be observed. On one hand, the variability is higher. This is probably because words from the afinn library, which is the categorical one, have more than one sentiment attached. The result is that when positive words are used the score goes higher than with the numerical library.

On the other hand, it can be seen that also in this case, positive feelings seem to dominate over negative ones through all the series. This could be considered a way to check how robust the used libraries are. As if behaviour changed between libraries it could be concluded that they have too much impact on the results.

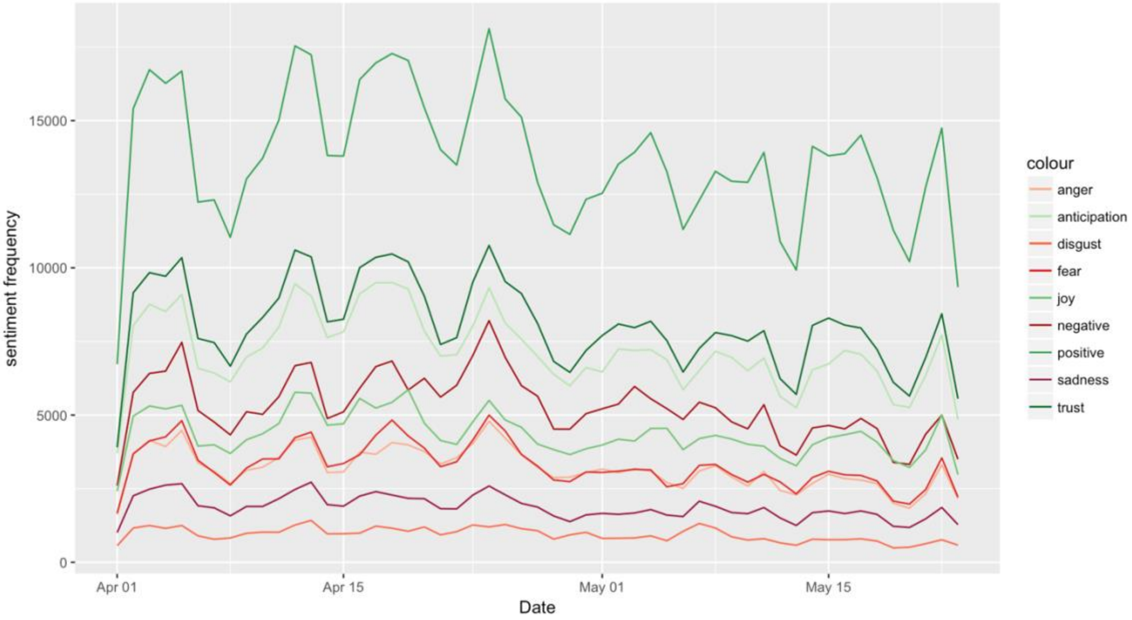


FIGURE 53 - DETAILED MULTICATEGORICAL TWEET SENTIMENT FREQUENCY THROUGH TIME

9.5 Rejoinder

As it has been seen in many plots during this thesis, Bitcoin prices boomed during 2017. Since then, a substantial decrease has happened during this year. This decrease can be seen in the following plot, which uses the new data batch directly scraped from coinmarketcap.

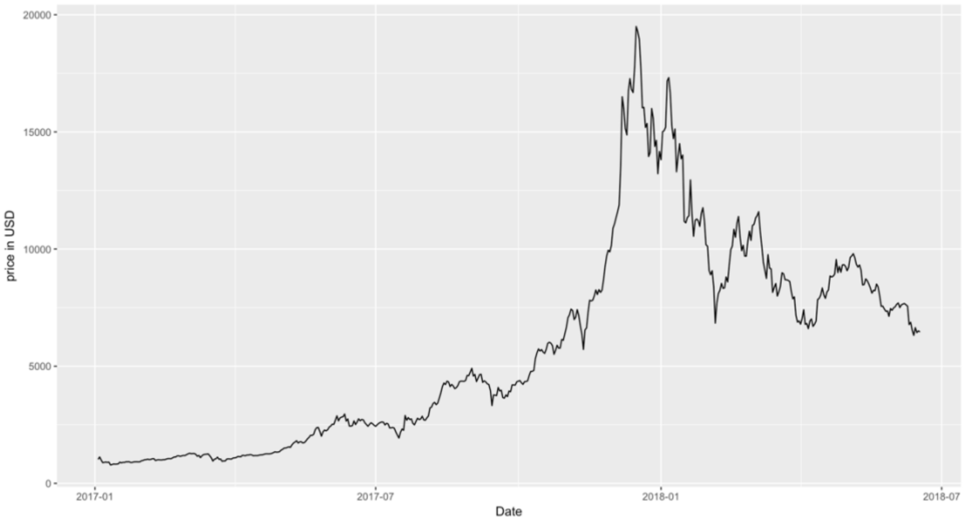


FIGURE 54 - BITCOIN PRICE IN USD - 01/2017 - 07/2018

As I have kept trying to model and forecast the series, I have experienced the difficulties of forecasting under such a volatile context. Soon, the tweet volume will be enough to be added to the models so the project can keep going forward. Also, with the new web scraping tool, real time forecasting will be an option as all data will be able to be obtained in real time. Probably then the computational time of the models will have to be assessed so it doesn’t become an obstacle in the real-time implementation.

To sum up, I would like to say that the fact of being a part of a wider scope projects is what makes this thesis exciting, and that gives me even more energy to keep working on this interesting area, where economics and statistics are interlaced.

10 BIBLIOGRAPHY

All references are written by the Vancouver methodology guidelines through the Mendeley platform and have been written down in appearance order.

1. Reifschneider D, Tulip P. Gauging the Uncertainty of the Economic Outlook Using Historical Forecasting Errors: The Federal Reserve's Approach. Financ Econ Discuss Ser [Internet]. 2017;2017(020). Available from: http://www.federalreserve.gov/econresdata/feds/2017/files/2017020pap.pdf

2. Mourelatos APD. The Ancients''Meteorology': Forecasting and Cosmic Natural History. Rhizai A J Anc Philos Sci. 2005;2:279–91.

3. Malkiel BG, McCue K. A random walk down Wall Street. Vol. 8. Norton New York; 1985.

4. Mauldin J. 8 charts prove economic forecasting doesn't work - Business Insider [Internet]. 2016 [cited 2018 Jun 15]. Available from: http://www.businessinsider.com/8-charts-prove-economic-forecasting-doesnt-work-2016-1?IR=T

5. Edwards RD, Magee J, Bassetti WHC. Technical analysis of stock trends. CRC press; 2007.

6. Tucker C. Can social media data be used to predict threats or identify fake news? | Penn State University [Internet]. 2018 [cited 2018 Jun 20]. Available from: https://news.psu.edu/story/504004/2018/02/15/research/can-social-media-data-be-used-predict-threats-or-identify-fake-news

7. Söhler E, Buelens B. Social media as a data source for official statistics ; the Dutch Consumer Confidence Index. 2016;(12).

8. Sismeiro C. Can Social Networks Help Content Websites Predict Traffic and Engagement? 2015;(May):1–44.

9. Gibert K, Horsburgh JS, Athanasiadis IN, Holmes G. Environmental Data Science. Environ Model Softw [Internet]. 2018;106:4–12. Available from: https://doi.org/10.1016/j.envsoft.2018.04.005

10. Gibert K, Sànchez-Marrè M, Izquierdo J. A survey on pre-processing techniques: Relevant issues in the context of environmental data mining. AI Commun. 2016;29(6):627–63.

11. Kearney MW. rtweet: Collecting Twitter Data [Internet]. 2017. Available from: https://cran.r-project.org/package=rtweet

12. Allison P. Imputation by Predictive Mean Matching: Promise &amp; Peril | Statistical Horizons [Internet]. 2015 [cited 2018 May 20]. Available from: https://statisticalhorizons.com/predictive-mean-matching

13. van Buuren S, Groothuis-Oudshoorn K. mice: Multivariate Imputation by Chained Equations [Internet]. 2018. Available from: https://cran.r-project.org/package=mice

14. Grothendieck G. sqldf: Manipulate R Data Frames Using SQL [Internet]. 2017. Available from: https://cran.r-project.org/package=sqldf

15. Robinson D, Silge J. tidytext: Text Mining using "dplyr", "ggplot2", and Other Tidy Tools [Internet]. 2018. Available from: https://cran.r-project.org/package=tidytext

16. Maechler M, Rousseeuw P, Struyf A, Hubert M. cluster: “Finding Groups in Data”: Cluster Analysis Extended Rousseeuw et al. [Internet]. 2018. Available from: https://cran.r-project.org/package=cluster

17. R Core Team. R: A Language and Environment for Statistical Computing [Internet]. Vienna, Austria; 2018. Available from: https://www.r-project.org/

18. Ward Jr JH. Hierarchical grouping to optimize an objective function. J Am Stat Assoc. 1963;58(301):236–44.

19. Dunteman GH. Principal components analysis. Sage; 1989.

20. Zivot E, Wang J. Rolling Analysis of Time Series. Model Financ Time Ser with S-PLUS. 2006;313–60.

21. Box GEP, Jenkins GM, Reinsel GC, Ljung GM. Time series analysis: forecasting and control. John Wiley & Sons; 2015.

22. Hyndman R, Athanasopoulos G, Bergmeir C, Caceres G, Chhay L, O’Hara-Wild M, et al. forecast: Forecasting Functions for Time Series and Linear Models [Internet]. 2018. Available from: https://cran.r-project.org/package=forecast

23. Wickham H, François R, Henry L, Müller K. dplyr: A Grammar of Data Manipulation [Internet]. 2018. Available from: https://cran.r-project.org/package=dplyr

24. Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep learning. Vol. 1. MIT press Cambridge; 2016.

25. from Jed Wing MKC, Weston S, Williams A, Keefer C, Engelhardt A, Cooper T, et al. caret: Classification and Regression Training [Internet]. 2018. Available from: https://cran.r-project.org/package=caret

26. Massicotte P, Eddelbuettel D. gtrendsR: Perform and Display Google Trends Queries [Internet]. 2018. Available from: https://cran.r-project.org/package=gtrendsR

27. • Global social media ranking 2018 | Statistic [Internet]. 2018 [cited 2018 Jun 12]. Available from: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

28. Wei T, Simko V. corrplot: Visualization of a Correlation Matrix [Internet]. 2017. Available from: https://cran.r-project.org/package=corrplot

29. Chang W, Cheng J, Allaire JJ, Xie Y, McPherson J. shiny: Web Application Framework for R [Internet]. 2018. Available from: https://cran.r-project.org/package=shiny

30. Kaggle: Your Home for Data Science [Internet]. [cited 2018 Apr 10]. Available from: https://www.kaggle.com/

31. Cryptocurrency Market Capitalizations | CoinMarketCap [Internet]. [cited 2018 Apr 10]. Available from: https://coinmarketcap.com/

32. Google Trends [Internet]. [cited 2018 Apr 10]. Available from: https://trends.google.es/trends/?geo=ES

11 TABLE OF FIGURES AND TABLES

11.1 Figures

11.2 Tables

12 APPENDIX

This thesis appendix main objective is to include the procedure to obtain the Arima and Arimax Structure in order to keep the focus on the relevant information on the main ideas of the thesis. Also, some extra figures will be shown. Finally before proceeding, it has to be stated that figures shown here will not be present at the Table of Figures and Tables of the thesis.

12.1 Arima structure

In this section the Arima's structure selection procedure is explained. Which will be done both conceptually and through the *auto.arima* function. Finally, a table will show how the different structures compare to each other in terms of goodness of fit, which will be assessed through the Akaike Information Criterium (AIC).

Time-series decomposition

Deconstructing the series into season, trend, cycle and error can help prepare the model.

The construction can be done through two methods:

- **additive** as $Price_t = T_t + S_t + E_t$. Appropiate when the seasonal or trend component is not proportional to the level of the series.
- **multiplicative** as $Price_t = T_t * S_t * E_t$. When the seasonality component changes over the trend.

Where T = Trend, S = Seasonality, E=Error. Usually Cicle is included in the Trend.

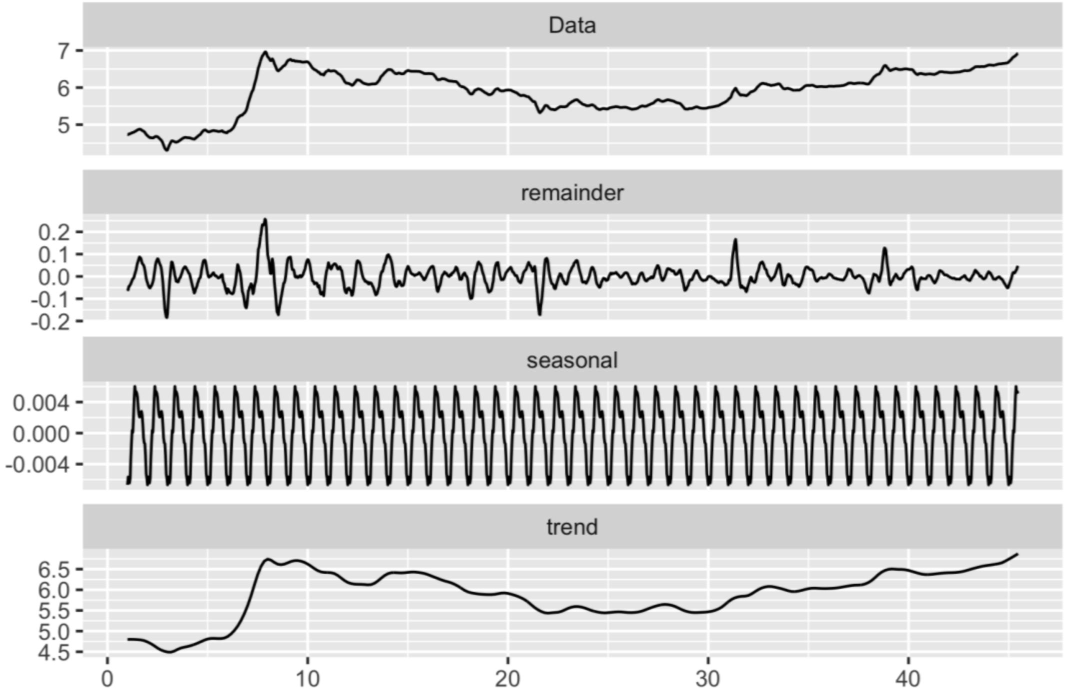


FIGURE 55 - PRICE DECOMPOSITION

4 - Stationarity

First of all, stationary should be checked as it is one of the requirements when fitting an ARIMA. Which means that the series mean, variance, and autocovariance are time invariant.

This will be done using an Augmented Dickey-Fuller test, which checks if a change in price can be explained by its lagged value and a linear trend.

H_0: the series is non-stationary

H_1: the series is stationary

ADF= -2.12   p-value=0.5273

With a 0.5272965 p-value, the H_0 of non-stationarity can't be rejected. As a consequence, in order to be able to model through an ARIMA, the time series will have to be differentiated.

Differentiation is a simple transformation which consists on substracting to the value of a period, the previous period's values. So the differentiated price would be calculated as: $Price_{dt} = Price_t - Price_{t-1}$. The higher the differentiation order, the more lags of the variable included.

5 - Autocorrelations and Choosing Model Order

In order to determine the order parameters of the ARIMA model two main plots will be used. On one hand, ACF (Auto Correlation Function) plots will be used to select the MA(q) as they show the correlation between a series and its lags. On the other, PACF (Partial Auto Correlation Function) plots will be used to select the AR(p) as they show the correlation between the correlation of a variable and its lags that is not explained by previous lags.

The dot lines indicate the 95% significance boundary.

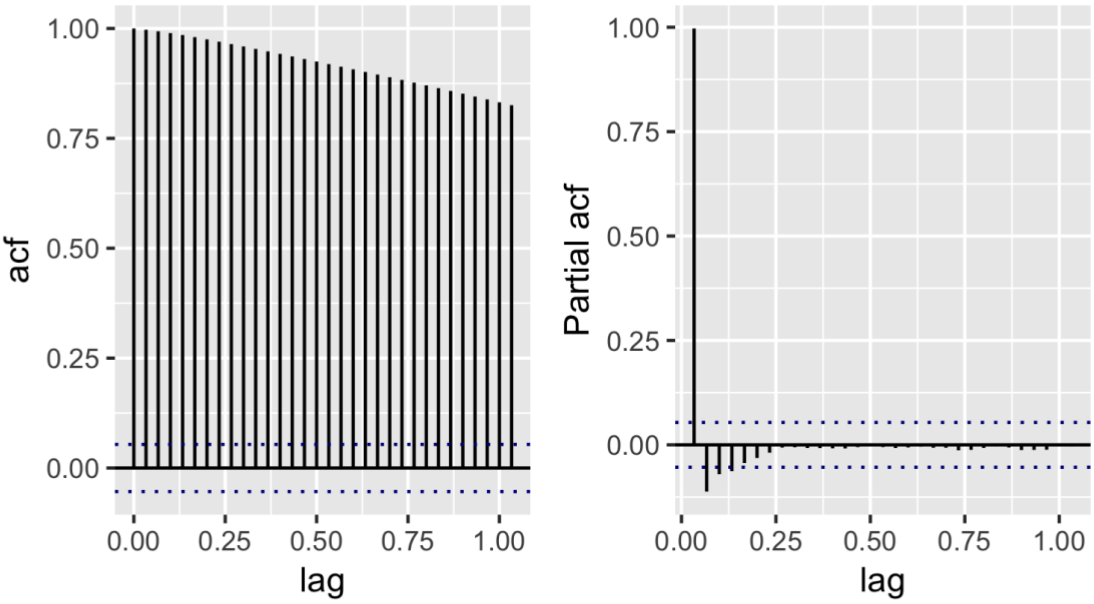


FIGURE 56 - ACF & PACF

It is clear that differentiating is necessary. Therefore, a first order differenciation will be applied and then it will be assesed again if a higher order differentiation is needed.

ADF= -7.317 p-value=0.01

The obtained p-value is now lower than 0.05. As a consequence, H_0 is rejected and it can be said that the first order differentiation achieves to make the serie stationary.

Now, the next step is to evaluate the ACF and Partial ACF plots of the differentiated serie:

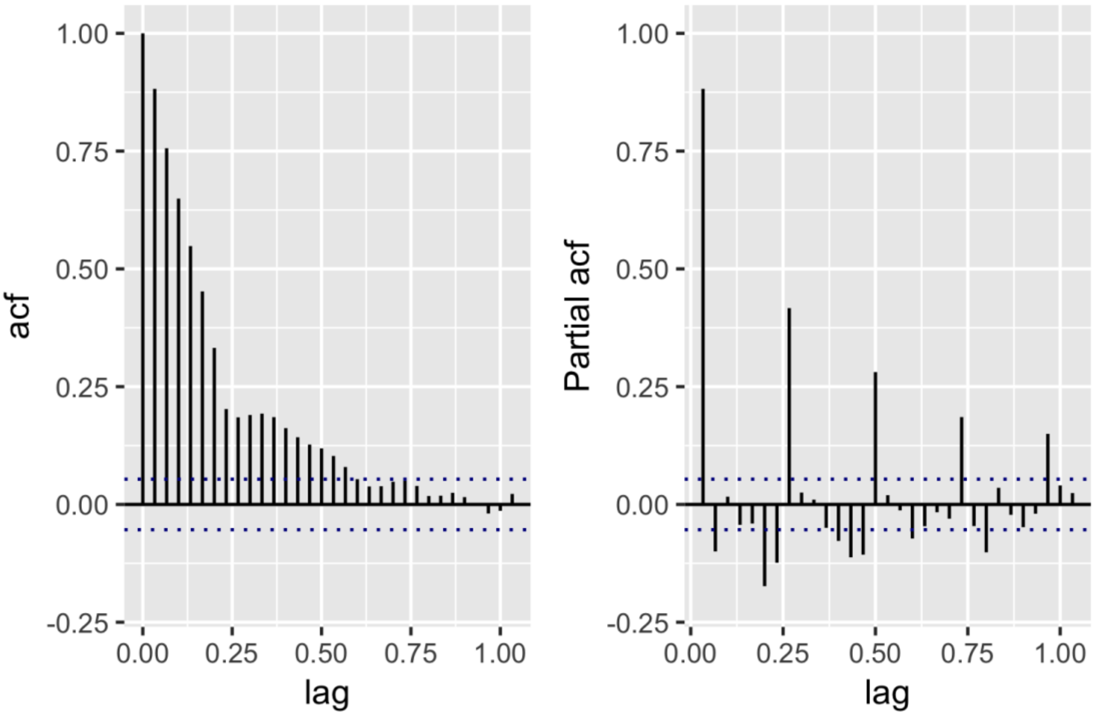


FIGURE 57 – FIRST ORDER DIFFERENTIATION ACF & PACF

Even though the ADF's H_0 test had been rejected, the series still has positive autocorrelations out to a high number of lags, and therefore a differentiation of order 2 is performed.

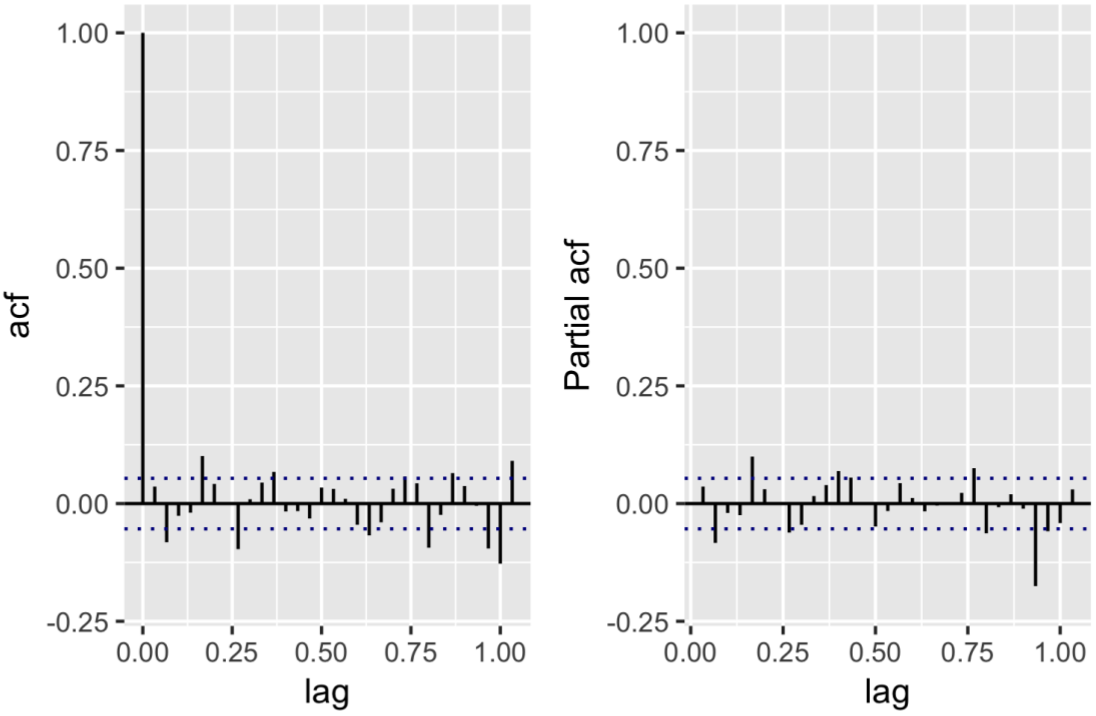


FIGURE 58 – SECOND ORDER DIFFERENTIATION ACF & PACF

The obtained plots suggest we might want to fit an ARIMA model with AR of order 1.

6 - Fitting the ARIMA model

In this stage, the fitting of the ARIMA model will be performed in 2 different ways:

- Using the auto.arima() function, to obtain the combination of parameters which optimizes model fit criteria (Akaike Information Criterium in this case). It is important to state that the default maximum parameter order is set to 5.

- Using the information extracted from the proccess done in the previous stages.

All ARIMAs will be calculated with the *whole* serie. But we have to keep in mind that in further staged of the project the calculations will be done only taking the “train” part.

Auto Arima

The result of the auto.arima function is a (2,1,1) with drift Arima structure.

The obtained model (ignoring the drift) can be written as:

$$\widehat{Price}_{dt} = 0.63P_{t-1} - 0.19P_{t-2} + \epsilon + 0.33\epsilon_{t-1}$$

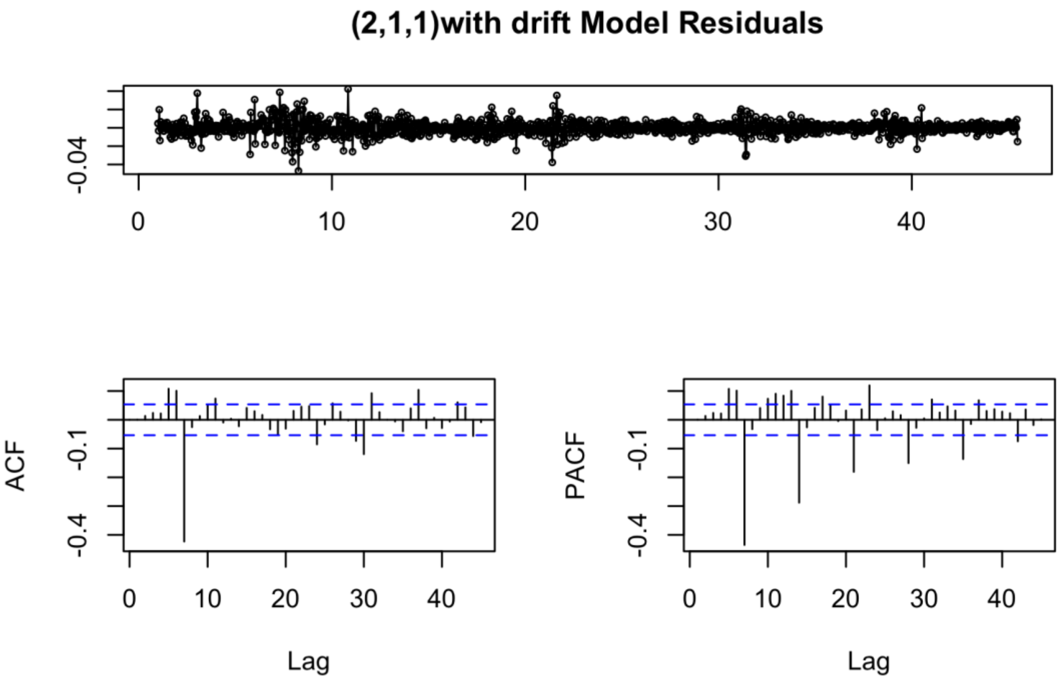
This means that the Price estimation on period t, is calculated by a lineal combination of the price in the last 2 periods and depends on the previous error. The price has one differentiation.

```
##   sigma^2   estimated   as   5.529e-05:     log   likelihood=4647.18
## AIC=-9284.37   AICc=-9284.32   BIC=-9258.39
```

7 - Evaluate and Iterate

Once the model has been fitted with the auto.arima function, we will evaluate it using again the acf and pacf plots:

autoArima:

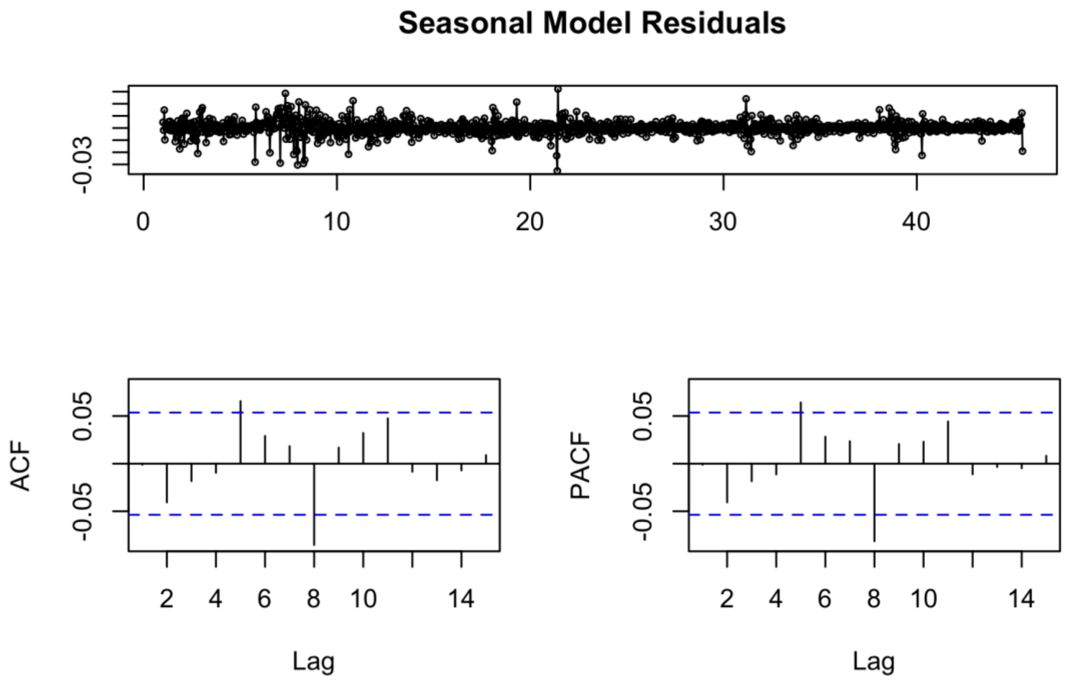


It can be seen that there is a clear pattern at lag 8 and repeats every 7 lags. Therefore, the model could be better specified with AR or MA parameter's equaled to 7.

An Arima (2,1,7) is calculated in order to asses if a lower AIC can be achieved while a a better behaviour of the residuals is obtained. In the following figuers, it can clearly be seen that both premises are attained:

ARIMA(2,1,7):

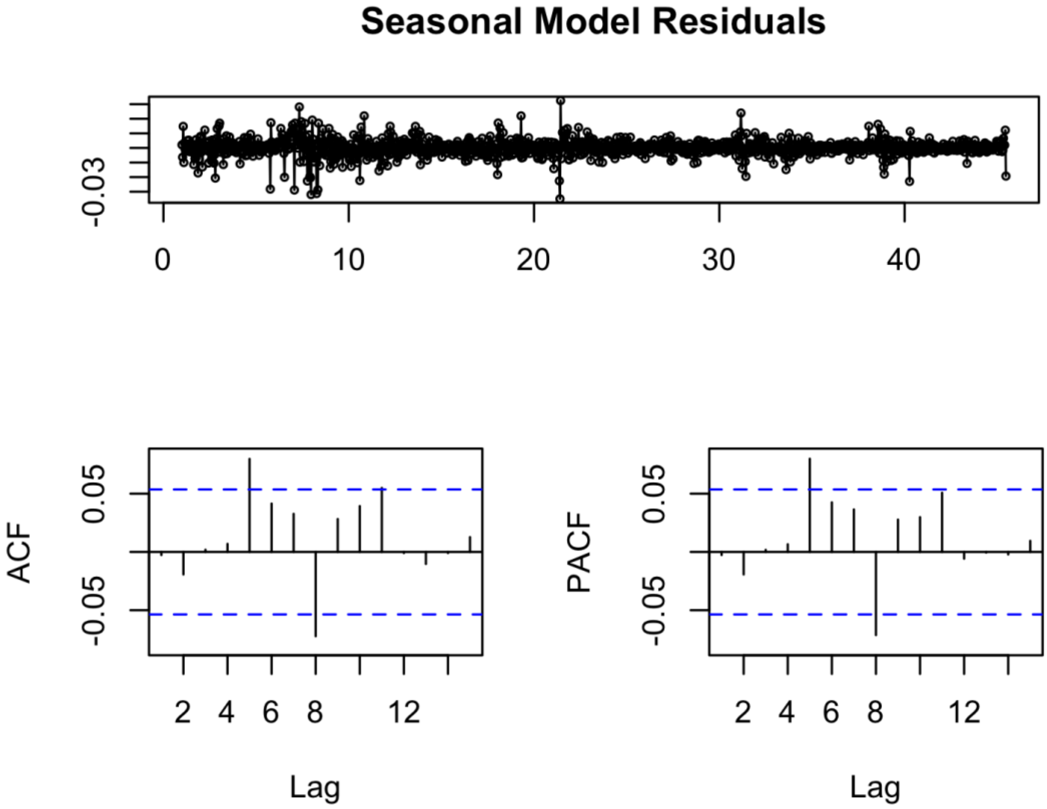
```
Coefficients:
##          ar1      ar2     ma1     ma2     ma3     ma4     ma5     m
a6
##       0.9528  -0.0038  0.0733  0.0746  0.0808  0.0796  0.0864  0.08
86
## s.e.  0.0509   0.0297  0.0442  0.0427  0.0443  0.0448  0.0430  0.04
34
##           ma7
##       -0.9185
## s.e.   0.0453
##
## sigma^2 estimated as 3.083e-05:  log likelihood = 5018.64,  aic = -
10017.27
```



Finally, one last ARIMA model is performed using the analysis done in the first stages of this chapter (doing a 2nd order differentiation.)

ARIMA(1,2,7):

```
## Coefficients:
##          ar1     ma1     ma2     ma3     ma4     ma5     ma6      m
a7
##       0.0276  0.0079  0.0099  0.0163  0.0149  0.0215  0.0237  -0.98
41
## s.e.  0.0302  0.0147  0.0132  0.0156  0.0158  0.0134  0.0144   0.01
67
##
## sigma^2 estimated as 3.109e-05:  log likelihood = 5009.4,  aic = -1
0000.8
```



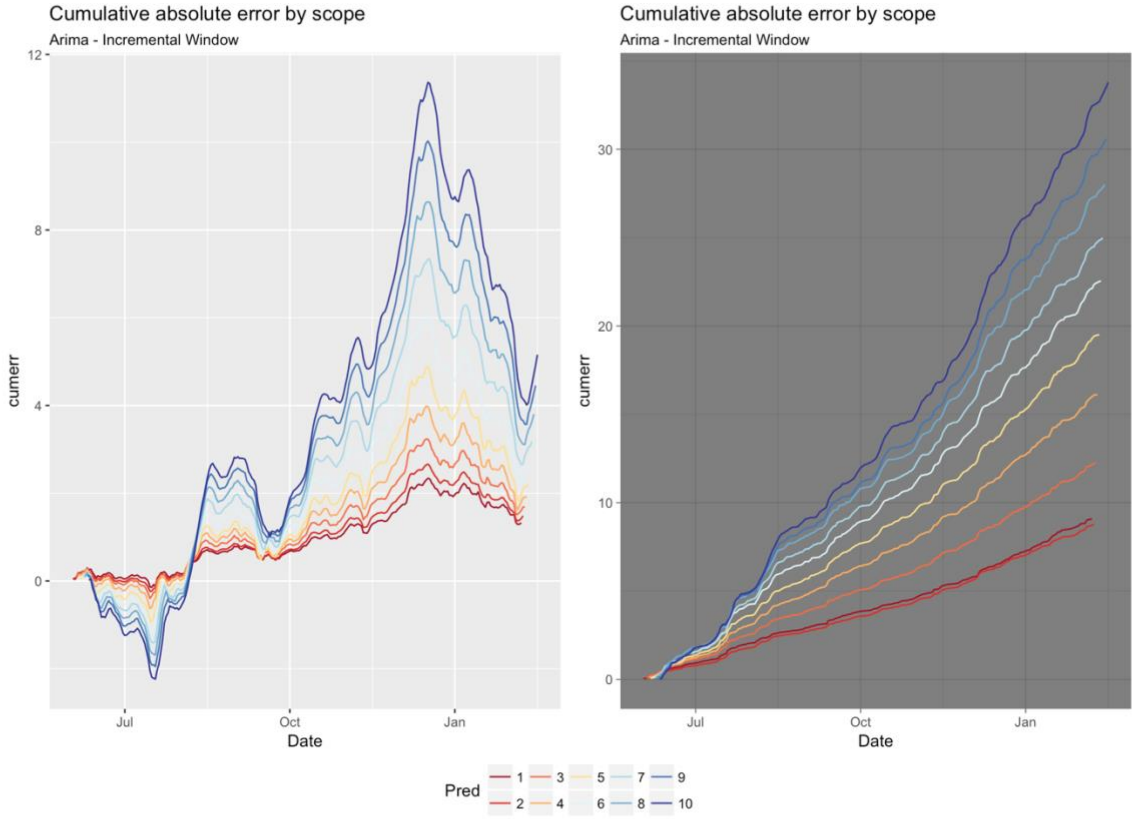
Between the three fitted ARIMAS, clearly the better one is the second: ARIMA (2,1,7). It has the lower AIC and the better residuals behaviour. The order 2 differentiation ARIMA also obtains a good fit, but it obtains a higher AIC than the (2,1,7) as it is probably over-differentiated.

Therefore, from now on the ARIMA (2,1,7) is the one that will be used.

Finally in the modelling stage, a seasonal ARIMA is calculated using the auto.arima function. The obtain AIC is -9305, which is higher than in the ARIMA(2,1,7) and therefore the current study doesn't follow this line of reasoning of trying to add a seasonal component.

Once the model has been chosen, it will be used in different ways. Specifically, using different kind of rolling windows, which are a way of re-estimating the parameters of the arima model while keeping the structure constant.

12.2 Additional Forecasting Plots



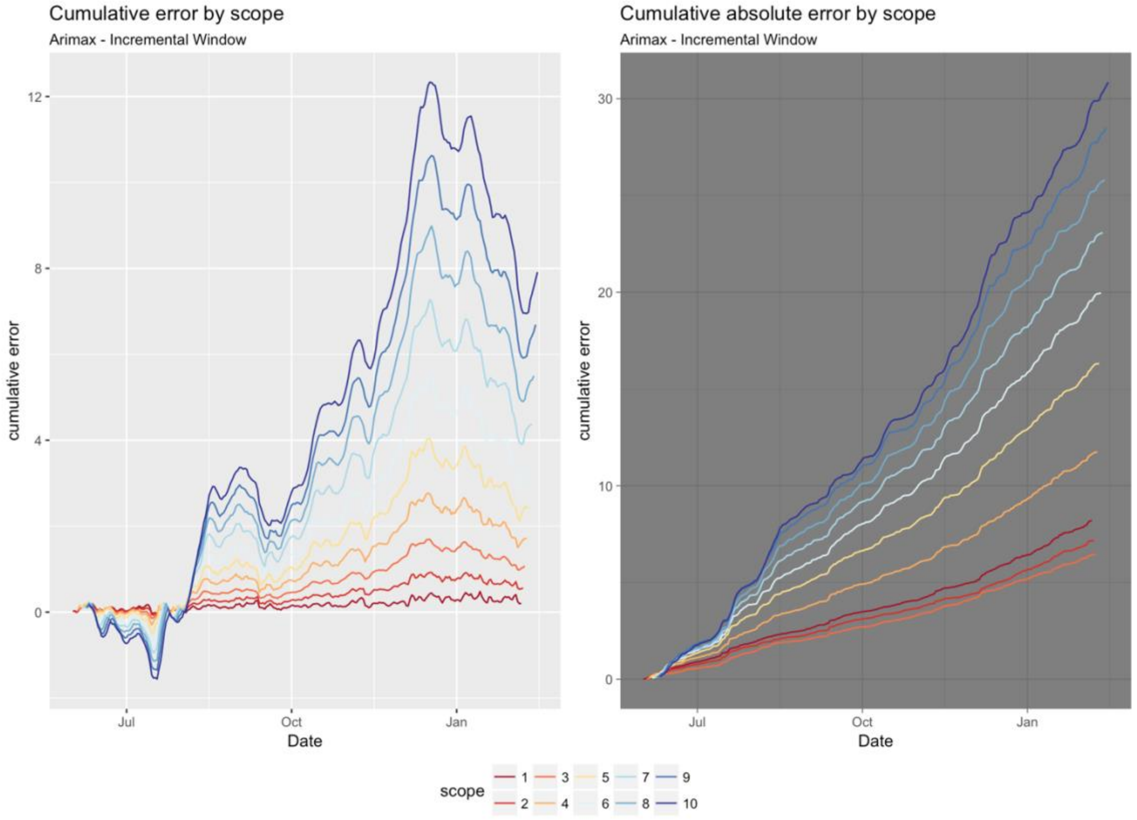
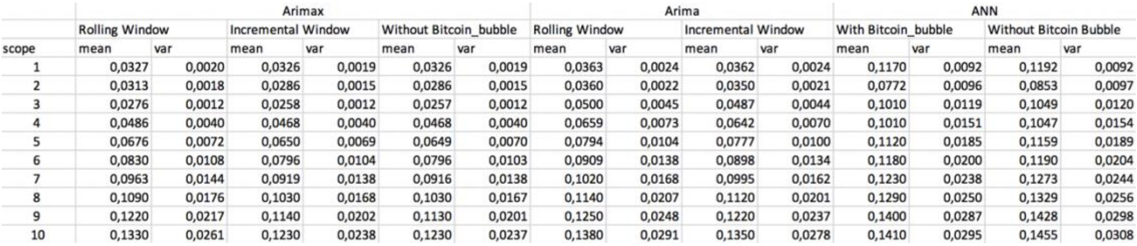
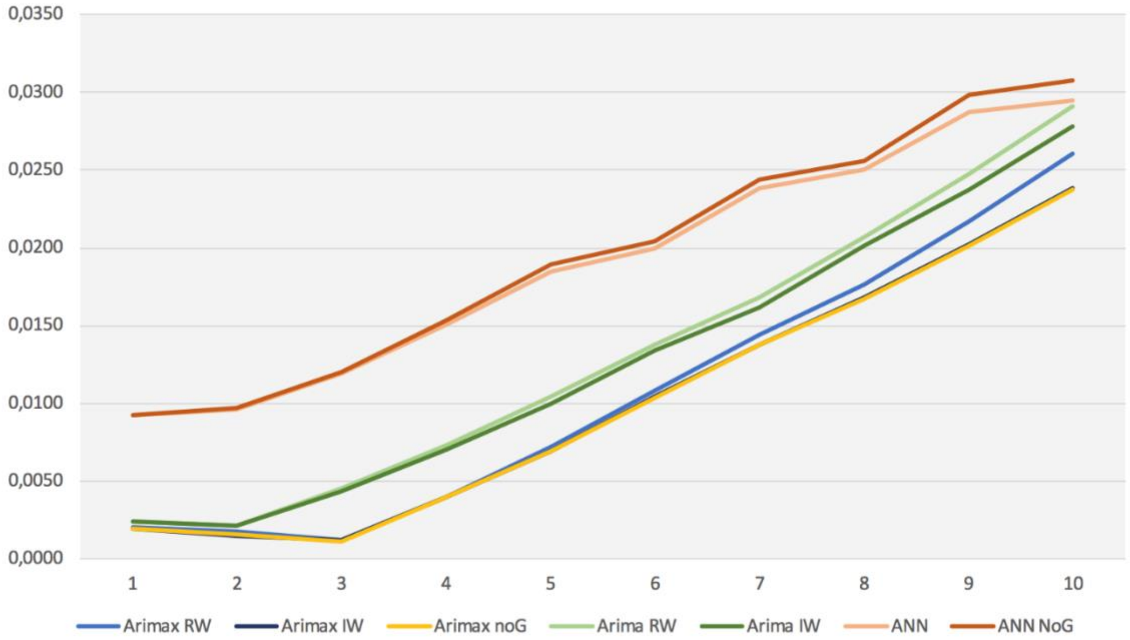


Chart containing Absolute Average Errors and Error Variance from all models and all scopes. Below, the Error Variance among models and scopes is ploted.

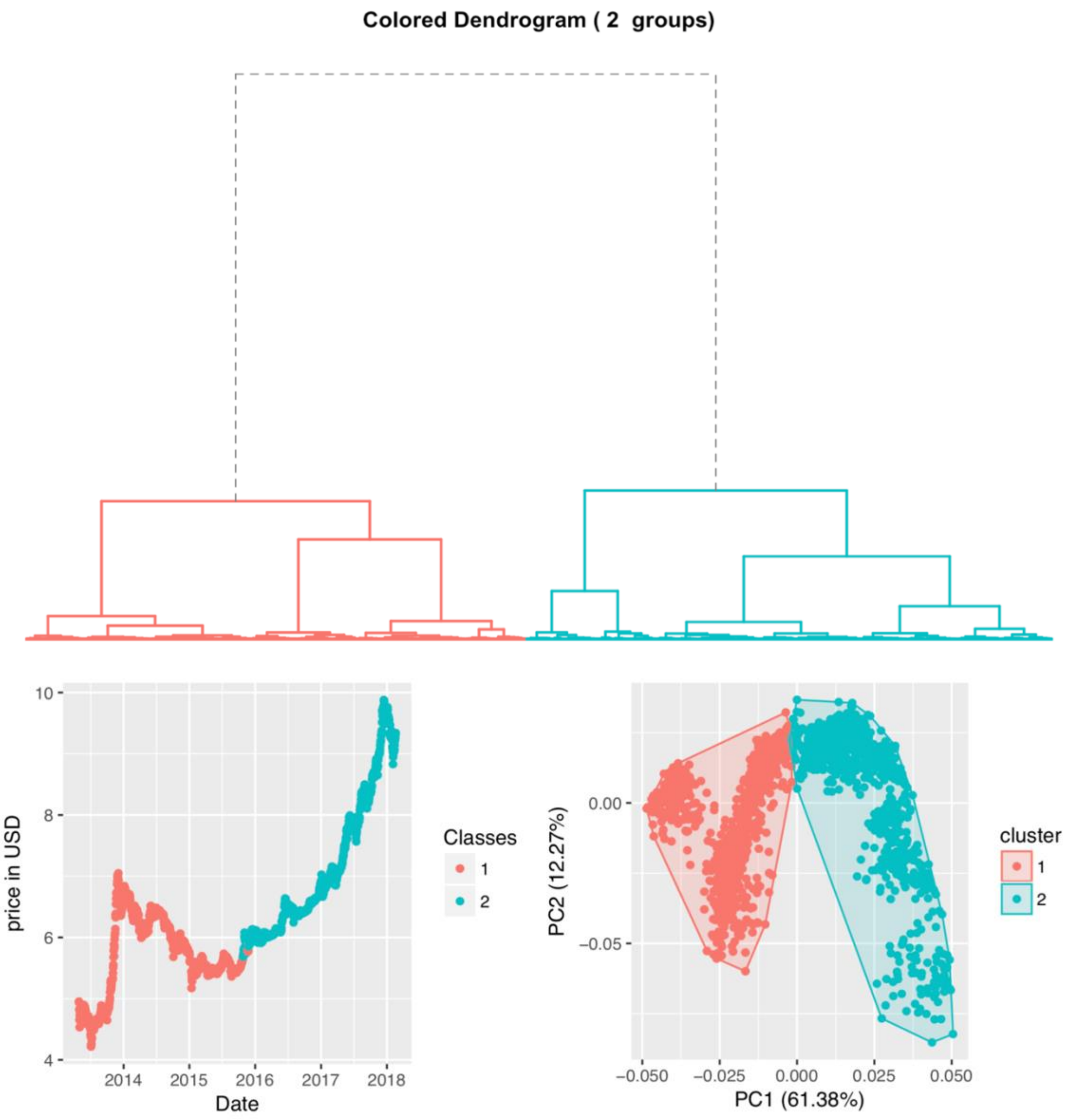


	Arimax						Arima				ANN			
	Rolling Window		Incremental Window		Without Bitcoin_bubble		Rolling Window		Incremental Window		With Bitcoin_bubble		Without Bitcoin Bubble	
scope	mean	var	mean	var	mean	var	mean	var	mean	var	mean	var	mean	var
1	0,0327	0,0020	0,0326	0,0019	0,0326	0,0019	0,0363	0,0024	0,0362	0,0024	0,1170	0,0092	0,1192	0,0092
2	0,0313	0,0018	0,0286	0,0015	0,0286	0,0015	0,0360	0,0022	0,0350	0,0021	0,0772	0,0096	0,0853	0,0097
3	0,0276	0,0012	0,0258	0,0012	0,0257	0,0012	0,0500	0,0045	0,0487	0,0044	0,1010	0,0119	0,1049	0,0120
4	0,0486	0,0040	0,0468	0,0040	0,0468	0,0040	0,0659	0,0073	0,0642	0,0070	0,1010	0,0151	0,1047	0,0154
5	0,0676	0,0072	0,0650	0,0069	0,0649	0,0070	0,0794	0,0104	0,0777	0,0100	0,1120	0,0185	0,1159	0,0189
6	0,0830	0,0108	0,0796	0,0104	0,0796	0,0103	0,0909	0,0138	0,0898	0,0134	0,1180	0,0200	0,1190	0,0204
7	0,0963	0,0144	0,0919	0,0138	0,0916	0,0138	0,1020	0,0168	0,0995	0,0162	0,1230	0,0238	0,1273	0,0244
8	0,1090	0,0176	0,1030	0,0168	0,1030	0,0167	0,1140	0,0207	0,1120	0,0201	0,1290	0,0250	0,1329	0,0256
9	0,1220	0,0217	0,1140	0,0202	0,1130	0,0201	0,1250	0,0248	0,1220	0,0237	0,1400	0,0287	0,1428	0,0298
10	0,1330	0,0261	0,1230	0,0238	0,1230	0,0237	0,1380	0,0291	0,1350	0,0278	0,1410	0,0295	0,1455	0,0308



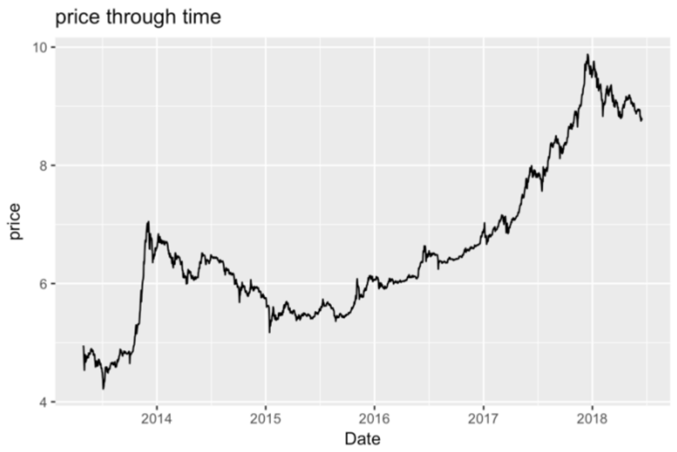
12.3 Additional Exploratory Plots

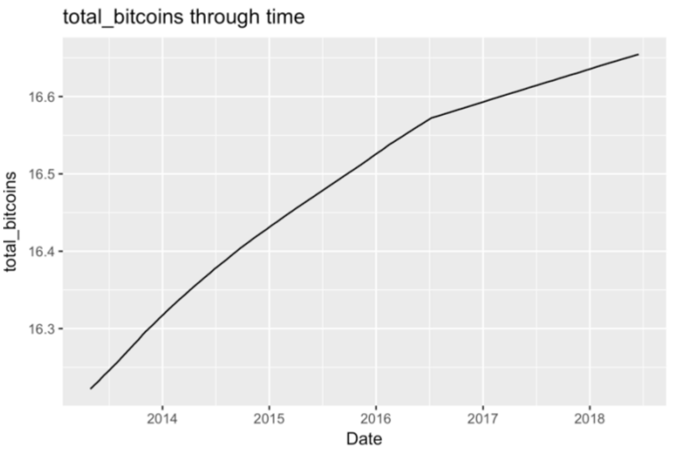
2 group dendogram:

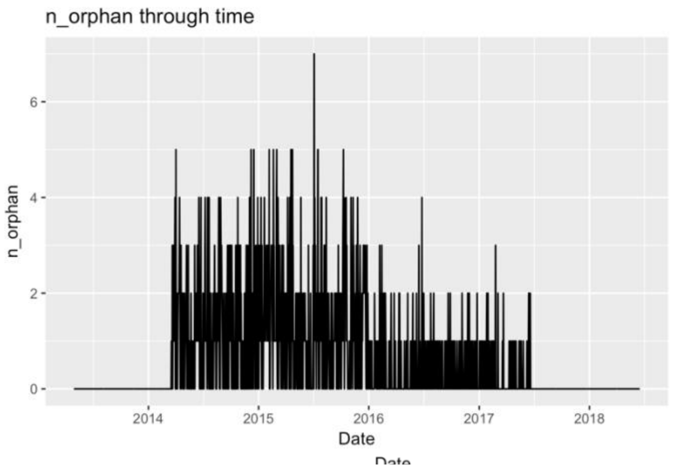


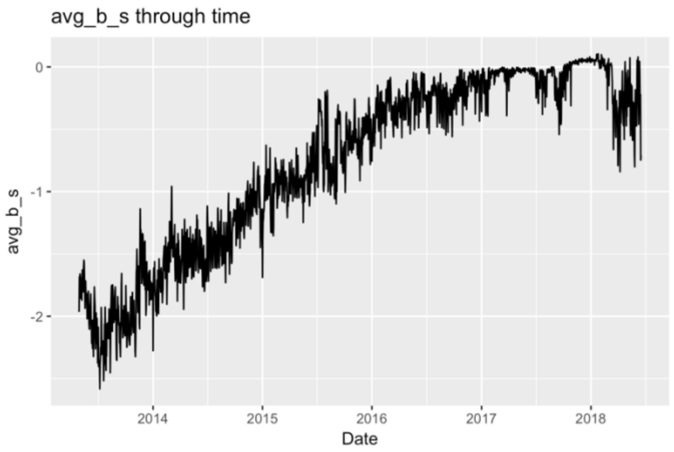
12.4 Bivariate plots

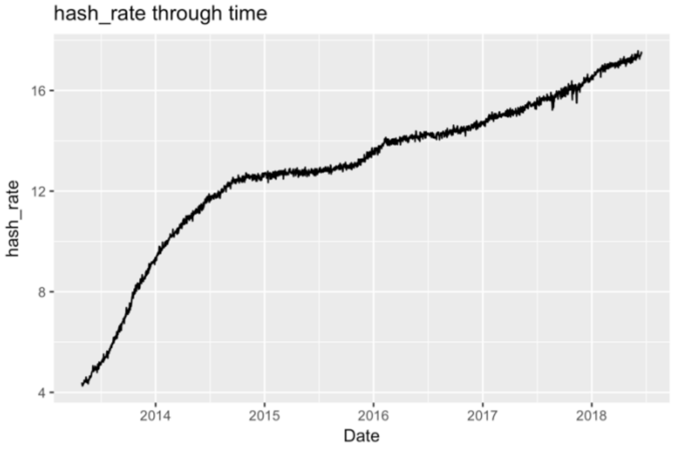
All variables available variables are plotted against the Date.

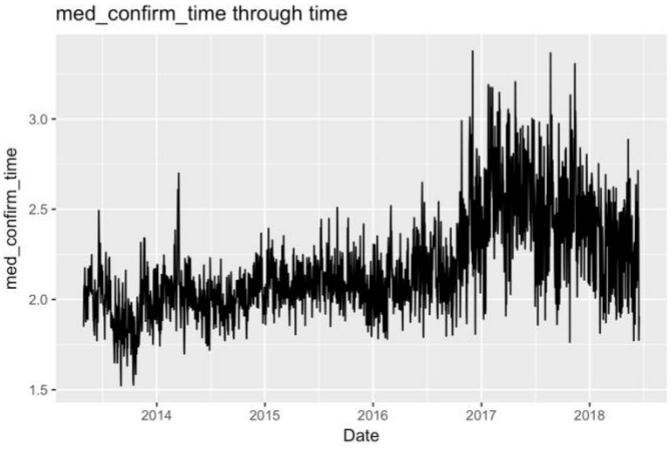

price through time
10
8
6
4
price
2014
2015
2016
2017
2018
Date

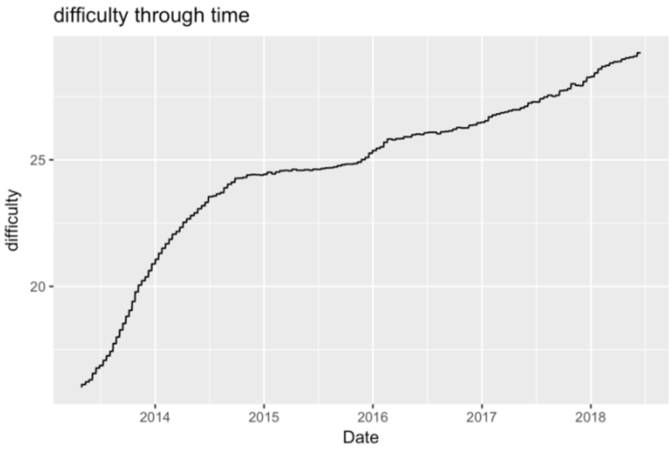

total_bitcoins through time
16.6
16.5
16.4
16.3
total_bitcoins
2014
2015
2016
2017
2018
Date

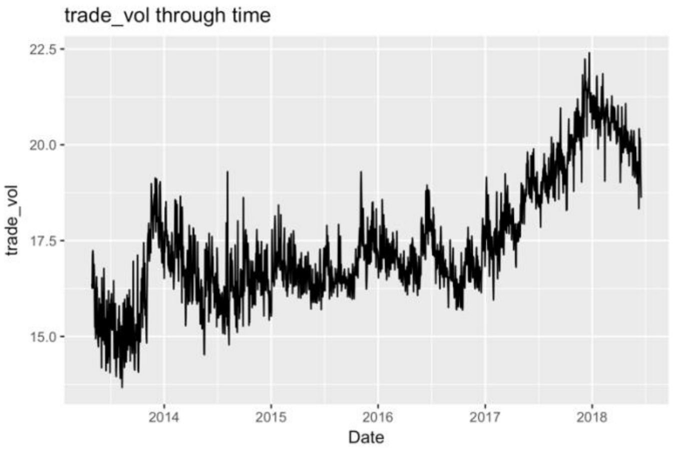

n_orphan through time
6
4
2
0
n_orphan
2014
2015
2016
2017
2018
Date
Date

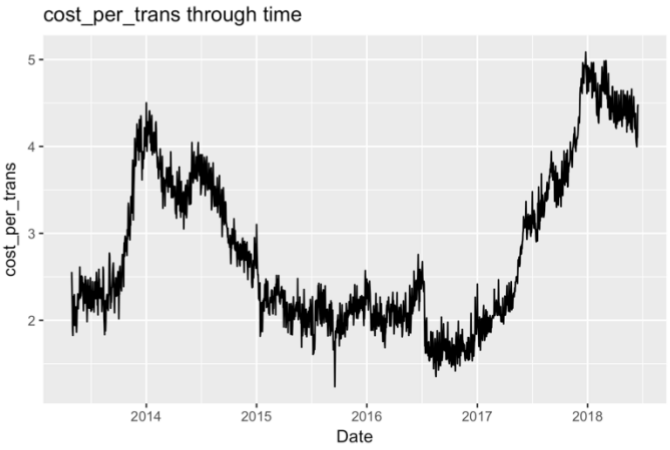

avg_b_s through time
0
-1
-2
avg_b_s
2014
2015
2016
2017
2018
Date

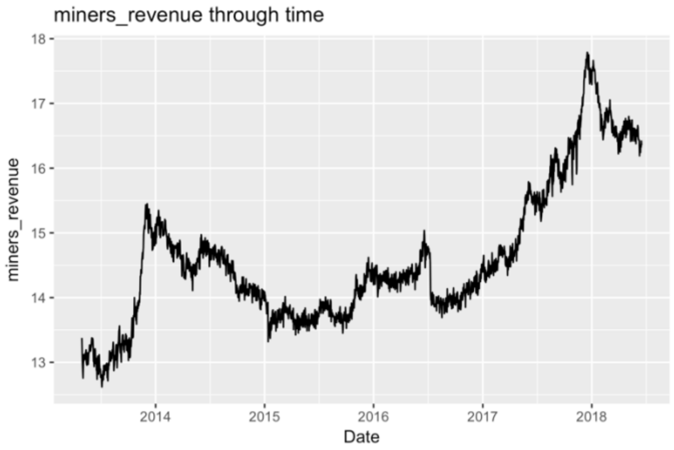

hash_rate through time
16
12
8
4
hash_rate
2014
2015
2016
2017
2018
Date

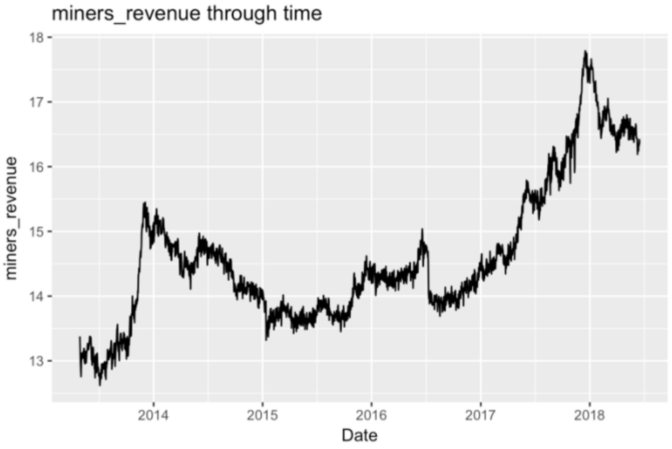

med_confirm_time through time
3.0
2.5
2.0
1.5
med_confirm_time
2014
2015
2016
2017
2018
Date

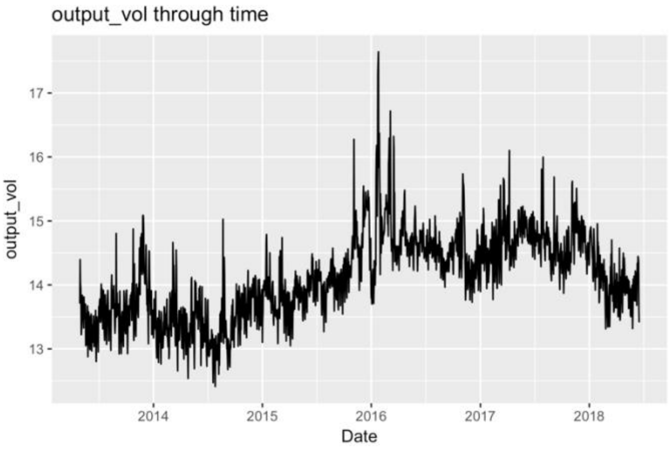

difficulty through time
difficulty
25
20
2014
2015
2016
2017
2018
Date

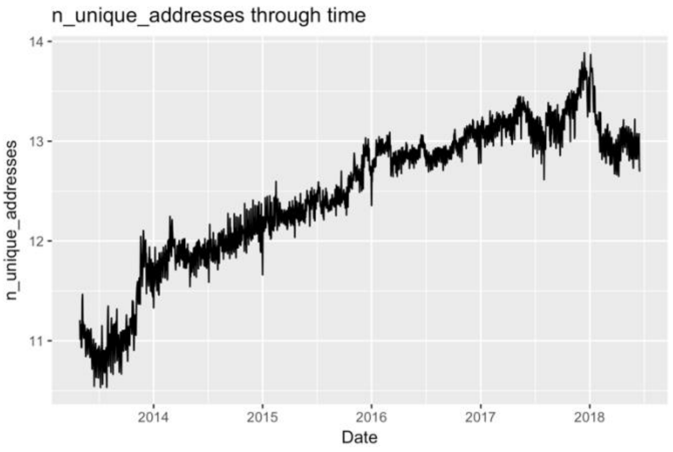

trade_vol through time
trade_vol
22.5
20.0
17.5
15.0
2014
2015
2016
2017
2018
Date

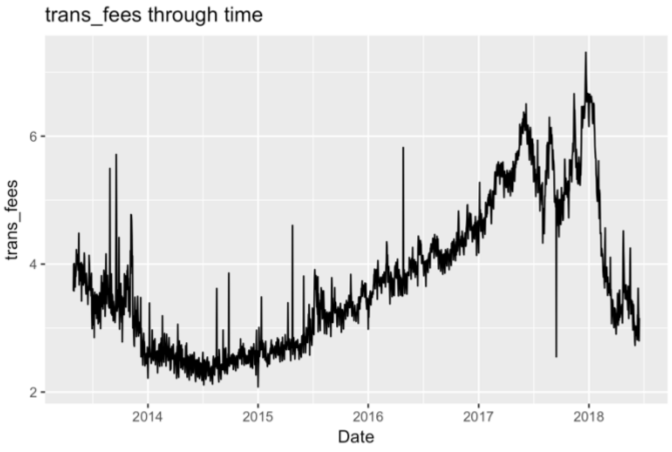

cost_per_trans through time
cost_per_trans
5
4
3
2
2014
2015
2016
2017
2018
Date

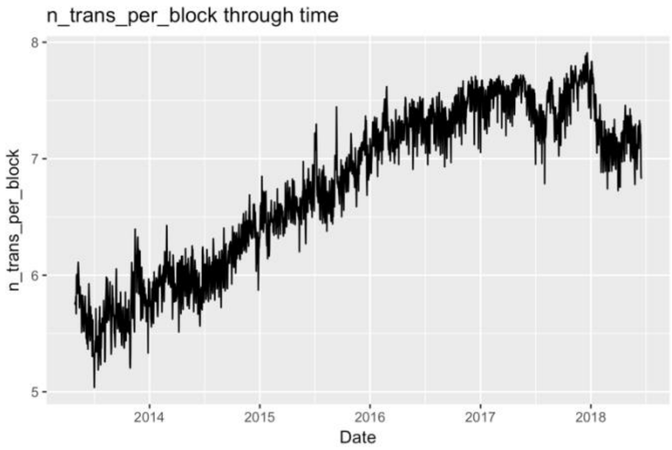

miners_revenue through time
miners_revenue
18
17
16
15
14
13
2014
2015
2016
2017
2018
Date

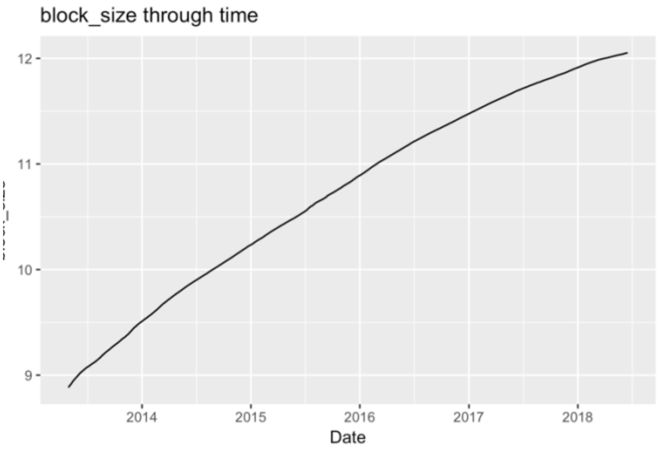

miners_revenue through time
miners_revenue
18
17
16
15
14
13
2014
2015
2016
2017
2018
Date

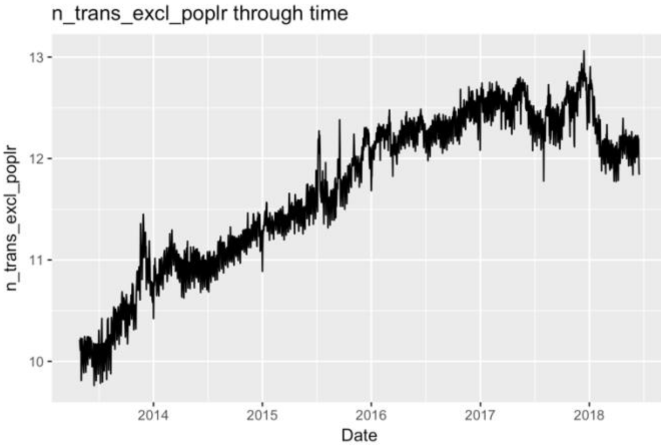

output_vol through time
output_vol
17
16
15
14
13
2014
2015
2016
2017
2018
Date

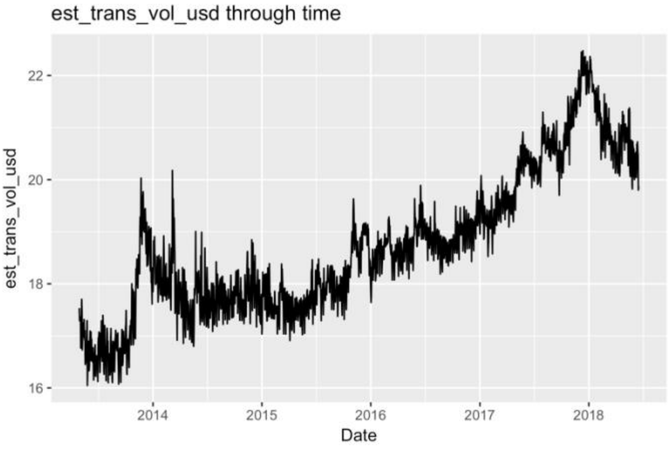

n_unique_addresses through time
n_unique_addresses
14
13
12
11
2014
2015
2016
2017
2018
Date

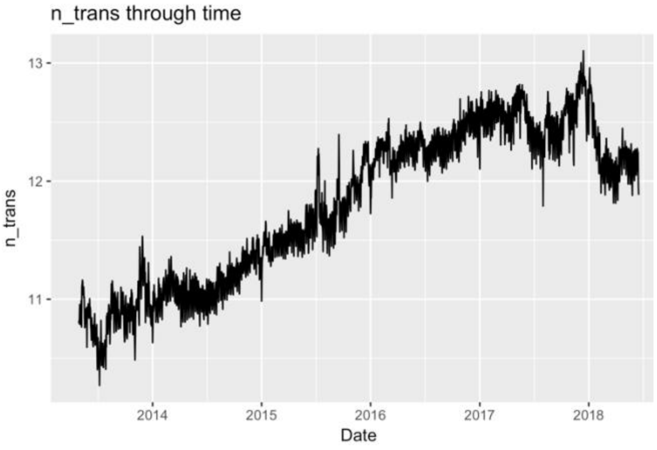

trans_fees through time
trans_fees
6
4
2
2014
2015
2016
2017
2018
Date

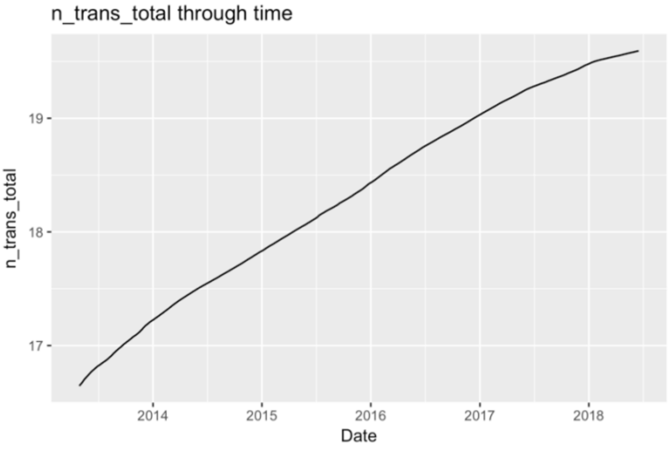

n_trans_per_block through time
n_trans_per_block
8
7
6
5
2014
2015
2016
2017
2018
Date

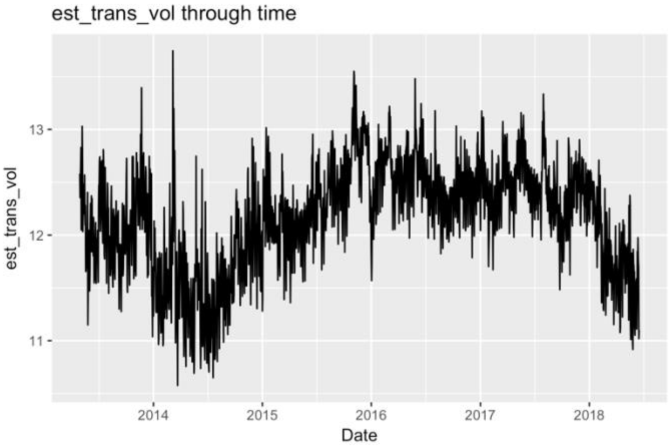

block_size through time
12
11
10
9
2014
2015
2016
2017
2018
Date

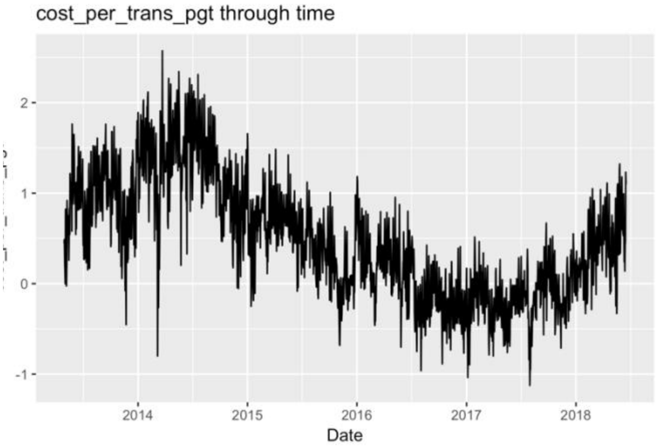

n_trans_excl_poplr through time
n_trans_excl_poplr
13
12
11
10
2014
2015
2016
2017
2018
Date


est_trans_vol_usd through time
est_trans_vol_usd
22
20
18
16
2014
2015
2016
2017
2018
Date

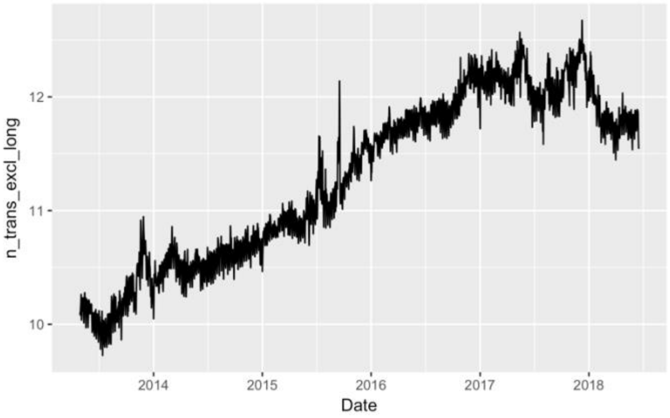

n_trans through time
n_trans
13
12
11
2014
2015
2016
2017
2018
Date


n_trans_total through time
n_trans_total
19
18
17
2014
2015
2016
2017
2018
Date

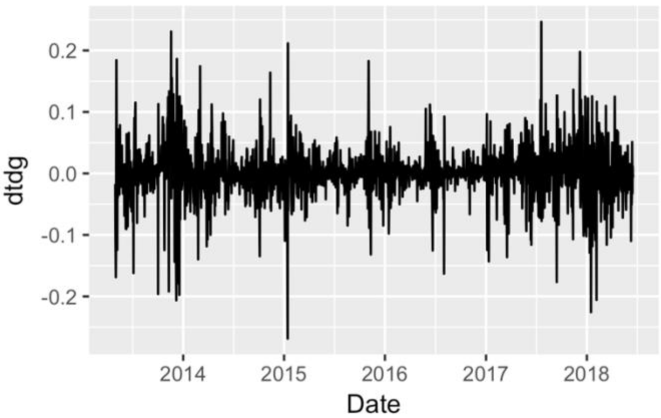

est_trans_vol through time
est_trans_vol
13
12
11
2014
2015
2016
2017
2018
Date


cost_per_trans_pgt through time
2
1
0
-1
2014
2015
2016
2017
2018
Date

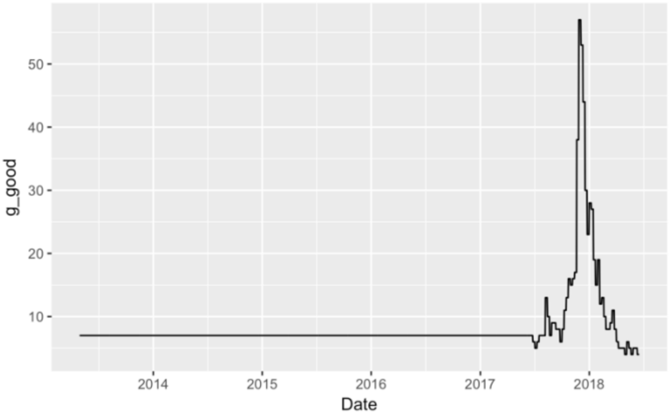
n_trans_excl_long through time



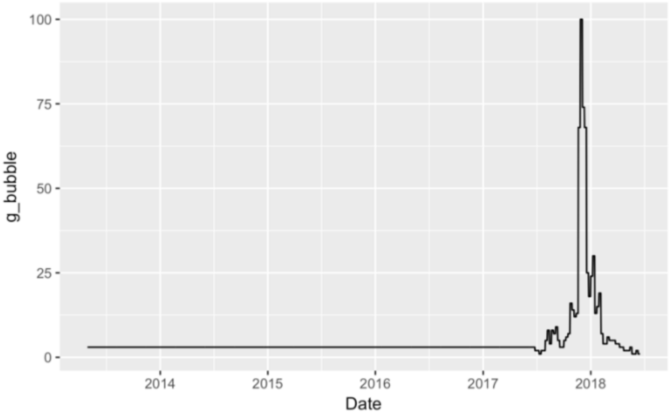
dtdg through time



g_good through time



g_bubble through time


g_bubble
Date