DEGREE FINAL PROJECT

USING
ARTIFICIAL INTELLIGENCE
TO PREDICT STOCK MARKET MOVEMENTS

Enric Roda Moya

Business and Technology

Supervisor: Vicente José Ivars

Date: 27th May 2022

Acknowledgements:

I would like to express my gratitude and appreciation to all the people who have been helping me during the development of this paper. Firstly, to my tutor Vicente José Ivars (Computer Architecture teacher) for his advice and assistance throughout the project. I would also like to thank my aunts Elena Moya (financial reporter in CQS) for her experienced point of view in the stock market, and Sofia Moya (Ph.D. in e-learning) for her help in the structure and format of the paper. A special appreciation to my partner Marie Wilkins, (credit risk in JP. Morgan) for helping me and for all the support. Also, my gratitude to Ramon Fabre (Marketing teacher) for the great conversations and exciting ideas.

Finally, I wish to thank my parents for their support and encouragement throughout my study.

Abstract:

Since the 1990s, research into financial investment in order to predict stock market prices and reduce risk has included Artificial Intelligence (AI). Especially in recent years, with the acceleration of technological development and the popularization of the personal computer. However, shortcomings have been identified in finding simple and efficient models. This project is contextualized on the first half of 2022, characterized by the rise in the inflation rate and a correction on the stock market. This situation creates huge instability in the market, and an increasing risk of stock market inversion. The aim of this study is to develop a forecasting stock market program using Python language. This paper is based on the two main stock market analysis: the technical and fundamental analysis. For the technical analysis we are going to use a machine learning (ML) algorithm based on portability, Binary Logistic Regression. For the fundamental analysis, we are going to study the financial situation of a company using financial ratios. The prototype has been tested on a sample of 17 companies during the last 4 months. The main conclusion of this study is that the program can beat the market.

Key words: Stock Market; Artificial Intelligence, Machine Learning, Investment, Financial Ratios.

Table of contents:

Table of figures:

Abbreviations:

ML	Machine Learning
AI	Artificial Intelligence
GPU	Graphics Processing Unit
CPU	Central Processing Unit
RAM	Random Access Memory
PC	Personal Computer
DB	Data Base
df	Data Frame
HDD	Hard Disk Drive

1. Introduction:

1.1 Introduction to invesment:

Nowadays, most people are aware of how important investments are for individuals, societies, companies and governments. We invest to obtain a financial return, but also to obtain or satisfy a need in education, security, entertainment, consumer goods, and many more fields. Investing is all about returns, either economic or personal. Investments are also crucial for private companies as capital allows them to grow, repay their debt with interest (generating profit for the lender) and improve their products and services for customers, therefore providing a source of profit and productivity.

Investing is as old as the human race. Prehistoric men already traded food, tools, or equipment with others to obtain profit in the future. This was the first type of investment. Since then, investments have evolved and adapted to circumstances and needs. For example, during the Roman Empire and in Ancient Greece, trading voyages around the Mediterranean became an extended practice. However, there were many risks during the trip, so entrepreneurs began to establish agreements with people back home who would help fund these maritime trade missions in exchange for some profit. The basic practice of giving investors a cut of a company's profit remains present today.

As outlined above, one of the most important elements of investing is the risk involved. Even if you invest in a Roman Empire cargo ship in the Mediterranean, or invest in the new restaurant in your neighbourhood, you are expecting future profits and are accepting the risk attached to the investment. Added to these risks there are many more problems which can make inversion complex.

In this project, we are going to create a program using complex algorithms, large amounts of stock price data, and financial ratios to reduce the risk of our inversions. To make it easier to use and understand, there is no need of financial or coding knowledge. The information will be displayed in a simple way, with the data generated automatically every day in an open twitter account, to make it open to everyone and without any need for installation.

1.2 Why develop an investment program?

1.2.1 Inflation:

During the last year of 2021, we saw all over the media how inflation has been increasing drastically, reaching the historical maximum since 1989. This year, 2022, does not seem to be getting better, or rather worse due to the Ukraine crisis and more macroeconomic factors. During this year we are going to suffer the impact of the expansionary fiscal policy of COVID-19, and at the same time the increasing price of raw materials and energy. During periods when inflation is rising, the loss of purchasing power is visible and remarkable. However, even during the economic prosperity periods where the inflation is really low, it has an impact in the long term.

There is not much we can do to reduce inflation, nevertheless as an owner of savings, investing is a great option to offset inflation in the long term.

To explain the impact of inflation, one example is seeing the differences between the real value of the savings and the same savings invested in the S&P500, one of the most used stock market index. To compare those two options, we used the Spanish annual salary between 2003 and 2021, the average saving, and the annual IPC (inflation rate). By the end of the period, we see astonishing differences between the two options, while just saving makes you lose 17.03% of the purchasing power of your savings, investing in S&P500 returns you 4.45 times the amount of savings.

In the graph bellow, we can appreciate how in the short term, neither the inflation or the S&P500 index have a notable impact. Even in 2008, during the financial crisis, the investing option has lower return. However, in the long term the investing line increases exponentially, due to the compound interest.

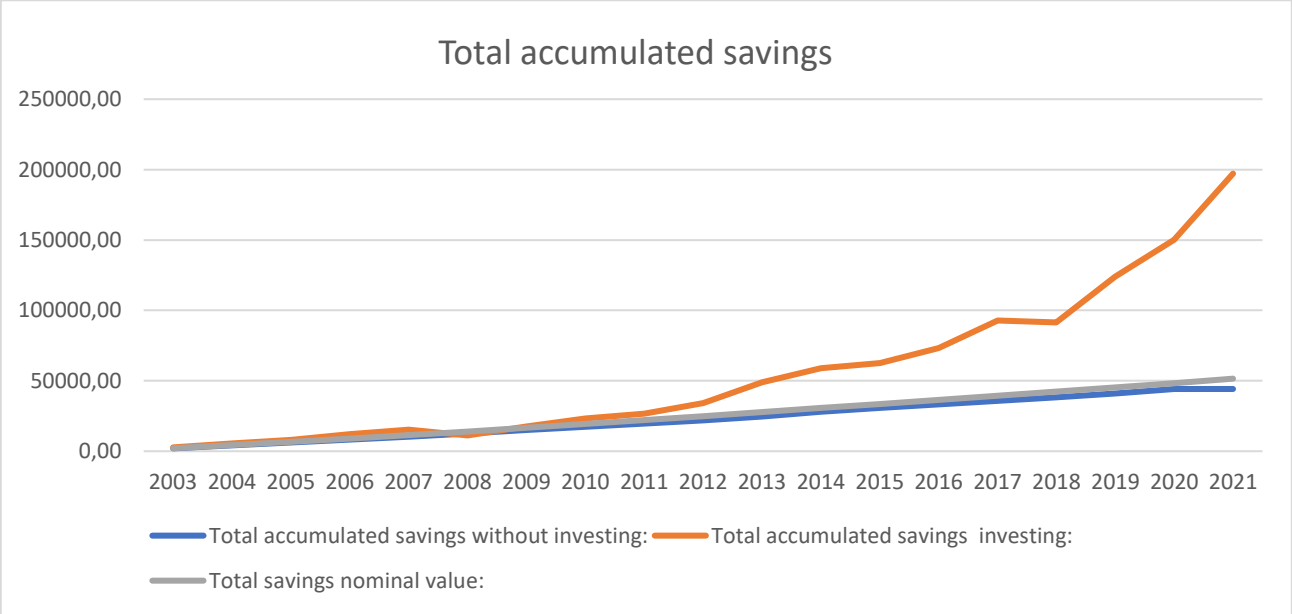

Figure 1 Inflation - S&P500 evolution (2003-2021)

1.2.2 Investor Irrationality:

We are often our own greatest enemy when it comes to investments. According to the Dalba, 2011 r studies the average investor consistently sees lower returns than the market. Meanwhile, the S&P 500 Index has an average of 6.06% per year. The average equity fund investor earned a market return of only 4.25%. Investor behaviour is illogical and often based on emotion. This does not lead to wise long-term investing decisions.

Therefore, new files of study have arisen to reduce the illogical behaviour. Behavioural Finance, a part of financial theory that combines investors' cognitive and emotional behaviour with economics, alleges that investors have computational, informational and rational limits that lead them to choose "good enough" options, rather than the best available choice.

This theory recognizes two types of biases: First, Cognitive biases are the errors in how we process or interpret the information; for example, when a person subconsciously gives more importance to a piece of information or does not contrast their conclusions. Secondly, "Emotional Biases", originate from impulse or irrationals feelings. These two behaviours cause the majority of investors to obtain lower results than expected, due to their irrationality and lack of consistency in their investments.(Dhaoui, 2017).

Combating these irrational behaviours can be complicated because it is human behaviour. However, experts say that simply being aware of them is a good way to mitigate their effects. Investment models, such as the one we are trying to develop, should acknowledge the existence of such investment biases.

1.2.3 Lack of knowledge:

A lack of financial understanding has significant impact on investments. Meanwhile, when someone who is buying a car, computer, or a house checks and studies the different options and analyses the future purchase, this does not always happen in the world of investments. This could be due to the inexperience, the inability to find the right information, or the lack of knowledge about finance, the company, the sector, or the market that one is going to invest in.

Investing is not easy and there is information overload: news, data, ratios, headlines, or even social media tips. While there is a lot of noise, knowledge is power, and one has to be able to select its information channels efficiently. Understanding and applying the information takes learning time, so not all investors know or use any metrics or information, making the risk of the investment much higher. This subject is explained by Alejandro Bernales (2021) on his document "Effects of Information Overload on Financial Markets" where he concludes by explaining that "Investors have limited capacity to process information and the effects of information overload on stock market".

1.2.4 Investment Professional:

There are many services to help investors, such as Investment Advisers, Financial Planners, and Brokers. Professional advice will recommend how to invest your money depending on your needs and they can also manage your portfolio. This service is useful and used by people with large amounts of capital, which they want to put to work.

But these services are not for everyone; in most cases they require a minimum amount of capital to contract their services and the cost is generally not worth it if you just want to invest a small amount. Therefore, small investors tend to end up making their investment without any professional help.

There are also some free services online that provide corporate or market financial information, but as outlined in the lack of knowledge and investor irrationality, the average investors do not know how to read and interpret the information, or do not know where to find it.

1.2.5 New methods:

During the last decade, technology has improved and new files such as Artificial Intelligence (AI) and Big Data have taken an important role in our society. These technologies allow companies to improve efficiency and productivity, improve the speed of business operations, and offer better customer service. Consequently, AI opens new business models and new methodologies of business, which have not been efficient or unprofitable until recently.

What is more, AI is applied in so many different fields, such as medicine, social media, the automotive, among others. Applying an AI system to a company improves productivity and generally improves a product or service for customers. The tools needed to create these systems are open source, which makes the cost of implementing those technologies not too high relative to the value they bring. These new methods open a new business model where data is a high-value asset, and the information generated, and the automatized systems are the base of the business. Many companies seem to be turning into technology companies.

Finance is no exception. Due to the massive amount of financial data that companies share online for the shareholders, there are vast amounts of information to use. Having the knowledge to create these AI systems and the data, we can apply our financial knowledge to create a system that can help us improve our investments.

1.2.6 Personal Experience:

One of the main reasons I chose this project was to invest in a rational way, and not intuitively. The second reason is the combination of my personal background, my field of study and my professional experience.

It was during the first COVID lockdown that I first started investing in the stock market, because I had some savings and I wanted to make some money. My first investment was on AMD, a semiconductor company with above-average growth, I became aware of it because I'm a hardware fan and I follow the news of this field. I learned that the expectations for this company were good. At the same time, I was investing in a few companies that someone had recommended me or that just sounded interesting. Of course, I was reading and trying to get information about those companies that I didn't know that well. But the main problem was that I didn't know how to analyze them well or which information had more value. By the end of the year, I made a positive return on the company I knew, but lost money on the one that I didn't know.

Consequently, I learned the lesson: So last year, I tried to do more exhaustive research before investing, including the checking some financial ratios. However, by the end I hadn't improved much. My hypothesis is that every time I wanted to invest in a company and I was reading, checking information or analysing the company I was trying to convince myself that was a good company and was the right moment to invest instead of taking a rational point of view.

Luckily for me, my poor financial results came at the same time that I started learning how to code in python. Quickly I focused on data analysis and ML, by doing online courses and reading forums. However, the real learning came last summer, when I got a summer internship in which I really started applying the knowledge. The internship was in a data science and risk department, was there when I really learned how to work with real-life data and prepare it to use in Machine Learning (ML) algorithms.

After the internship I kept working with those methods, creating my own projects about different issues. It was that moment when I had to decide the TFG's topic, so I thought this would be a perfect opportunity to create this project that really matches Business and Technology, my degree.

This Project is a great opportunity to apply my knowledge in both fields and create a program that would let me invest in a rational way. Also I have the opportunity to learn more in both fields, making it more exciting for me because these are the fields that I'm most interested in.

1.3 Objectives:

The main objective of this Project is to develop a program that predicts the stock market movements using AI, and to invest in a rational way, avoiding emotional bias and overloading information. To achieve the main objective, we must consider the following three sub-objectives.

- Study the trends and patrons on the stock market prices in order to gain insight and prevent the inversions with more risk.

- Recreate the decision-making process of an investment professional by using the most crucial financial ratios.

- Obtain a satisfactory result for our predictions, and to supply the program to the unexperienced investor, and reduce the risk of their inversions.

By meeting these 3 goals, we can help investors improve their investments by reducing the risk and emotional bias. These two characteristics are usually linked to unexperienced investors, which is our target audience. However, the professional or experienced investors can also use this program to obtain additional information to improve the decision-making process.

2. Theoretical background:

2.1 Stock market:

2.1.1 Definition stock market:

According to the leading global online trading broker IG Group (2021): "The stock market, or equity market, is a series of exchanges where shares in public companies are issued, bought and sold. Its role is to give private investors a way to own a stake in a listed company, while providing the companies themselves with capital to reinvest in their business". In other worlds, the stock market provides a source of capital for a company and for an investor.

For the company, the stock market provides access to the capital in the form of cash, and from the shareholders' perspective, the stock market provides a way to participate in a company's growth and also quickly convert shares into cash.

2.1.2 Analysis of the stock market:

Due to the uncertainty, volatility, and fluctuations of the stock market, investors try to mitigate risks by analysing companies and markets where they are planning to invest. There are two main methods to analyse the stock market: Technical and Fundamental - the difference lies in how to determine the value of the company:

- Technical analysis: Purely based on the price of the stock, technical analysis says that price is the one and only one factor to consider because it has all information to analyse a market or a company. These analysts use statistical methods, trend analysis, and the constant supervision of the market. This method is used more in short and mid-term investments because you are taking profit from market fluctuations.

- Fundamental analysis: This is based on the study of the intrinsic value of a company. In other words, it considers all the facts that affect the value of the company - macroeconomics, or microeconomics:

    - Macroeconomics or top-down: The status of the national or international economy, the status of the sector, analyses the competitors and all the external facts that can affect the operation of the company from an external point of view.

    - Microeconomics or bottom-up: Studies company-specific metrics such as price-to-earnings ratio, return on equity (ROE), cash flow, liquidity, revenues – among many others, which give us an accurate picture of the company.

  Following a thorough fundamental analysis, and depending on its results, one may or may not invest in a company. This method is more used in long-term investments as it is a deeper view that aims to invest, not only trade.

2.1.3 The Analysis used:

In this final project we are going to use both analyses to improve our predictions and at the same time have more supported variables in our model. Now we are going to see the advantages of each type of analysis for our model:

Technical analysis:

- **Data availability**: Getting real-life data of the stock price is easier and more accurate than having financial information from some companies.

- **Testing the outcome:** In the technical analysis, investment periods are short enough to prove some results. Proving or test the results of an ML model is an important part of any ML model.

- **Methodology**: Data analysis and forecasting are more linked to statistical and mathematical methods that are correlated with the methods used in the technical analysis

Fundamental analysis:

- **Solid base**: This method will help us to have a detailed analysis of the financial state of the company based on trusted metrics and ratios.

- **Detect overprice or under-price stocks**: We are going to detect if the price of the stock is overpriced or not, using this analysis.

- **Reduce risk:** With this method, we are able to detect if the companies have debt or liquidity problems, and so be aware of that risk and prevent it.

2.2 Tools Used:

2.2.1 Python:

The coding language used for this program is Python. This language is a high-level, interpreted programming language. The use of Python is so extended in so many files such as AI, Maths, Science, and much more files. Consequently, Python is the most used coding language nowadays.

What makes Python one of the best languages is that is so easy to learn and use compared to the other options such as C++, Java or R, due to the simple syntax. Furthermore, Python contains a huge number of libraries, that provide many functions which make coding much easier to write and understand. We are going to explain what is a library in the next point (2.2.2).

Python is also the most used language for ML and this is because it is so powerful when it comes to data analysis and for processing massive amounts of data. Also, an extraordinary selection of AI libraries is another primary reason why Python is the most mainstream programming language utilized for AI.

2.2.2 Python libraries:

2.2.2.1 What is a library?

Python libraries, also called "modules", are a collection of functions and methods which allow us to perform lots of actions without writing your own code. Libraries make Python Programming simpler and more convenient for the programmer. Python libraries play a very vital role in fields of Machine Learning, Data Science, Data Visualization, etc.

These modules must be installed in the Python environment to be able to use them. To install these libraries, the only need is to run the code: “pip install” and the module name in the Python file or the kernel. Installing and using these libraries is completely free, and all the information necessary for using them is available online in the documentation of the library.

2.2.2.2 Libraries used

All the modules have different functions and applications, so we are going to explain the main purpose of the modules used. The following libraries are the ones we used in our program:

Pandas: This is the main library to work with data in Python. Panda offers data structures and operations to manipulate numerical tables and time series. This library provides all the functions needed to undertake data analytics and data cleaning.

Numpy: With NumPy, we are able to create large multidimensional vectors and matrices, and it also provides a large collection of high-level mathematical functions to operate on.

Seaborn and **Matplotlib**: These two libraries are used in data visualization. Using this module, we can convert a table with data to graphs. There are more than 50 different kinds of graphs, so we can use the plot that best suits our needs.

Datetime: The datetime module supplies classes for manipulating dates and times.

Scikit-learn: Also known as *Sklearn*, is one of the most used machine learning libraries for Python. This module includes various machine learning algorithms, and all the functions needed to create a ML model.

Mlflow: Mlflow is a tracking library for saving and comparing the different ML models that we had created. It is very useful to compare and study the different metrics, variables or target variables in the different models.

Tweepy: Is a library provided by twitter that allows the user to interact, read, write, send direct messages with these social media, using an API. To use this module, you have to register to Twitter developers, and use your unique authentication.

Requests: This module allows you to send HTTP requests using Python. This library is done to read information from web pages.

2.3 Companies:

To limit the scope of the project, and also analyse some different industries and sectors, we have selected 17 companies. The selection of these companies was done through research of which company would be interesting to invest in, and also larger companies were chosen obtain the data easily. Only preliminary analysis was conducted to select the companies that we are going to use, as the program will be used to further narrow this down.

Those are the selected companies, and the main source of data is yahoo finance:

- Microsoft Corporation (MSFT)
- Alphabet Inc. (GOOG)
- Meta Platforms, Inc. (FB)
- Amazon.com, Inc. (AMZN)
- Alibaba Group Holding Limited (BABA)
- Oracle Corporation (ORCL)
- SAP SE (SAP)
- NVIDIA Corporation (NVDA)
- Nike (NKE)
- Taiwan Semiconductor Manufacturing Company Limited (TSM)
- Advanced Micro Devices, Inc. (AMD)
- Intel Corporation (INTC)
- HP Inc. (HPQ)
- Tesla, Inc. (TSLA)
- PayPal (PYPL)
- Walmart (WMT)
- Booking Holdings Inc (BKNG)

2.4 Machine Learning:

2.4.1 Description of how ML works:

There are thousands of definitions of what machine learning is on Google, which all explain that ML is a Branch of IA that applies algorithms and complex mathematics methods to find patrons, and is used for predictions. Sometimes these definitions are hard to interpret because those concepts are complex and intangible.

The definition below is based on an example that Javier Lopez, a teacher of maths in the University College of London shared with me:

When you have this equation (2 + 5 – 3 = x) you have the input and the mathematical equation to obtain the output (x).

When you have this other formula ((2 + 5)2 + x = 10) you have the output and the mathematic formula to obtain the input necessary for that output.

In ML you have a DB or a table with all the inputs and the outputs and the ML obtain the mathematic formula, based on the inputs and outputs.

2.4.2 Types of ML:

In this section we are going to explain the three types of ML according to Wakefield, 2022 moreover we will see the differences between them.

- **Supervised learning:**

The main goal of this type of ML is to learn a model from labelled data, which is going to allow us to make predictions about future or unseen data. The term ‘supervised’ refers to a set of samples where the desired output is already known. These types of algorithms analyze the labelled data or

training data and produces an inferred function, which can be used for mapping new examples. Supervised learning is the most important methodology in machine learning, and also the type that we are going to use.

Three types of supervised learning

1. **Classification**: In classification tasks, the machine learning program must draw a conclusion from observed values and determine which category new observations belong in.
2. **Regression**: In regression tasks, the machine learning program must estimate – and understand – the relationships among variables. Regression analysis focuses on one dependent variable and a series of other changing variables – making it particularly useful for prediction and forecasting.
3. **Forecasting**: Forecasting is the process of making predictions about the future based on the past and present data, and is commonly used to analyse trends.

- **Unsupervised learning:**

In unsupervised learning, we are working with unlabelled data or data of an unknown structure. With this type of ML, we use techniques to explore the structure of our data to extract meaningful information without the guidance of a known outcome variable. In contrast to supervised learning where data is tagged by an expert or a human, unsupervised methods exhibit self-organization that captures patterns as probability densities or a combination of neural feature preferences.

Two different types of unsupervised learning

1. **Clustering**: Clustering involves grouping sets of similar data (based on defined criteria). It's useful for segmenting data into several groups and performing analysis on each data set to find patterns.
2. **Dimension reduction**: Dimension reduction reduces the number of variables being considered to find the exact information required.

- **Reinforcement learning**:

The goal of reinforcement learning is to develop a system (agent) that improves its performance based on interactions with the environment. This method focuses on regimented learning processes, where a machine learning algorithm is provided with a set of actions, parameters and end values. By defining the rules, the machine learning algorithm then tries to explore different options and possibilities, monitoring and evaluating each result to determine which one is optimal. Reinforcement learning teaches the machine trial and error. It learns from past experiences and begins to adapt its approach in response to the situation to achieve the best possible result.

2.4.3 Logistic Regression:

2.4.3.1 Introduction to Logistic Regression

In our model, we had decided to use the Logistic Regression, which is a supervised learning algorithm. It is one of the most used aphorisms in statistical software and is used to understand the relationship between the dependent variable and one or more independent variables by estimating probabilities.

There are 3 different types of logistic regression: Binary logistic regression, Multinominal logistic regression and Ordinal logistic regression. We are going to be using the Binary Logistic Regression. This branch of Logistic Regression is used to work with binary dependent variables (a categorical variable that has two values such as "yes" and "no") rather than being continuous. For this reason, our program will try to predict if a stock is going to return more than a value, 'X', of the inversion or not.

On the appendix 1, we are going to explain the differences between the linear regression and the logistic regression and how binary logistic regression works.

2.4.4 Process to build a ML model:

There are many theories that divide the steps of how to create an ML model in different ways. However, in our case we decided to use and explain the one of the *datadrivenscience* team describe in their blog, since is the one it fits better in our model.

The document describes that there are three different parts to creating a ML model: the business value, proof of concept and production. Each of these parts contains different steps to complete the model. There are 7 steps in total. Next, we are going to explain them in detail:

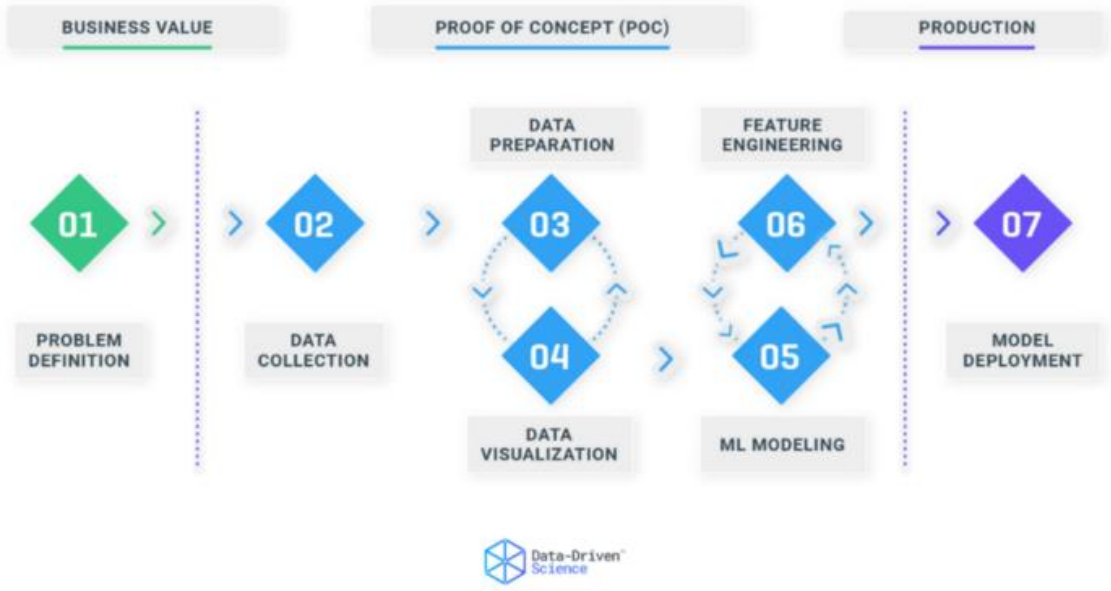

Figure 2 steps to create ML Model by Data-Driven

Business Value:

The business value is absolutely crucial because it is where the problem and the scope of the model is defined. It is also important to define how this model is going to bring benefits or improvements to the company. In short, are going to study how this model is going to bring value to the company.

1. Problem Definition: We are going to analyse the main functionality of the model, and define and understand the goals and the reason to apply the ML. Another important task in this step is to make sure that the data available will be useful to create the model.

Proof of Concept (POC):

Proof of Concepts (POC): In this second part, we are going to look at the main functionality of the model which is the most technical part of our framework. At the same time, we will define if it is viable putting the ML Model into production. In most cases, the first model which is completed is not going to be definitive, and so this part has to be completed again using different variables, algorithms or data. As we will see, in this part we are going to be searching for the most valuable model, and changing anything that improves the results of the model. Those are the 5 steps of this (POC) Part:

2. Data Collection: In this step, we are going to obtain the data needed for our model, it must be easy and fast to load because otherwise the model would run slowly, so it is important to store the data in an efficient way. Another important point is to verify the data we are going to work with is correct.

3. Data Preparation: The third stage is the most time-consuming and labour-intensive. Data Preparation can take more than 70% of the overall project timebecause it is very important to convert the raw data into high-quality data. This is the step where the data is manipulated, using data operations, fixing data errors, creating new data, and grouping the variables. The main goal of this step is to obtain the variables that perform better in the model.

4. Data Visualization: Visualization is an incredibly helpful tool to identify patterns and trends in data, which leads to clearer understanding and reveals important insights. In this step, we are going to analyse the performance of the data, try to find trends and maximize them. To find trends, we can use graphs or tables with information, these methods are going to give us a global image of the data. If the performance of the data is not convincing, it is important to make and step back and prepare again the data.

5. ML Modeling: In this stage of the process, we will apply the mathematical, computer science, and business knowledge to train the Machine Learning algorithm that will make predictions based on the provided data. It is a crucial step that will determine the quality and accuracy of future predictions.

6. Feature Engineering: In this step, we have to study the performance of the model, with a different set of features or variables. The goal of this step is to remove or add different features to improve the results of the model. This will allow us to delete the ambiguous and contradictory features and keep the ones that perform better.

Production:

The last part is testing the model in a real-life environment. It is important to integrate the model into the business in the most automated and accurate way, this will make the model more useful and easier to test. Lastly, it is also essential to test the machine learning model over time to improve it and check the results.

7. Model Deployment: The last stage is about putting a Machine Learning model into a production environment to make data-driven decisions in a more automated way. Robustness, compatibility, and scalability are important factors that should be tested and evaluated before deploying a model

2.5 Hardware:

ML projects involve the use of large amounts of data and complex algorithms that require powerful computation. Those complex algorithms are usually executed on servers, clusters, or Supercomputers, which have powerful computation power. Consequently, the predictions are executed in a short period of time, due to the optimum management of the hardware and software and the massive computation power which the servers have. However, it is possible to run an ML model on a personal computer, but there are drawbacks that we are going to see next.

Most of the ML models are programmed in a PC, because they are creating the model with a data sample instead of all the available data. However, it is not common to find a ML model in production running on a PC, as personal computer cannot process the same amount of data that Clusters or Supercomputers can. Once you test that your ML model works with a small and representative amount of data on a PC or business computer, you can start the implementation of the model with all the data on the clusters or the servers.

Computer power is the one of the mains problems of running a ML model on a PC. When you work on a PC, your computational power is limited by your CPU, while in a server or a cluster, you have more processing power and the operations can be distributed in an optimal way. Processing power makes a huge difference to time and resource optimization. The amount of computing time depends on the hardware you have, the operations you do, the amount of data involved, and the optimization of the code.

In our case, the hardware available is the one we had on our personal computer (The PC hardware) . In order to adapt the code that we wanted to create to the hardware available, we had to consider three main limitations:

1) **The amount of CPU and RAM used**: We want to limit the use of computer power during the code execution because we want to be able to use the computer while it is running the code. To do this, we adjusted the amount of CPU used from 50 % - 60 %, by the amount of the CPU threads used during the execution. To do this adjustment, it is necessary to use the python library Threads, which we are going to explain in the point (2.5.1.2) . Once the threads are assigned, these would define the necessary RAM, which is between 65%- 75%. Using all processor capacity would mean greater wear and tear of computer components, and the computer would be completely useless during execution time; however, the execution would be around 30% better, which is highly relevant taking this project forward.

2) **The amount of data we feed in our model**: The model processes 17 companies, in 5 different models,with 3 different periods of time. The amount of data that these codes process every day is more than 2 million rows and more than 30 variables; the total execution time is 160 minutes, or 2 hours and 40 minutes. Therefore, we had to limit the number of companies in the model - any company we add to the model is equal to 9 minutes 30 seconds of additional execution time. If we had more computation power, we would be able to fit more companies and variables to improve our predictions.

3) **Cybersecurity:** The other big problem comes with Cybersecurity. Every year the number of cyberattacks increases drastically, and so does the methodology and technology to prevent them. The hosting or server providers invest millions of euros to prevent these attacks. Consequently, it is safer to execute the code on their server than on a PC. To prevent the cyberattacks, we made our code not based on a server-client structure, so the clients cannot interact with our code or the hardware.

The PC hardware:

CPU : Ryzen 2700x
Base frequency **:** (4.2GHz)
Number of cores: 8
Number of threads: 16

RAM: **Team Group Delta**
Memory type: DDR4
Clock memory speed: 3200 MHz
Modules x size: 2 x 8 GB (16GB)

2.5.1 Improve the performance:

To combat the limitations of hardware, there are different tools that improve the performance of the hardware or the management of the hardware, and therefore the performance our code. On this project, we decided to use Multithreads to adjust the code to the hardware and Overclock to improve the performance of our hardware. Next, we are going to explain these tools and how we applied them.

2.5.1.1 Overclock:

Overclocking is the process of forcing your computer to run faster than it is intended to go. You can overclock both your CPU and graphics card, which can help you run advanced programs on a PC.

In our case, our code is executed on the CPU, so we overclocked the CPU. To overclock the CPU, it is necessary to improve the base frequency of our CPU by increasing the GHz of the CPU clock. To change the default velocity of the clock, you need to enter to the BIOS of the computer and chose the velocity which optimizes the performance of the CPU. It is important to track the CPU temperature and the energy consumed, because if these are too high it can wear out the components of the computer, or even break it. There are so many overclocking forums or web pages that can help you to find the right velocity for your CPU. In our case, our default base frequency was at 3.7 GHz, and after the overclock, it waws at 4.2 GHz. The increase in the base frequency has a visible improvement in the performance of the CPU.

2.5.1.2 Multithreads:

There is a default python module called Thread. This library does not have to be installed and is very convenient to improve the performance of our code. Thread allows you to run the code in different threads at the same time, in parallel.

To understand what this means, it is important to understand what a thread is. The CPU is composed of cores and threads. The threads are the virtual components or codes, which divide the physical core of a CPU into virtual multiple cores. The thread of the CPU allows the CPU to processes faster and reduce the Idle time significantly.

Working with Multithreads means that you can run different parts or functions of the code at the same time, using different threads. This method improves the execution time, and allows you to adjust the amount of computer power you want to use during the execution, as using more threads uses more computer power. The Multithreads cannot be used in all situations because the code executed in the different threads has to be independent to the others. For example, if you are executing function 1 in thread 1 to obtain the variable “A”, but you need the variable “A” in the function 2 executed in thread 2, this is not going to work because they are running at the same time so function 2 is not going to have the value of the variable “A”.

3. Methodology:

This research is based on mixed methodologies, both quantitative and qualitative. Qualitative research has been used for the tectorial background. In this section literature and documentary analysis has been the main methodology for data collection. Based on the results, our program has been developed and tested on a sample of 17 companies using quantitative methodologies such as correlations.

3.1 Development of the program:

This project is about an investing forecasting program which is going to help investors reduce their risk using complex algorithms and a massive computing of data. The program is focused on the stock market, and more specifically on companies with high market capitalization. To create the program, we used ML models to predict the future moves of the stock price, and a code that simulates the decision-making process of a professional investor, using financial ratios.

The program works in two steps: 1) the models itself uses the Binary Logistic Regression (technical analysis) and 2) the use of fundamental financial information (fundamental analysis). The first step is to create different models that will predict the return of the inversion during a specific period of time. More specifically, our model will predict whether the return on investment for a period will be higher or lower than a limit. For example, one period is 4 months, and the model will predict if the stock that we are analysing is going to return 15% in 4 months or not. The second analysis is going to get the companies that our first model predicts and study the financial situation, to get an accurate image of the financial state of the company and take a final decision.

Next, we are going to explain both steps in much more detail:

3.2 Technical analysis code

For the technical analysis, we are going to create different ML models that predict the price of the stock market using the Logistic Regression algorithm. To improve our forecasting results, we created 5 different models with similar parameters but with a few different variables or data fitted in the models. We are going to test the 5 different models to select the one with better performance. All five models study the evolution of the stock price of a company (technical analysis) and try to predict it in different periods of time. Therefore, we have 5 different models and 3 different periods of time which are: 2-week predictions, 1-month predictions and 4-month predictions.

Let's start explaining the data used on the technical analysis part.

3.2.1 Data:

Technical analysis is based in the stock price of the companies, and for this reason, the price of the stock is going to be our main raw data. The most used representation of the stock prices are the plots that represent the evolution of the stock price during that time, called stock charts. Those plots are done with tables of daily data of the stock price, and that is what we needed to create our technical analysis study.

To obtain these tables, we are going to use a pandas function called pandas_datareader, which is used to access remote data. This function reads the data from an online source to work with it in real life. In our case, the online source is yahoo finance (https://finance.yahoo.com/). Yahoo

finance is a service offered by Yahoo that provides financial information from international markets. Yahoo finance contains the stock data of more than 5,000 companies.

To access the data of the stock price we need to provide 3 variables:

- **Name:** In this variable, we need to introduce the acronym of the company that we want to obtain the data from. For example, if we want to study the stock price of Amazon, is necessary to introduce the acronym "AMZN" into the variable name.

- **Start:** In this variable we must upload the date of the day from which we want to obtain the data. It must be in "YYYY-MM-DD" format. To have the same accuracy in the models, we are going to fix the date for all the companies to have the same amount of data in each company.

- **Data_source:** The font of our information, in our case, Yahoo, will be always be the same.

Once we enter these parameters in the pandas_datareader function, it will return a data frame where every row contains the stock data of each laborable day. It's going to contain six columns which are: 'High', 'Low', 'Open', 'Close', 'Volume', and the index that is the date. For our study we are going to use the “Close” column as the final stock price which is the final value of the day. Also, we are going to use the Volume in one of the five models

Date	High	Low	Open	Close	Volume
2012-02-10	7.120000	6.960000	7.080000	7.050000	16473400
2012-02-13	7.370000	7.170000	7.240000	7.290000	19481700
2012-02-14	7.430000	7.240000	7.240000	7.320000	18756000
2012-02-15	7.500000	7.300000	7.380000	7.300000	15269400
2012-02-16	7.600000	7.270000	7.300000	7.590000	13140100
...	...	...	...	...	...
2022-04-22	91.459999	87.940002	90.029999	88.139999	75017700
2022-04-25	91.370003	88.610001	89.860001	90.690002	93481000
2022-04-26	90.120003	85.080002	89.739998	85.160004	89127400
2022-04-27	87.900002	84.019997	84.250000	84.910004	83125100
2022-04-28	90.580002	84.779999	86.669998	89.639999	91495400

2571 rows × 5 columns

Figure 3 Raw data obtained from Yahoo finance

The example above is going to be the data that yahoo finance will provide, for the company “AMD” and the start data 2012-02-10. This data is in a Data Frame format, which is a python table of data provided by pandas. In the next point we are going to explain all the variables created from this df, and the difference data fitted in each model.

3.2.2 Models and variables.

In this section we are going to explain the five different models that we had created. We are also going to explain the differences between the models and data fitted in each model. The explanation of the models are going to help us to understand how the program works.

3.2.2.1 Model Basic:

The model Basic was the first model completed and the main model from where all the other will be created off. We are going to explain this model in detail, and will just explain the differences for the other models.

Again, these models are based on technical analysis, so the model will be based exclusively on the stock price, as well as the price of the well-known NASDAQ index, which we will compare with our companies' performance.

Let's begin with the variables. There are 18 total variables on this model (17 independent variables + 1 dependent variable), separated into 4 different groups: 1) Variables Balance, 2) Comparative with NASDAQ, 3) Distance from maximum and minimum and 4) Target variable (dependent variable). All the 18 variables are calculated for each row of data, or what is the same for each day of the stock data. Once we have created all the variables, we will be able to find trends and analyse the values of variables that perform better and have a better return on the inversion.

1) Variables Balance:

This group contains 7 different variables, and the objective of this group is to fit the stock chart of the company in our model. To convert a graph into a numerical value, we create the balances which is the evolution of the company during a period of time. For this model we had created the balances of the following different periods of time: 3 years, 1.5 years, 7 months, 3 months, 3 weeks, 1 week, and 2 days. With the seven variables, we are going to be able to recreate the evolution of the stock of the last 3 years, with a numeric recreation equivalent to the stock chart.

<u>The equation for Balance variables:</u>

SP = Stock Price
Day = Day or row with daily data that contain the SP
Period = the amount of time of the variable

$$Balance_{\ periode} = \frac{\text{SP of Day}_n}{SP\ of\ (\text{Day}_n - period\ )}$$

Example of how to calculate Balance last 3 years of the 31/07/2020 of MSFT stock:

$SP_{=}$ Price stock MSFT 31/07/2020 = 205.01
$(Day_n$ – period) = (31/07/2020 – 3 years) = 31/07/2017
Price stock MSFT 31/07/2017= 72.68

$$\text{Balance}_{3\text{years}} = \frac{205.01}{72.68} = 2.82$$

The same process is used for the different periods, and for each row. So once we applied these variables for each day of the data, we are going to have the evolution of the stock price of the last 3 years in a mathematical way. On the appendix 2 "Code" we are going to see the python function to create all the 7 variables balance.

2) **Variables comparative with NASDAQ:**

These variables are based in the same system as the Variables Balance, but instead of creating the balance for the company, is going to do it for the NASDAQ Index. There are 3 different periods of time for these variables: 1.5 years, 3 months and one week. When the 3 NASDAQ balances are created, we are going to save the variables and create 3 extra variables. To create the other 3 variables, we are going to compare the results of the NASDAQ balances with the balances of the companies for the same periods, and return 1 if the balance of NASDAQ is lower than the one of the company or 0 if is higher. Therefore, if the company is growing more than NASDAQ, it will return 1, and 0 if not. This group of variables will help us understand how the company is behaving compared to the NASDAQ Index, and how the market performing.

3) Variable distance from maxim and minim:

The variable distance from maxim and minim are going to analyse how is stock price respect a local maxim or minim. As the other variables in this there are also different periods of time, in this case there are only two different periods 1 month and 4 months. To Analyse the distance from the minim and the maximum, it is necessary to obtain the data of the last month or the last 4 months, depend of the period. Then we find the max and the minim of the stock price of that period. Once you have the minim or the maxim you must divide the actual value of the stock price for the mini mor maxim of that period. As a result, we are going to know how much the variable fluctuated in that period, and the actual state of the stock price. See the coding function of this variables on Appendix 2 (Variable distance from maxim and minim).

The equation of distance variable:

SP = Stock Price
data_ period = Daily data of the SP during all the periods.

$$Min\_period\_distance = \frac{\min\{data\_period\}}{SP}$$

$$Max\_period\_distance = \frac{\max\{data\_period\}}{SP}$$

- **Target variable:**

The target variable or dependent variable is what we are going to use to predict our model. In our case, this is going to be the return of the investment of a specific period of time. Since we have a table with all the historic stock prices, we can find out the future price of a past date. The method is similar as the balance variables but instead of getting the past data we use the "future" data of that specific past date. Using this methodology, we can know the return of the inversion, of a determinate period.

The equation for Target variable:

SP = Stock Price
Day = Day or row with daily data that contain the SP
Period = the amount of time of the variable

$$Retun\_periode = \frac{SP\ of\ (\mathrm{Day}_n + \ period\ )}{\mathrm{SP\ of\ Day}_n}$$

As we said before, for each model there are 3 different periods of prediction. The different periods of prediction are 2 weeks, 1 month, or 4 months. These models try to predict the target variable, which is done by defining the desired return of the inversion for each period. Those are the 3 target variables for each period of predictions.

- The target variable for the 4 month prediction is that the return of the inversion is higher than 15 %.

- The target variable for the 1-month prediction is that the return of the inversion is higher than 7 %.

- The target variable for the 2 weeks prediction is that the return of the inversion is higher than 4%.

Once we have defined the target variables for each period, we are going to use the other seventeen variables to predict it.

Data used on basic model:

In this model, we are going to use historic data from 2012-02-10. However, the data fitted in the ML model is from 2015-02-10 until the actual date minus 4 months. Even the data we have is from 2012-02-10 until the actual day, we need to have 3 years of extra historical data to create the Balance 3 years. Also, as we need to create the target variable, which is the future return of the inversion, we are not going to have the last 4 month. We need to do these "cuts" of data, because we cannot fit it in our ML model a variable without a value, and so we need 3 extra years

from past to create the 3_years variable and 4 months in the "future" to be able to create the target variable. The same cut of data is going to be done in all the different models. To do this process we used the next code script.

3.2.2.2 Model 2000:

The model 2000 is using the same variables as the Basic model, but instead of using data from 2012-02-10, it is using the data from 1997-02-10, to fit the data from the year 2000.

This model was added to include the data of the financial crisis of 2008 in our model, because the data from 2012 has a growing trend that could be unrealistic.

3.2.2.3 Model 3 Var:

The model 3 var is the only model that contains volume data, which is the total number of shares that are traded. We are going to create two extra variables using the volumen: the "Volume_Grouped" and the "Impacto_Volumen". In addition, we add a third extra variable in the balance variables group, which is the 4 months time period. With this model, we consider if a stock is highly traded, because typically this reflects whether they are viewed as strong and healthy or not. Next, we are going to explain the two volumen variables of the 3var model.

- Impacto_Volumen:

On this variable we want to study the impact of the daily volume of trading on the Price of the stock. To do that, we need to correlate the daily variation of the stock price and the volumen, using a division. The result of that division is going to be the number of trades which result in a 1% increase in the stock price. The results are going to tell us if the trading trend is buying or selling, as fewer trades leading to a 1% increase in the stock price means more selling trades.

The equation is going to be the following:

$$\text{Impacto Volumen} = \frac{Daily\ Volume}{Daily\ Variation}$$

- **Volume_Grouped:**

This variable is easier to understand and interpret than the "Impacto_Volumen", we are just going to group the volume into 3 groups. The groups are going to be: if the trade volume is high (25 % above), average (25% - 75%) or low (the 25 bellows). This variable is going to give us information about how much the company is traded.

3.2.2.4 Model Double:

This model, similar to the other one, is based on the Basic Model. However, the data of the last year is duplicated, and the data of the last 2 months is duplicated again or quadrupled compared to the initial model.

This model is trying to give more importance to the later data instead of giving all the data the same importance.

3.2.2.5 Model BM:

The BM Model is different compared to the other models given that the data fitted in the Model is not just from the company that we are analysing, and instead the model is fitted with the data of the 40 biggest companies of NASDAQ. The data used is from the first day of data available for the company, so there is data from 1987 for some companies. This model contains much more data compared to the other model and makes the predictions different compared to the other models. The variables are the same as the once in the Model Basic. To reduce the computing time, the variables were created and saved on a table, so do not need to be created every time.

3.2.3 Understand the output Data Frame:

Once we had created all these variables, we are going to obtain a df with all the results of the operations. Each variable is saved on a column, and the result of each variable is showed with a numerical value. In the next picture we are going to see the result of the df with the variables crated for the company Microsoft:

Date	Balance_3years	Balance_1.5years	Balance_7months	Balance_3months	Balance_3weeks	Balance_1week	Balance_2days	Balance
2015-02-05	1.738247	1.187173	1.023932	1.133611	1.069643	0.985197	1.041739	-0.166664
2015-02-06	1.674405	1.143571	0.986325	1.060662	1.030357	0.949013	0.961667	-3.672789
2015-02-09	1.641323	1.023525	0.966838	1.041621	1.01	0.983652	0.94424	-1.975735
2015-02-10	1.600634	1.023525	0.966838	1.041621	1.01	0.983652	0.980243	0.0
2015-02-11	1.600634	1.023525	0.966838	1.041621	1.01	0.942667	1.0	0.0
...	...	...	...	...	...	...	...	...
2021-11-09	1.26902	1.576415	0.998651	1.039584	1.005703	0.99946	0.986677	1.507673
2021-11-10	1.276217	1.585356	0.954872	1.139535	1.010309	1.029014	1.020833	0.567116
2021-11-11	1.278361	1.505567	0.898155	1.029901	0.99212	0.988092	0.985957	-1.960267
2021-11-12	1.269663	1.491134	0.875545	1.027273	1.016301	0.963496	0.970999	-0.958637
2021-11-15	1.28125	1.421504	0.925907	1.016435	1.022415	1.000274	0.999452	0.912616

1708 rows × 21 columns

Figure 4 Data with all the variables created

As we can appreciate in the picture above, all the results are expressed in multiples, instead of as a percentage. This format is much easier to operate and reduces the execution time. At the same time, these variables are just to work with, so the main importance is to be fast to execute more than to understand easily. However, in the investing filed, it is more common to see that a company has grown a 73% in the last 3 years than the company, rather than having grown 1.73.

Once we understand the values of the df with all the variables, the next step is to prepare the data to find trends or maximize correlations to improve the performance of the Logistic Regression.

3.2.4. Extracting value from data:

3.2.4.1. Group the data:

On data preparation part of creating or obtaining new features to work with, we are going to obtain the maximum value from the data we already have. The numeric values of the df seen above contain more than 6 decimal , so trying to find trends with numeric values with that level of detail can be hard to process for the ML model. Even the quality of information that these variables support is very high and obtaining conclusions with a huge amount of data is difficult, due to the amount of detail in each value. The next step is therefore maximizing the value of the data we already have.

One of the most used technics for obtaining high-value variables for a ML model is to create groups for the different features or variables. This helps the ML algoristhm find trends in the data, and is also easier for data visualization. With this method, we group the variables according to their behaviour on a characteristic that we want to study. In our case, the behaviour we want to study is the correlation of the variables with the target variable in the different groups. Also, it is important that the groups have a similar distribution of data, as if one group has a small sample of the data, the results might not be accurate. This practice is highly extended on statistics, where for example in studies that want to correlate some behaviour with age, we see that the study groups the people into kids, young, adults, old, instead of using the years of each participant.

There are not any rules or mathematical formulae to create the groups, but the objective is to create groups that maximize the trend of the data. In order to do that, it is important to first create the different groups, and proceed to check the distribution of data on the groups to see the behaviour of the groups in our target variable. In our case, once we group the variables, we can see what the average return of the inversion for each group is, and also the average time when the return of the inversion is negative. It is recommended to keep creating groups until we find the parameters that maximize the trend of the performance of the target variable.

In the next example, we are going to see the difference between the data with the numerical result of the variable balance 3 months, and the same variable with the data grouped for the company Microsoft. To group this variable, we defined 3 groups. The first group is when the company has grown more than 12% in the last 3 months, the second is when in the company has grown between 12% and 4%, and the last one is for when the company has grown less than 4%. Once we have defined the groups, we can group all the results and see the behaviour of each group (Like in the Figure 7). In this case, we can see that when the company grow less during 3 months, there is

more return of the inversion and less negative return of the inversion. The key of this process is to try to find trends, such as the one on the Figure 7, for all the variables and for all the companies that we study.

Distribution of the 3 months balance:

1) Graph of 3 months balance and return of the inversion 4 months:

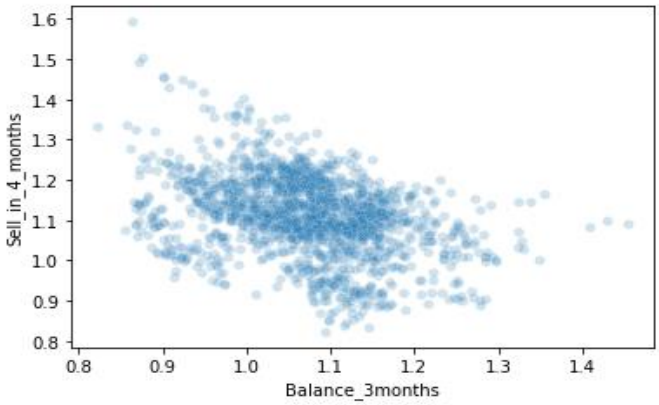

Figure 5 Graph with all the data. Axis y = Return in 4 months, Axis x = Evolution of the last 3 months

2) Table of the balance 3 months grouped:

Balance_3months_Grouped	Return_inversion_4_months	Amount_of_data	Losses_rate
1	1.154354	546	3.66%
2	1.111030	695	12.23%
3	1.066805	484	19.01%

Figure 6 Data grouped according to the balance 3 months

Looking at the two different pictures, we can appreciate that it is easier to appreciate the real trend in the grouped data. Even in the graph, we can detect a trend. We are not able to say that the differences of the two sides of the graph are that significant. Meanwhile, in the table we can appreciate that the group 1 (15.43% of return inversion), perform more than twice better that the 3 group (6.68%). These tables are going to help us to maximize the trends of the variables and play a key role in the predictions.

3.2.4.2 List with the values:

The process undertaken to make all the different groups of the variables is the following step Firstly, we needed parameters or numerical values that determine which group will be assigned to each value of the variable. In the example above, we saw that for the variable 3 months, the parameters were 12% and 4%, but this is just one variable and for the company Microsoft. The second step is once we have the two parameters that maximize the trends of each variable is to save the parameters on a list with all the parameters of a company. Next, we repeat this process for all the companies, since we have a list with all the parameters for each company. Once we have all the lists, we created a dictionary with all the lists of the companies. Finally, we are going to create a function for each variable. The function will receive the column or variable to be grouped and inside the function we are going to get the parameters of the dictionary using the company acronym and the position of the parameters needed. The function will apply the restrictions using the parameters, in order to group the variable. On appendix 2, we are going to see an example for how we group the balance_3months, according to the company. With this method, we save execution time and minimise the file size, as we do not have to create a function for each variable of all the companies.

3.2.5. Apply the Logistic Regression:

Now, we are going to have all the functions needed to convert the raw data to high-value data. The part which was completed was the data preparation and data visualization, where we had been working with the data until we obtained the data that we are going to fit in our model. The next step is the ML modelling, where we are going to fit the data in to our ML algorism to train and test the model.

The ML modelling starts by defining the target variable and the features. The features are going to be all 17 variables which have been grouped, and the target variable if the return was higher or lower than the limit determined according to the period on the Target Variable section. We are going to save the target variable on a pandas series, and the features on a bidimensional array. Next, we going to split the data in training and testing, by defining the percentage of data that we are going to train and test. In our case, we are going to train the 80% of the data and test the 20%. This means that we going to take an aleatory sample of 80% of the data to fit in to the Logistic Regression algorithms in order to study the data and find and accurate function. Once we enter the target variable, the features and we defined the percentage of train, the *sklearn* library will get the training features and training target variable and fit them into a function where we will apply the Logistic Regression algorism. That function is going to return to us a logistic regression function of our training data that we are going to use for the predictions. Finally, we are going to apply the function to our testing data and see how the model predicts the non-trained data, or data that the model never saw. We are going to use the accuracy score and more ratios that we are going to see next, to prove the performance of our model. This subject is explained on detail by Sebastian Raschka on his book *"Python Machine Learning".*

3.2.6 Improve the ML performance:

The next step in the technical analysis code is to select the variables that we are going to fit in the model. This part is called Feature Engineering, and is an important part of improving the performance of the ML model.

Selecting the best variables for our program can be hard, as some variables can perform better on a certain company, and worse on other ones. Besides, selecting the target variable that performs better is hard when you are making daily predictions in long term (4 months), because it takes time to prove the results. To obtain the best features, we used two different tools that help us to understand the value of each variable and their performance in the ML. The tools are NumPy correlations and the MLFlow library.

1) **Correlations:**

NumPy contains a function that returns a matrix of correlations with the correlation coefficient. A correlation coefficient is a statistical measure of the change in one variable defined by another variable. Simply speaking, you can say that the degree of intensity of the relationship between two variables is defined by the coefficient of the correlation. Therefore, we are going to apply this function and calculate the correlation between the 17 variables and our target variables. This will help us understand which variables have more value to predict our target variables. However, even if the variables have a high correlation, we don't know how they affect the model, so the next tool is going to help us to study the impact of each variable in our ML algorithm.

2) **MLflow:**

MLflow is going to help track the different executions of the logistic regression algorisms on the different companies. MLflow returns any metrics that we had selected from the Logistic regression and any piece of information that we want to add or study in the performance of the execution. The metrics used are the Accuracy test, which is the average of the correct predictions of the testing data, the average return of the inversion of the predictions, and the average correct predictions of the last year. It is important to check or use the metrics on the testing data, because it is data that the model has not trained. Next, we are going to explain the steps to obtain variables that perform better in our ML model using this tool.

Firstly, we have to integrate the MLflow to our code and determine which metric we want to check. The second step is to introduce or delete variables in our model to see the impact of each variable on the model. This process has to be repeated since we find the combination of variables that perform better in our models. We used this tool to define the periods of time of the 17 variables used in our program, so the reason for the periods of time in our variables is because they are the ones that have better prediction results.

The last step is determining the target variable. We tried different target variables, such as the minimum stock price of a period of time and the return of the inversion of a period of time. However, once we test the results of both dependent variables, we selected the return of the inversion. Then, once we select this target variable, we reaped the process of testing to find the best period of predictions. We found out that the best periods of targets variables are the 4 month and 2 week predictions, however ,we added the 1-month prediction as the results were acceptable too.

3.2.7 Flowchart and code:

In summary, the technical analysis code is based on the creation of a forecasting model based on a binary logistic regression algorithm using stock price data. We have explained all the steps that we used to create the deferments models above, and now we are going to outline the code in a flowchart.

The flowchart is a perfect option to represent the code and very helpful, to show or explain complex codes to a someone without an IT background. In our case, we are going to use the flowchart to explain the code flow of our model, more specifically the Basic model. All the code of the flowchart is going to be inside the "model_basic" function.

Firstly, it is essential to understand that the flowchart that we are going to explain is a python function. Therefore, we are going to have a block of code that, for a given input, executes a certain action and returns an output. In addition, the function we are going to see at the same time contains functions inside. Working with functions makes the code easier to understand and cleaner. For this reason, we are going to create a function for each model.

Point 1.
To start the code, it is necessary to receive the input of the "model_basic" function, which is a list (Comp_list) with the acronyms of the different companies that we are going to study.

Point 2.
Once we received the list, it starts with a "For" loop, that is going to execute the code for all the companies of the list.

Point 3.
The Next step is getting the name of the company on the "Comp_list", and to read and save the stock price data in a df. Then we obtain the data for the NASDAQ and save it too.

Points: 4,5,6,7.
For the next four points we are going to create the variables described on the point (3.2.2.1). For each of the types of variables there is a function. The input is the df with the stock price, and the output is the df with the new variables created.

Point 8.

In this step, we are going to delete the rows with the first 3 years in our df, as we are not going to be able to have the 3-year balance in these rows. We also delete the last 4 months rows, because that is unlabelled data that we cannot feed in our model.

Point 9.
Then we are going to apply the groups that have already been created, and convert our unclean df to a clean df ready to be fitted on the logistic regression algorism.

Point 10.
The next step is training the logistic regression algorism with our clean df.

Points: 11,12.
On the next two steps, we are going to get the data of the actual day and prepare it using the same method used before. Once we have the data of the last day prepared, we are going to fit it in to our trained logistic regression model. This is going to make the prediction.

Points: 13, 14, 15.
If the model predicts that the company is going to reach the target variable, it saves the company in a list (Saved_list), or else discards the company.

Point 16.
If there are no more companies in the list, return the "Saved_list", with the companies that the model has predicted.

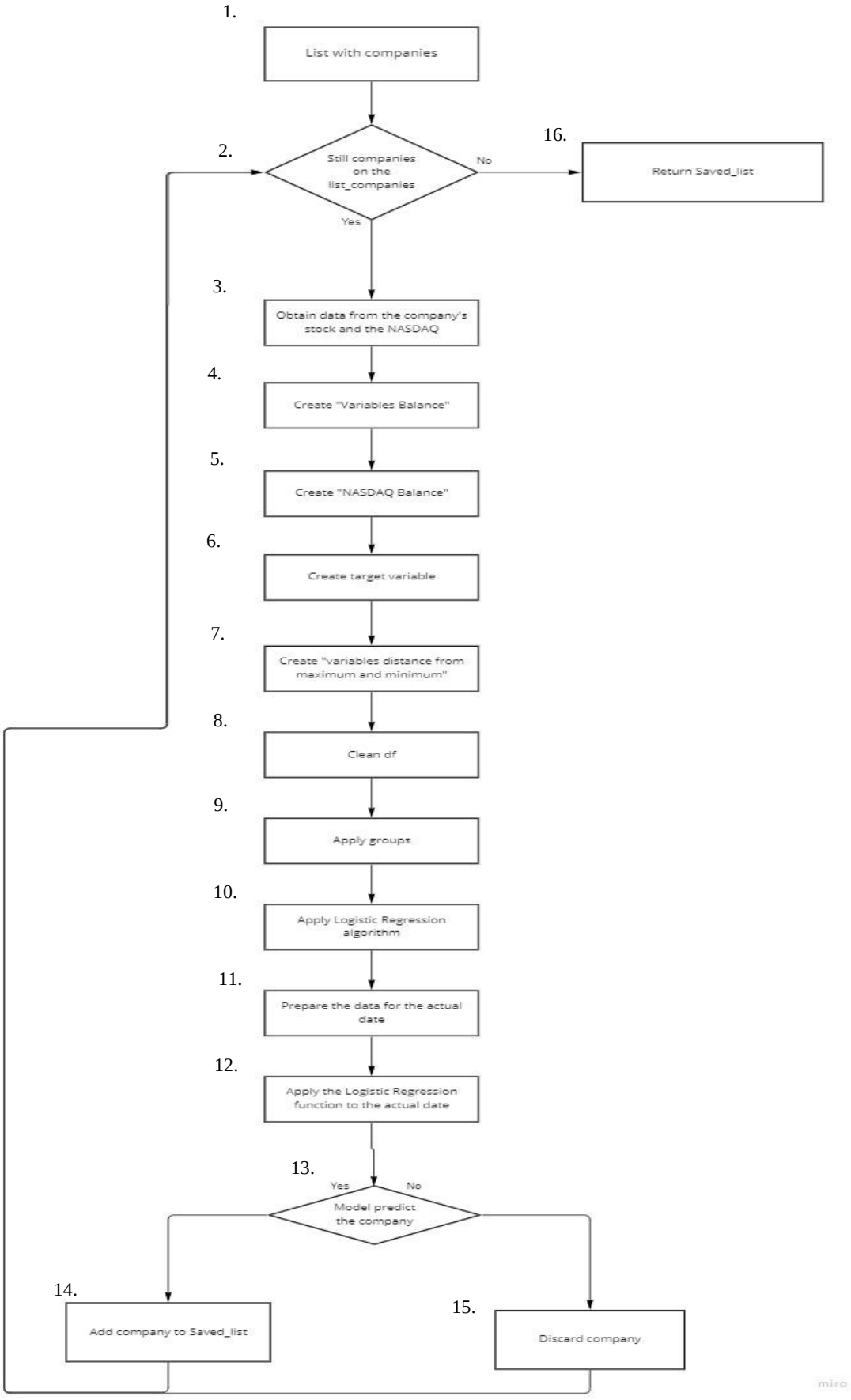

Figure 7 Flowchart with the Technical analysis code.

On the Appendix 2 we are going to have this flowchart on the coding format, and also the points 3 and 6 of this flowchart.

To finish with the technical analysis code, we are going to create a function like the one we had seen above for the 5 different models. Then, when we finally have the five functions created, we are going to be able to execute the different models, by just executing the function. In addition, as all models are independent, we are going to assign a thread CPU to each model, in order to execute the code in parallel.

In brief, the technical analysis method is going to return us more than fifteen daily predictions (5 models x 3 periods of time). The predictions of each model are going to be saved on a predicted list, and this list will be the input of the fundamental analyst code.

3.3. Fundamental Analysis:

Within the fundamental analysis, we are going to study and evaluate the financial situation of the companies using Financial Modeling Prep (FMP). FMP is a Financial API that provides financial statements, stock historical prices, stocks and Forex. The function of this part is to support the predictions of our ML models, and add some highly-valued ratios in our program. Using those ratios and financial information, we are going to analyse the companies that the ML model predicted, and discard the companies with a bad financial situation.

In the next section, we are going to explain how the FMP works, which data we are going to get, and how are we going to interpret them. Finally, we will use all the information obtained to make a final decision, and decide if is a good time to invest in a company or not.

3.3.1. Financial Modeling Prep:

The first step to understand what FMP is, is to understand what an API is and how it works. An Application Programming Interface (API) is a software intermediary that allows two applications to talk to each other. In particular, the FMP API connects our python code to their web page that contains all the financial information. Another API example is the twitter API which allows our code to connect to our twitter account.

In contrast to the twitter API ("tweepy"), the FMP API does not have a python library to interact with, so the method to used to read the data is through the request python library. Using this library and the API key, that FMP is going to provide us once we sign up on their webpage, we will be able to obtain the data.

3.3.2. Data:

FMP contains more than five hundred financial metrics, ratios, and financial information for more than 40.000 companies. The metrics are grouped on different pages depending on the type of information that they contain, for example: Financial Ratios, Financial Statements, Price Target and more. However, in this project, we are only going to work with 19 ratios from the Financial Ratios page.

To obtain the data for the Financial Ratios page, we are going to use the following python code:

```
Balance_Sheet = requests.get("https://financialmodelingprep.com/api/v3/ratios-ttm/" + company + "?apikey=" + API_Key )
Balance_Sheet = Balance_Sheet.json()
```

Figure 8 Coding script to access to the JSON file.

Where:
The variable company is the acronym of the company we want to get the data from, and the "API_Key" variable is the key received when we signed up.

This code is going to return a JSON file. JSON is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute-value pairs and arrays. On python, JSON files are similar to python dictionaries, so for each key there is one or more value. In our case, each key is the name of the financial ratio, and the value is the numeric value of the financial ratio.

The JSON file open contains fifty-eight different ratios, but we only going to use nineteen (See example of JSON file for Intel). The reason we have selected these variables is because these are the ones we had learned in the Finance subject, or because we believe these are essential when it comes to stock market investing. In the next point, we are going to explain the nineteen variables.

3.3.3. Financial ratios:

The use of financial ratios is one of the most extended technics of the fundamental analysis. Studying and examining the financial ratios allow to understand the financial situation of a company. There are many financial ratios, and each one determines the aspect of the business which the ratio measures, for example the Cash Ratio determine the cash liquidity in the short term.

The results of the financial ratios can be interpreted according to the state of the company, the sector where the company operates or even the market situation. Therefore, even if two companies have similar results for a certain ratio, this does not mean that they are in a similar situation. However, there are some ranges in all the ratios that define whether the company is doing well in that specific financial statement or not. We are going to use these ranges to score the different

ratios with values from 0 to 2, and try to qualify and determine the general financial situation of the companies. The higher the result, the better the performance of the ratios. To define the scores of the ratios, we used the well-known webpage Investopedia. This is the article used to score the *Debt-to-Equity Ratio Ratio* write by Katrina Munichiello (2022). Finally, we are going to group the different ratios according to these four categories: 1. Stock price, 2 Liquidity, 3. Profitability and return and 4. Leverage.

1) Stock price:

The stock price group is going to analyse if the stock is overpriced or not using the following two ratios: Price-to-earnings and Price/Earnings-to-Growth. These two ratios determine approximately if a stock is overpriced, or if it is going to be overpriced in future.

- **Price-to-earnings ratio (P/E ratio):**

The Price Earnings Ratio (P/E Ratio) is the relationship between a company's stock price and earnings per share (EPS). It is a popular ratio that gives investors a better sense of the value of the company.

$$PE\ Ratio = \frac{Market\ value\ per\ share}{Earnings\ per\ share}$$

For this ratio, we are going to score zero when the P/E ratio is higher than 60 or lower than 5. If the P/E ratio is between 30 and 60 or 5 and 10 the score will be one. Finally, if the P/E ratio is between 10 and 30 the score is going to be two.

- **Price/Earnings-to-Growth (PEG):**

PEG is a stock's price-to-earnings (P/E) ratio divided by the growth rate of its earnings for a specified period. This ratio compares the actual value of the stock, with the future growing of the company. So as bigger the future EPS expectation, the lower the PEG ratio, and is therefore the better option to buy.

$$PEG\ Ratio = \frac{\frac{Price}{EPS}}{EPS\ Growth}$$

On the PEG ratio, the score will be two if the ratio is lower than 0.75, one if it is between 0.75 and 1.25, and zero if it is higher than 1.25.

2) Liquidity:

Liquidity ratios measure a company's ability to pay debt obligations and its margin of safety through the calculation of metrics, including the current ratio, quick ratio, and cash ratio.

- **Current Ratio:**

The current ratio measures a company's ability to pay current or short-term liabilities (debts and payables) with its current or short-term assets, such as cash, inventory, and receivables. The result of this ratio indicates how many current assets there are for one unit of current liabilities.

$$Current\ Ratio = \frac{Current\ asset}{Current\ liabilities}$$

In this case, when the result is higher than 1.75, the score is going to be two, if it is between 1.75 and 1, it is going to be one and if it is lower than 1, the score will be zero. This is because the higher the Current Ratio, the better the company's capacity is to pay the short-term liabilities, and the better the score.

- **Quick Ratio:**

The quick ratio measures the dollar amount of liquid assets available against the dollar amount of current liabilities of a company. Liquid assets are those current assets that can be quickly converted into cash, while current liabilities are a company's debts or obligations that are due to be paid to creditors within one year.

CE = Cash & equivalents
MS = Marketable securities
AR = Accounts receivable
CL = Current Liabilities

$$Quick\ Ratio = \frac{CE + MS + AR}{CL}$$

For the results of the quick ratio, if the ratio is higher than 0.95, the score will be two, for the ones between 0.95 and 0.7 the score is going to be one, and if it is lower than 0.7, the score is zero. Since the higher the quick ratio, the better short-term liquidity of the company, and so the score will be higher.

- **Cash Ratio:**

The cash ratio is a measurement of a company's liquidity, specifically the ratio of a company's total cash and cash equivalents, to its current liabilities. The cash ratio determines the capacity of paying the short-term liabilities using just cash or cash equivalents.

$$Cash\ Ratio = \frac{Cash\ or\ Cash\ equivalents}{Current\ liabilities}$$

The scores for these ratios are: two if the cash ratio is higher than 0.4, one if the ratio is between 0.4 and 0.15 or zero if it is lower than 0.15. Similar to the Current and Quick ratios, the higher the result, the more capacity the company has to pay the current liabilities.

3) Profitability and return:

Profitability ratios are a class of financial metrics that are used to assess a business' ability to generate earnings relative to its revenue, operating costs, balance sheet assets, or shareholders' equity over time. Meanwhile, the Return ratios offer several different ways to examine how well a company generates a return for its shareholders.

- **Return On Assets (ROA):**

ROA indicates how profitable a company is in relation to its total assets. Corporate management, analysts, and investors can use ROA to determine how efficiently a company uses its assets to generate a profit. The results of the ROA indicate how much net income is generated by one dollar of assets.

$$ROA = \frac{\text{Net Income}}{\text{Total Assets}}$$

If the ROA is bigger than 0.2, this means it has a great return, so the score is going to be two. If the ROA is between 0.2 and 0.075 the score is going to be one, and if it is lower than 0.075 the ROA score will be zero.

- **Return On Equity (ROE):**

ROE is a measure of financial performance calculated by dividing net income by shareholders' equity. ROE is considered a gauge of a corporation's profitability, and how efficient it is in generating profits.

$$ROE = \frac{\text{Net Income}}{\text{Avg. Shareholders Equity}}$$

When the ROE is bigger than 0.5, this means that the company has great result so the score is going to be two. If the ROE is between 0.5 and 0.2 the score is going to be one and for the rest the score will be zero.

- **Return On Capital Employed (ROCE):**

ROCE can be used to assess a company's profitability and capital efficiency. In other words, this ratio can help to understand how well a company is generating profits from its capital as it is put to use.

$$ROCE = \frac{\text{EBIT}}{\text{Capital Employed}}$$

If the result of ROCE is bigger than 0.2, the score is going to be two. The score is one if the ROCE is between 0.2 and 0.075, and when the ROCE is lower than 0.075, the score is zero.

- **Gross Profit Margin:**

Gross profit margin is a metric analysts use to assess a company's financial health by calculating the amount of money left over from product sales after subtracting the cost of goods sold (COGS). Gross profit margin is often shown as the gross profit as a percentage of net sales. The result of this ratios is the gross profit for a one dollar of net sales.

$$Gross\ Profit\ Margin = \frac{Net\ Sales - COGS}{Net\ Sales}$$

For this ratio, the score will be two if the gross profit margin is higher than 0.65, one if it is between 0.65 and 0.3, and zero if it is lower than 0.3. As the higher ratio means more profit margin from the sales, this means that the productivity is better too.

- **Net Profit Margin:**

The net profit margin measures how much net income or profit is generated as a percentage of revenue. Also, it illustrates how much of each dollar in revenue collected by a company translates into profit.

$$Net\ Profit\ Margin = \frac{Net\ income}{Revenue}$$

For the net profit margin ratio, the score will be two if the ratio is higher than 0.25, one if it is between 0.25 and 0.1, and zero if it is lower than 0.1. As the higher ratio means more net income per unit of revenue, this means that the company is more productive.

- **Effective Tax Rate:**

The effective tax rate is the percent of the income that a corporation pays in taxes. In other words, the effective tax rate is the overall tax rate paid by the company on its earned income.

$$ETR = \frac{Total\ Tax}{Earnings\ Before\ Taxes}$$

The scores for this ratio is two if the effective tax rate is lower than 0.15, one if the ratio is between 0.15 and 0.6 or zero if it is higher than 0.6. For this ratio, the lower result is better, due to the company having fewer taxes to pay.

4) Leverage:

A leverage ratio is any kind of financial ratio that indicates the level of debt incurred by a business entity against several other accounts on its balance sheet, income statement, or cash flow statement. These ratios provide an indication of how the company's assets and business operations are financed (using debt or equity).

- **Debt Ratio:**

This financial ratio measures the extent of a company's leverage. The debt ratio is defined as the ratio of total debt to total assets, expressed as a decimal or percentage. The result of the ratio indicates how much debt a company has per dollar of assets.

$$Debt\ ratio = \frac{Total\ debt}{Total\ assets}$$

For this ratio, we are going to use different ranges for each score. When the debt ratio is between 0.6 and 0.3, the score is going to be two. If the result of the ratio is between 0.6 and 0.75 or between 0.15 and 0.3 the score will be one, and for the rest, the score is going to be zero. While a low debt ratio suggests greater creditworthiness, there is also risk associated with a company carrying too little debt, which is the reason of our classification.

- **Debt Equity Ratio:**

The debt-to-equity ratio measures a company's total debt relative to the amount originally invested by the owners, and the earnings that have been retained over time. It is used to evaluate a company's financial leverage. The interpretation of the result is the amount of liabilities for one dollar of shareholders equity.

$$Debt\ Equity = \frac{Total\ Liabilities}{Total\ Shareholders'Equity}$$

The score given to the debt equity ratio is going to be two if the result is lower than 0.75, one if it is between 1.45 and 0.75 and zero if it is more than 1.45.

- **Long Term Debt To Capitalization:**

The long-term debt to capitalization ratio is a solvency measure that shows the degree of financial leverage a firm takes on. It calculates the proportion of long-term debt a company uses to finance its assets, relative to the amount of equity used for the same purpose.

$$Long\ Term\ Debt\ To\ Capitalization = \frac{Debt}{Debt + Total\ Shareholders'Equity}$$

For this ratio, the score will be two if the ratio is lower than 0.15, one if it is between 0.6 and 0.15, and zero if it is higher than 0.6. As higher the ratio means more long-term debt.

- **Total Debt To Capitalization:**

The total debt to capitalization ratio is a measure that shows the proportion of debt a company uses to finance its assets, relative to the amount of equity used for the same purpose. So, this ratio measures the total amount of outstanding company debt as a percentage of the firm's total capitalization.

SD = short-term debt
LTD = long-term debt
SE = shareholders' equity

$$Total\ Debt\ to\ Capitalization = \frac{SD + LTD}{SD + LTD + SE}$$

In this case, when the result is lower than 0.15, the score is going to be two, if it is between 0.25 and 0.6, it is going to be one and if it is higher than 0.6, the score will be zero. A higher ratio means that a company is more highly leveraged, which carries a higher risk of insolvency.

- **Interest Coverage:**

The interest coverage ratio is a debt and profitability ratio used to determine how easily a company can pay interest on its outstanding debt. This ratio is usually used to determine a company's riskiness relative to its current debt or for future borrowing.

EBIT = Earnings before interest and taxes.

$$Interest\ Coverage\ Ratio = \frac{EBIT}{Interest\ Expense}$$

For the interest coverage, the score will be two if the ratio is higher than 10, one if it is between 10 and 1.4, zero if it is between 1.4 and 0 and negative one if it is lower than 0. In this ratio we added a negative value, due to it being more risky to invest in a company with an interest coverage lower than zero.

- **Cash Flow To Debt Ratio:**

The cash flow-to-debt ratio is the ratio of a company's cash flow from operations to its total debt. This ratio is a type of coverage ratio and can be used to determine how long it would take a company to repay its debt if it devoted all of its cash flow to debt repayment.

$$Cash\ Flow\ to\ Debt = \frac{Cash\ Flow\ from\ Operations}{Total\ Debt}$$

The score given to the cash flow-to-debt ratio is going to be two if the result is higher than 1, the score will be one if the result is between 1 and 0.5, and zero if the ratio is less than 0.5. So the higher the ratio, the more operations needed to pay the debt.

- **Company Equity Multiplier:**

The equity multiplier is a risk indicator that measures the portion of a company's assets that is financed by stockholder's equity rather than by debt. This ratio also used to indicate the level of debt financing that a firm has used to acquire assets and maintain operations.

$$Equity\ Multiplier = \frac{Total\ Assets}{Total\ Shareholder's\ Equity}$$

In this case, when the result is higher than 0.5, the score is going to be two, if it is between 0.5 and 0.15, it is going to be one and if it is lower than 0.5, the score will be zero. A low equity multiplier means that the company has less reliance on debt.

3.3.4. Evaluating the companies:

After defining the sores for all of the ratios, we are going to create a function for each group of ratios, so there are going to be 4 functions. Each function is going to contain the extraction of the respective ratios of the JSON file and the qualification of the ratios. The input of the function is going to be the JSON file, meanwhile the output is going to be the average score of the ratios that the group contains. That result is going to give us an approximate image of the financial statement which the ratio measures. For example, if the output of the liquidity function is 1.66 ([2 Current Ratio + 2 Quick Ratio + 1 Cash Ratio] / 3 = 1.66), we are going to know that company does not have any liquidity issues.

Finally, once we have the average of the groups, we are going to add them to obtain "the final result" value. The "final result" is going to give us an approximate image of the financial state of the company. This result is going to be a value between zero and eight, and the higher the value, the better the financial situation of the company is. Using "the final result", we are going to decide if the company we are analysing is a good recommendation or not. To make that decision, we are going to discard the companies with a score lower than five, and the highest ones are going to be added on a final list, recommendations. The recommendation list is going to be the list with the companies that our ML model predicts, and that our fundamental analysis proves that the financial situation is acceptable. This is the list that is going to be displayed on the twitter account.

The fundamental analysis method gives our program a higher level of credibility. This is because it is giving our program the capacity to analyse and study the financial situation of the company using trusted and proved methods such as the study of the financial ratios. With this part, we use metrics and methods used by experts to simulate their decision making.

3.3.5. Flowchart and code:

Similar to the technical analysis code, the fundamental analysis code is going to work with a list of the acronyms of the companies as the input. However, on this part, the input list is going to be the output list of the predictions of the technical analysis model. For this reason, we need to execute the technical analyst part first.

The execution of the fundamental analysis is less complex than the ML models done before, because we get the data already prepared and we just need to score them and make the final decision. Nevertheless, this part contains a lot of values, and is a fundamental part of our program. Next, we will see and explain the flowchart of the fundamental analysis function.

Point 1:
Obtain the list_predictions with the result of the ML model. The list contains the acronyms of the companies predicted.

Point 2:
Loop for all the companies of the list_predictions.

Point 3:
Use the HTML request with the FMP API, to access to the JSON file of the company.

Points 4, 5, 6, 7:
Functions of the four groups of ratios. For each function, search and save the pertinent ratios, then apply the score and finally return the average score of all the ratios.

Point 8:
Add the average score of all the groups and save it as "final result".

Point 9:
Take the decision with the "final result". Save the company on "list_recommendations" if the "final result" is higher than 5 or discard if it is less.

Point 10:
Return the "list_recommendations".

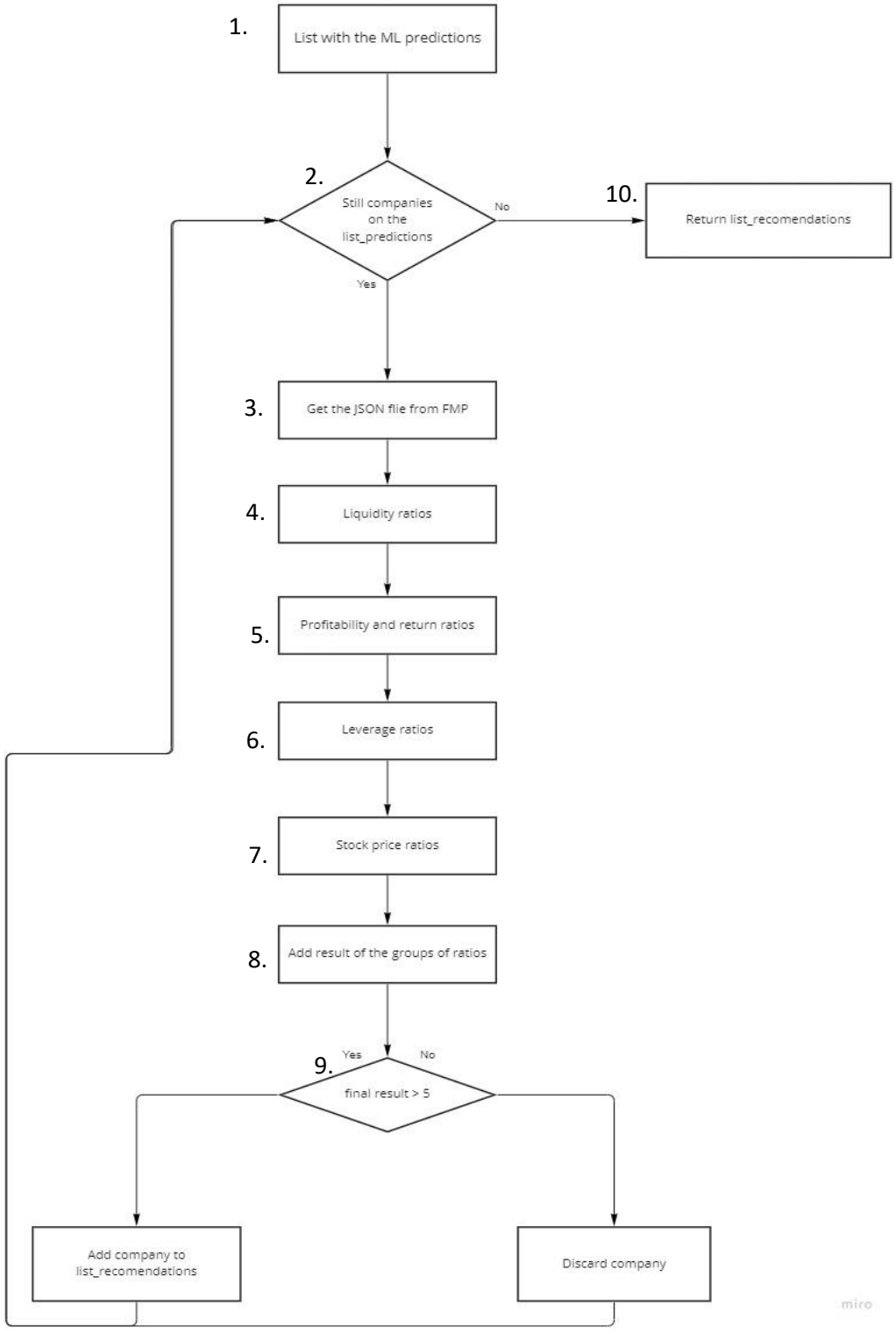

Figure 9 Flowchart of the fundamental analysis code.

Once we create the fundamental data function, we are going to be able to apply it in all the different prediction lists, due to all of them having the same output format.

3.4 The server:

The last step on the program is merge the technical and fundamental code and make it a daily forecasting model. To do that, we need to convert the static code that we had created to a dynamic and daily real-life prediction program. For this reason, we had created what we call the "Server" code. This code is an infinite loop that executes the predictions every weekday at 22:00, Spanish time, and prints the results on our Twitter account. During the rest of the time the code is reading if there are any new follower in our Twitter account. If there is a new follower, then the code will tell the Twitter account to follow it back and send a welcome direct message.

The execution of the predictions, start with the technical code. Firstly, we execute the functions of the five models, and save the lists with the companies predicted. Once all the functions are executed, and we have the list with the predictions, we are going to introduce those lists in the fundamental analyst code, which is going to select the companies with better financial performance, and return them in the recommendation list, for all the different models. The recommendation list with the companies is the information we are going to update on our Twitter account. We are going to explain how we display the information on the next point (3.4.1)

Although we already have our predictions saved in our Twitter account, we also need to save the predictions internally. The predictions are going to be saved in a ".csv" file. This format allows data to be saved in a tabular format and can be displayed as a table. There is going to be one .csv file for each period of predictions, and inside the file we are going to have a column for each model-company prediction. Because we have seventeen companies and four models, we are going to have sixty-eight columns. For every daily prediction, we are going to add a row, with all the result of the predictions. Finally, we are going to save the file in our hard disk drive (HDD), so we can update it every day. These files are the ones we are going to use to study the results of our predictions.

3.4.1 Displaying the results:

The final step on our program is to print the final recommendation into our Twitter account. There is going to be one tweet for each period of predictions and the tweet will contain the four different models. However, if the information that needs to be displayed has more than 280 characters, which is character limit in Twitter, there is going to be more than one tweet for that period. In addition, we are going to have a fourth tweet for the BM model with the recommendations of the three periods of prediction. Therefore, we are going to have at least four daily tweets, the recommendations for i) the 4 month prediction, ii) the 1 month prediction, iii) the 2 week prediction and iv) the predictions of the BM model.

The information we are going to display for each period is firstly going to be the date of execution. Secondly, we will display the period of the predictions, and lastly, the results of the recommendation list from the four models. The next tweet is an example of the how the information is displayed.

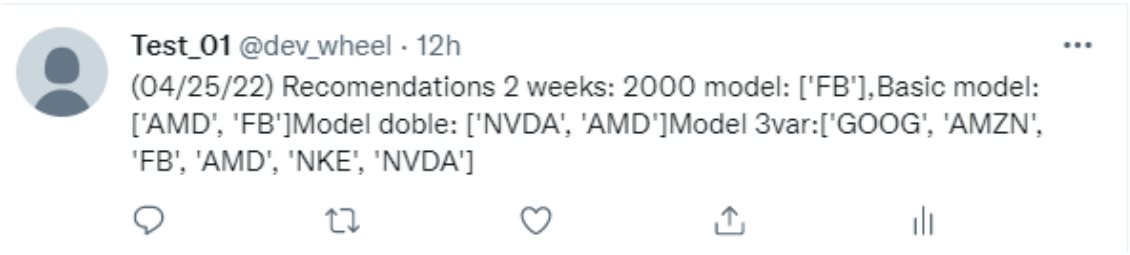

Figure 10 Example of a results displayed on our Twitter account.

In the example above, we can see one the tweet with the 2 week predictions for the day 25/04/2022. For that day, the model 2000 predicts that Facebook ("FB") is going to return a 4% of the investment in 2 weeks, and The Basic Model predicts that AMD, and FB are going to going to return 4% of the investment in 2 weeks. For the Model Doble the predictions are NVIDIA and AMD. Facebook, AMD, Nike, and NVDA are going to be the predictions for the 3var Model.

3.4.2 Twitter:

There are four reasons to use Twitter as a displayer of the information. We are going to explain them below:

1. The first one is that twitter, with more than 217 million users, is one of the largest social media. What is more, the twitter target audience is people that are aged between the 25 and 49, which is also our main target, as it is the average years where people to start investing. According to Gallup Poll, this is around 29, when the investor is more unexperienced. Furthermore, twitter have an impressive investing community, for example, the well know financial company Bloomberg have more than 8 million followers. Nevertheless, there is also a large and active community of professional investors, small investors and investing companies.

2. Another reason to use Twitter is that they have *tweepy*. This useful API, which give us a lot of options to allow us to interact with twitter. Displaying the information on twitter is an easy process and never presents any problems. What is more, this API, allows us to follow back any new followers and send a welcome direct message.

3. On more reason to use the twitter API is because the execution of our code is automatic and at the right time. Initially, we were planning to create a webpage or an app to display

the results, however, creating a webpage to execute a code that takes more than one hour is not efficient. Consequently, we decided that using the twitter API would be an optimal option to show the results.

4. Finally, the last reason for using twitter as the displayer of the results is that is open to anyone. Everybody can access to the information for free and use it. There is another reason to make the results open to everyone, and this is because anyone can check the performance of our model and prove whether the results of our predictions are correct or not. This program is totally automatized and transparent, so there is not any human interaction.

The Twitter account is: @dev_wheel

4. The results:

4.1 Context:

Unluckily for us, it seems that we started testing our model at the worst possible time. The testing started at the middle of January, and since then the NASDAQ Index has drop around 20%. This is an unexpected and unusually event in the stock market, and it was the first time since the 2008 financial crisis where the stock market had such a remarkable drop. Therefore, creating a model based on the return on the inversion during 2008 would not have been good timing similar to creating one in the current climate.

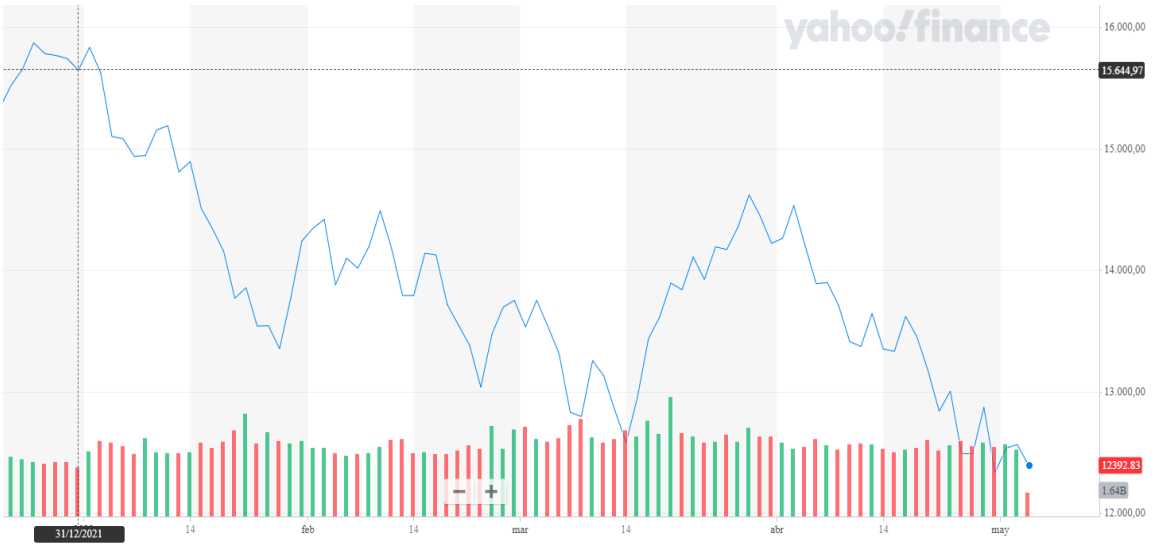

Figure 11 Graph of the evolution of the NASAQ Index on the first half of 2022.

During this period of instability that we are currently in, investing is difficult and it is very hard to make predictions. For this reason, we emphasise that this program is just a recommendation, and that there is always a risk on an investment. Even the most experienced investors or programs

cannot predict a war, the increase in fuels price, a virus, or any of these events that affect the global economy and the stock market. For these reasons, our predictions are not as good as they could be in a stabile period. However, even though our results were not as expected, we find that our model has better performance than the market.

To analyse the results of our predictions we are going to use a linear graph. The plot is going to contain 7 lines (outlined in the graph below). The model lines are going to contain the average return of all the predictions of each specific model. The market return line shows the average return for all the companies that we are analysing. For The average no recommendation line is the average return of the companies that our model does not recommend investing in. We are going to analyse the 3 periods of time with the models Basic, 2000, 3var and Dob, and then we are going to analyse the BM model separately.

4.2 Results 2 weeks:

As we will see in all the models, our predictions usually do not arrive to the return defined for the target variable. For the predictions in 2 weeks, our target variable was that the return of the stock would be 4% or more in two weeks, however, the average return of all the models together is –1.09%. If we analyse the different models separately, we are going to see that the one with better performance is the Basic Model with a 0.8% of the return of the investment. In addition, the models 3var and Dob also have a positive return even though they are very close to zero. Nevertheless, the performance of the market during this period of time is significative lower (–2 .73%). This means that even our predictions do not achieve the 4% predicted for our models, the return of the predictions is 2% better than the market. This can be an achievement considering that the average investor gets lower returns than the market.

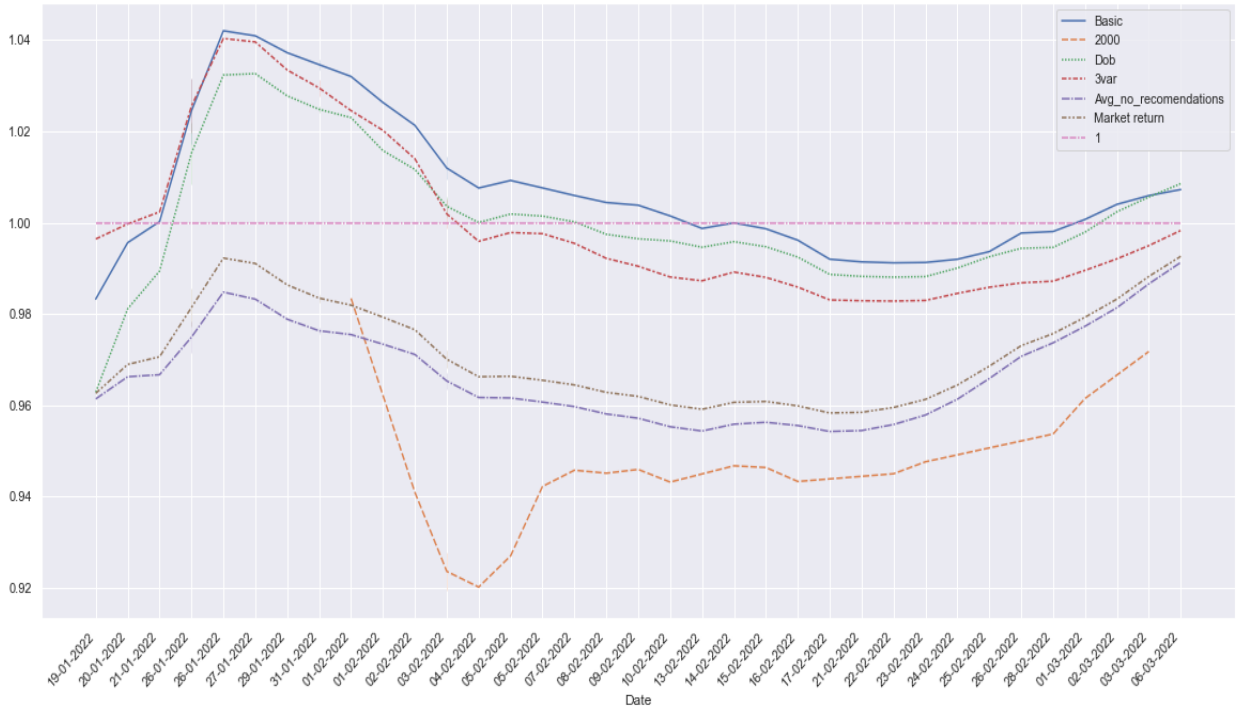

Figure 12 Graph with the results of the 2-weeks predictions.

In the graph above, we can see the results of the different models, the market return and the average return of the companies that we didn't invest in, from the 19th of January until the 6th March. We can easily appreciate the gap between our predictions and the market return where in most of the cases is bigger than 2%. Also, there is another gap between the market return and the companies that we do not invest in, which mean that our program is selecting the companies that perform better instead of the worse ones. Another fact that we can notice in this plot is that the different models follow similar trends, and this is due to the market fluctuations affect all the companies in some way, and so our predictions are impacted too. If we take a look at the model 2000, we are going to see that it is the one with worse performance compared to the other models.

Considering the Basic Model as the definitive model, we can affirm that our program is able to select the companies with better performance and discard the once with the worst. What is more if we look to the results of companies selected the return of the investment is 3.5% above the others, so considering that we are analysing 2 weeks inversions, that is a remarkable difference

4.3 Results 1 month:

For the one-month predictions, the predictions of our models also have a negative return, (-3.36 %) on average. Specifically, for this period there are just fifty-seven predictions in total that arrived at the 7% expected, return, with over seven hundred companies predicted. However, similar to the other model, the performance of our model is better than the performance of the market, as the market has a performance is (-6.47 %). In this case, the difference between our models and the market is even higher, almost double.

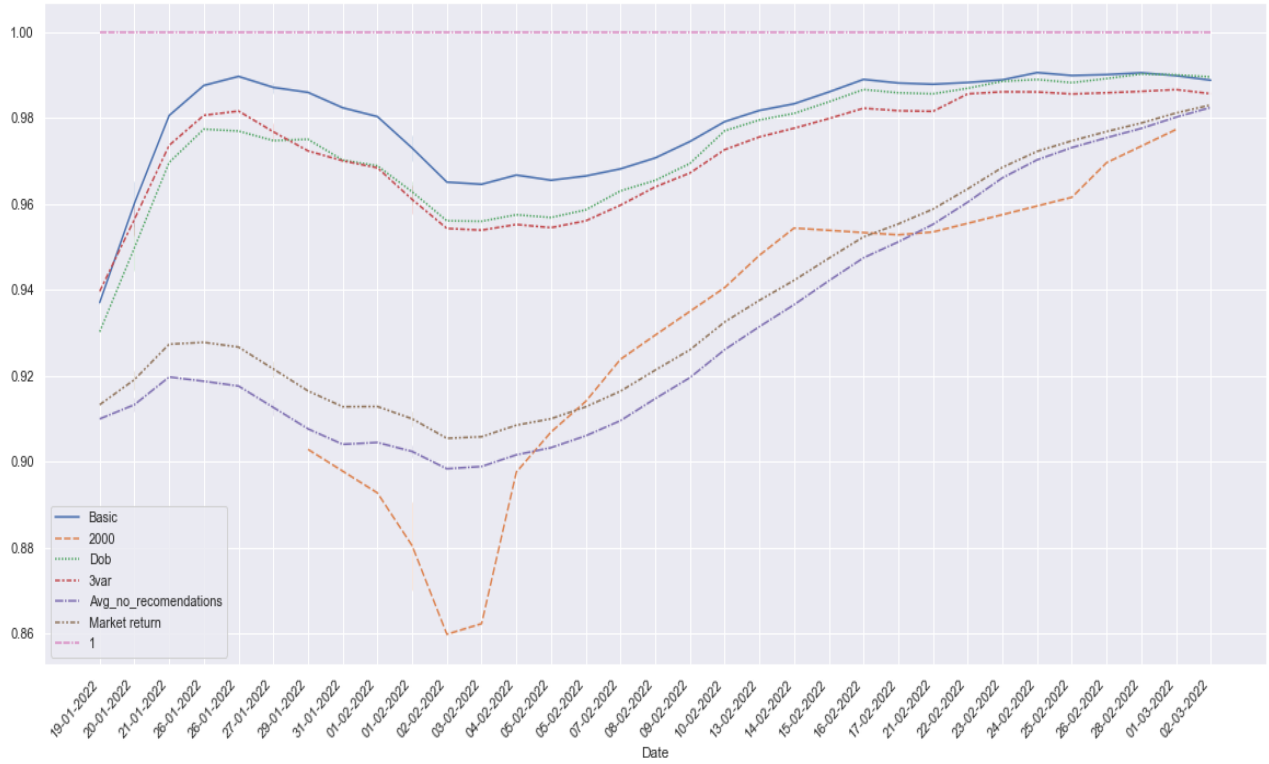

Figure 13 Graph with the results of the one-month predictions:

In the one-month prediction graph, the gap between the predictions of our models and the market return is clearly evident. As seen in the 2-week predictions, the model with better performance is the Basic model, followed by the 3 var. The model 2000 has the worst performance as well. If we analyse the performance of the Basic Model, the average return is (– 2.11%), that is 4.36% better than the average market return in just one month. The gap between the market and that model is remarkable.

4.4 Results 4 months:

Unfortunately, for this model we did not have enough time to prove our results. Although we have 5 days of predictions, we believe this is not enough data to prove the results of this model. However, if we analyse the results obtained during the 5 days, they follow the same trend of our predictions. The Basic Model is one of the best performance, and the 2000 model is the worst model. The average predictions are mainly negative too.

4.5 BM model:

Next, we are going to analyse the BM results. The main reason why we are going to analyse this model separately is because there are not many predictions, and the predictions behave differently to the other models.

For the results of the BM model, we can only show two predictions of the 2 weeks period, both from the Tesla company. Both predictions have a negative return, (-10.38%) on average. These are the worse results in all the models, however there is not enough data to prove that the model is not accurate.

This model does not contain any one-month predictions, being the only model-period without predictions. The BM model has 129 four-months predictions; however, we do not have any results, as it has still not been four months since the predictions. This model should be tested in more detail, as it has different behaviour, and it could be interesting to take some insights from that model.

4.6 The interpretation of the results:

There are two different ways to interpret the results of our predictions. The first method is to study the results in an absolute way. Using this method, we are going to analyse if our program gets the predictions right or not. In this case, our program has really poor performance, as the results of the predictions do not usually reach the limit expected. The other way to analyse the results is in a relative way. In this case, our program is able to select the companies that have better performance and as a result, our predictions have higher return of investment that the market.

As we saw in the different results, our predictions are significantly higher than the average market return. This gap is remarkable, as the average investor sees lower results than the market. According to Robert Laura (2019), president of SYNERGOS Financial Group, the average

individual investor has little chance of beating the market. Statements like that reinforce the significant performance of our program.

In addition, the performance of the different models behaves similarly in the different periods, and during the time. This confirms that the behaviour of or model is not random and that the performance of the different models is constant. If we just consider the 2 week and 1 month periods, our models predict 2720 companies. This is a considerable sample of data, to prevent outliers and provide a smaller margin of error. For this reason, we can affirm that the Model Basic Model is able to select the companies that are going to have better performance, and discard the companies with worse performance just by analysing and processing the data.

We believe that if we tested our model during a period of stability, the results would have been much better, and also our program would have been right in the majority of predictions. However, this is a hypothesis, and to prove this, we should extend our testing period.

Finally, once we analyse the results of the different models and periods of time, we must study the performance of each model and each period of time in order to take the maximum information and decide which model we are going to keep, which we are going to discard, and also if we want to add more variables to a model or merge two different models in one. For our program, we are going to discard the model 2000 due to the poor performance in all the periods, and also the BM model, due to the low number of predictions. The model Basic is the one that we would take as the definitive one. We note that the 3 var model and the Dob model also have a positive performance.

In conclusion, we would use the Basic model for the future predictions, but we will still test the 3 var, the Dob and the BM model, as there is maybe a characteristic of those models that we could add in our main model. In addition, if we add more companies to our predictions, we will have more predictions, and so the models would be more accurate. Once the results have been analysed, we can identify that the information provided by our program could be interesting, and therefore we could provide it, to possible users. To analyse if this project could become a viable business, we create a study of the viability of the project, as shown in appendix 3

5. Conclusions:

5.1 Main contributions of the Project

During the development of the project, we have been using and analysing different web pages that provide financial information, such as yahoo finances, Financial Modeling Prep, FinViz, etc. These web pages contain a massive amount of information about many companies, which helps us to understand the situation of the companies and their background. However, all this information is useless if you don't know how to interpret it or use it. Consequently, a significant group of investors use their intuition to invest, and as we saw in investor irrationality, often this is not the best option.

For this reason, our program is one of the first investing programs created for people without any basic knowledge of finance or IT. We believe that the technology and AI must take one step further

and try to help all investors reduce their risk. As we are aware that a considerable number of small investors do not use or understand the information provided in the well-known webpages, we believe it is important to create a program that can do that for them and also reduce the risk of their investment. What is more, the program is so easy to use and interpretate, as the user just needs to check the twitter account to see if there is any recommended company that he or she is interested to invest in.

The fact that our predictions are posted on a twitter account also makes a difference. Firstly, because this offers the users the execution of a complex ML program for free, and also because there are no requisites of installation or waiting for the code to be executed. Secondly, this method offers the user the ability to prove the results of our predictions, at any moment.

In addition, the target of our program is an important part of the stock investors, that have almost no kind of support. This unexploited niche of small investors can use the ML algorithm and the processing of the financial data to supply their lack of knowledge on the field.

5.2 Limitations:

However, our program has some limitations and problems than have to be mentioned.

The time was always our major disadvantage during the creation of the program. The reason is that as any forecasting model, we need to prove the results, and due to our model have a long-term predictions (4 months predictions and 1 month predictions), we need at least 4 months of predictions to prove these results. Another problem related with the time we had is that we had to limit the scope of the project. The most affected part was the fundamental analysis. As we saw on the section (3.3.2), the FMP API contains a huge amount of information that we could use in our model. However, in order to use that information, it is necessary analyse and study the ratios before introducing them in our program. This means that with more time to analyse the FMP API ratios, the image of the financial state of the company would be more accurate, and so this would reduce the risk even more.

The second problem is the limitation of computational power. As we saw on the section Hardware requirements (2.5), our PC is not the indicated hardware to execute this complex ML models. For this reason, instead of executing or testing our model with 1000 companies, we had to use just 17. As per any ML model, the more predictions wee test the more accurate the model is going to be and so the more trusted. In our case, we had to use our limited capacity to create a limited sample of companies, however we have to consider that this was a test with a sample of data.

The final problem or limitation is that investing is never free of risk. There is one basic rule of investing," the risk and return correlation". This rule explains that the more risk averse you are, the lower your expected return would be. This means that there is not an investment without risk. Although our model can reduce the risk of the investment, this does not mean that there is no risk. For this reason, investors should only take our model as a recommendation. where to invest, instead of advising. There is also another sentence "Past performance is no guarantee of future

results", which means that studying or working with past data does not guarantee future gains. This is the main disadvantage of the technical analysis, which is why we were applying the fundamental data to reduce this risk of just working with the historic data of the stock price.

5.3 Improvements and Directions for further research

Once we are aware of our limitations and our differentiating points, we believe that by studying, and adding more variables in the program, we will be able to improve the results. Considering the low cost of the study, and the amount of information we are able to extract from this, we believe that if we keep improving or testing the model with new variables, we can reduce the risk even more, and improve our predictions.

We would need to create a web page to create a flexible program that gives the user the options to interacts with our code. Firstly, we could create a page where the user could enter the different companies that he/she wants to study and execute the code to obtain the predictions. In addition, during the execution of the code, we could explain to the user the financial situation of the companies, based on the result of the ratios, helping the non-finance users to start learning the meaning of the ratios and also giving more information of where he/she wants to invest in. Furthermore, we could rank all the companies studied based on their scores and the results of our predictions. With this ranking, we could help the user decide which company could have better performance based on our program. There are also more ideas to be explored and tested based on our program, that would be interesting apply.

What is more, extending our program would allow us to introduce more information, for example macroeconomic metrics such as: the interest rate, inflation rate, the economic growth and more. Improving the information of the fundamental data is going to give us a more accurate image of the background of the company, and so reduce the risk of the investment.

The creation of a web page implies the need of a server or a hosing service. In the point (2.5.) we saw that if we run a program in a server, the execution would be much faster, and also it would allow us to add more information and make our code even more complex and accurate. The server is the only expense needed in the hypothetical initial investment. There are different options to obtain a server: The first option is creating our own server, this option requires a bigger initial investment, around 2,000€. The second option is hiring a server service that for our requisites the expense would be around 720€ - 1.800€/year.

5.4 Final conclusions:

The stock market contains a lot of factors or variables than influence their moves, making investment a hard and risky task. However, considering the main risk factors in an investment is an important key to improve or succeed in the investment field.

During the study of the performance of the stocks, we saw that even by analysing the behaviour of the main risk factors, there isn't any rule that determines cause-effect with the investment return. However, if we work with probabilities, we can find some patterns, that even though they are not fulfilled in all cases, we can affirm that there is a correlation. In addition, if we use multiple variables, we can consider the risk of different camps and therefore be more aware of the risk and prevent it.

Studying the different risk factors of the companies is what professional investors do. During the development of the program, we saw that with the necessary data and a great knowledge of the main ratios, metrics or information, we can recreate the decision making of the professional investors. Considering that our decision-making algorism is basic, and we are not experts on the stock market, the results of our program are remarkable. For this reason, with the recreation of an expert decision process, the results could be even better.

In conclusion, after the deep and meticulous research of the stock market using the technical and fundamental analysis, we can affirm that in most of the cases the average investor makes poor conclusions due to the limited scope of their research or study. In addition, the background and information that the experienced or professionals have, gives them the ability to prevent risks that the small investors do not consider. However, technologies such as AI and Big Data can consider multiple risk factors and try to prevent them. For this reason, this program can help the small investor to take some profitability of their saving, without deep analysis or high risk.

References:

Abderrazak Dhaoui ,Sami Bacha & David McMillan (2017): *Investor emotional biases and trading volume's asymmetric response: A non-linear ARDL approach tested in S&P500 stock market, Cogent Economics & Finance*.
(10.1080/23322039.2016.1274225)

Bernales, Alejandro and Valenzuela, Marcela and Zer, Ilknur (April 1, 2022): *Effects of Information Overload on Financial Markets: How Much Is Too Much?*
(https://ssrn.com/abstract=3904916 or https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3904916)

Dana Anspach and David Kindness (November 22, 2021): *Why Average Investors Earn Below-Average Market Returns.*
https://www.thebalance.com/why-average-investors-earn-below-average-market-returns-2388519#:~:text=Research%20by%20Dalbar%2C%20Inc.%2C,investor%20earns%20below%20average%20returns.&text=Why%20is%20this%3F,wise%20long%2Dterm%20investing%20decisions.

June Yee Felix (2022): *Stock market definition.*
https://www.ig.com/uk/investments/support/glossary-investment-terms/stock-market-definition

Katrina Wakefield, 2022. A guide to the types of machine learning algorithms | SAS UK.
https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html

Data-Driven Science Team (Jul 20, 2020): *7 Stages of Machine Learning — A Framework.*
https://medium.com/@datadrivenscience/7-stages-of-machine-learning-a-framework-33d39065e2c9

Sebastian Raschka (2016): *Python Machine Learning.*
https://books.google.es/books?hl=es&lr=&id=GOVOCwAAQBAJ&oi=fnd&pg=PP1&dq=machine+learning&ots=NdbEPeRUUD&sig=oyoTI7rA5zkf1W84GWNb8cEsLUQ#v=onepage&q=machine%20learning&f=false

Ana Gonzalez Ribeiro (October 16, 2019): *Can Regular Investors Beat The Market?*
https://www.investopedia.com/articles/trading/10/beat-the-market.asp

Robert Farrington (October 18, 2021): *How To Start Investing After College*
https://thecollegeinvestor.com/17809/start-investing-after-college/#:~:text=Final%20Thoughts-Why%20Start%20Investing%20Early%3F,versus%20your%2030s%20or%20later

Katrina Munichiello (March 03, 2022): *What Is Considered a Good Net Debt-to-Equity Ratio?*
https://www.investopedia.com/ask/answers/040915/what-considered-good-net-debttoequity-ratio.asp

Appendix

Appendix 1: Logistic Regression:

Linear Regression vs Logistic Regression

Whereas the Linear regression is suitable for predicting a continuous value such as predicting the price of property based on area in square feet. Logistic regression on the other hand is used for classification problems which predict a probability that a dependent variable Y takes a value of 'one', given the values of predictors. In binary logistic regression, the regression curve is a sigmoid curve.

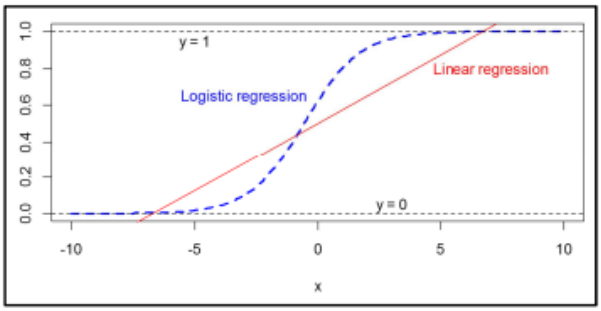

Figure 14 shows the difference between linear regression and the binary logistic regression. Where the variable y is going to be our target variable or the binary response variable

In particular, the key differences between these two models can be seen in the following two features of logistic regression. First, the conditional distribution y | x is a Bernoulli distribution rather than a Gaussian distribution because the dependent variable is binary. Second, the predicted values are probabilities and are therefore restricted to (0,1) through the logistic distribution function because logistic regression predicts the probability of particular outcomes.
On binary logistic regression the values of the dependent variables represent the percentage of probability, given by the independent variables.

Binary Logistic Regression

Next we are going to explain how Binary Logistic Regression works:

Letting y be the binary response variable, it is assumed that p(y = 1) is possibly dependent on X, a vector of predictor values. The goal is:

$$\boldsymbol{p(X)} \ = \ \boldsymbol{p(y = 1|X)}.$$

If the model p(X) as a linear function of predictor variables , then the fitted model can result in estimated probabilities which are outside of [0,1]. What tends to work better is to it's called

multiple logistic regression. The outcome of the regression is not a prediction of a *y* value, as in linear regression, but a probability of belonging to one of two conditions of *y*, which can take on any value between 0 and 1 rather than just 0 and 1. However a further mathematical transformation (a log transformation) is needed to normalize the distribution. This log transformation of the p values to a log distribution enables us to create a link with the normal regression equation. The log distribution (or logistic transformation of *p*) is also called the logit of *p* or *logit(p)*. L*ogit(p)* is the log to base odds of the odds ratio or likelihood ratio that the dependent variable is 1. In symbols it is defined as:

$$Logit\ (odds)\ =\ log\left(\frac{p(X)}{1-p(X)}\right) = ln\left(\frac{p}{1-p}\right).$$

Where p can only range from 0 to 1, logit (p) scale ranges from negative infinity to positive infinity and is symmetrical around the logit of 0.5 (which is zero). The form of the logistic regression equation is:

$$Logit\ (odds) = ln\left(\frac{p}{1-p}\right) = \beta_o\ + \beta_1\, x_1 + \cdots + \beta_K\, x_K$$

This looks just like a linear regression and although logistic regression finds a 'best fitting' equation, just as linear regression does, the principles on which it does so are rather different. Instead of using a least-squared deviations criterion for the best fit, it uses a ML method, which maximizes the probability of getting the observed results given the fitted regression coefficients

Appendix 2 Code:

Variables Balance:

```
list_balance_days = [750, 375,120,60, 15, 5, 2]

def all_balances (df_auto):

    for dates in list_balance_days:
        i = 750
        total_rows = df_auto["Close"].count() -1

        # Colomn to crate:
        column_name =  "Balance_last_"+   str(dates)  +  "_days"
        df_auto[column_name] = ""



        while i <= total_rows:


            df_auto[column_name].iloc[i] = df_auto.iloc[i]["Close"] / df_auto.iloc[i - dates]["Close"]

            i = i + 1

    return(df_auto)
```

Figure 15 Variables Balance

Target Variable:

```
sell_list = [ 10, 21, 92]

def Sell_in_periods (df_auto):

    for dates in sell_list:
        i = 750
        total_rows = (df_auto["Close"].count() -1 ) - dates

        # Colomn to crate:
        column_name =  "Sell_in_"+   str(dates)  +  "_days"
        df_auto[column_name] = ""

        while i <= total_rows:


            df_auto[column_name].iloc[i] = df_auto.iloc[i + dates]["Close"] / df_auto.iloc[i]["Close"]

            i = i + 1

    return(df_auto)
```

Figure 16 Target Variable

Variable distance from maxim and minim:

```
days_distance_min = [ 21, 92]

def dist_min (df_auto):

    for days in days_distance_min:
        # Loop variables
        i = 750
        total_rows = df_auto["Close"].count() - 1

        # Column to create:
        column_name =  "Distance_min_of_last_"+   str(days)  +  "_days"
        df_auto[column_name] = ""


        while i <= total_rows:

            df_auto_month = df_auto[i - days: i ]

            min_months = df_auto_month["Close"].min()

            value_today = df_auto.iloc[i]["Close"]

            distance = value_today / min_months

            df_auto[column_name].iloc[i] = distance

            i = i + 1

    return(df_auto)
```

```
days_distance_min = [ 21, 92]

def dist_max (df_auto):

    for days in days_distance_min:
        # Loop variables
        i = 750
        total_rows = df_auto["Close"].count() - 1

        # Column to create:
        column_name =  "Distance_max_of_last_"+   str(days)  +  "_days"
        df_auto[column_name] = ""


        while i <= total_rows:

            df_auto_month = df_auto[i - days: i ]

            max_months = df_auto_month["Close"].max()

            value_today = df_auto.iloc[i]["Close"]

            distance = value_today / max_months

            df_auto[column_name].iloc[i] = distance

            i = i + 1

    return(df_auto)
```

Figure 18 and 18 Function to create the variables distance form max and min

Clean df:

```
## Clean the df:
rows = df["Close"].count()
limit_top = rows - 21
df_clean = df[750:limit_top]
```

Figure 19 Code to cut the extra data.

Model Basic Code:

```
def model_basic(stoks):

    for stock in stoks:
        try:
            #Get the data
            #NASDAQ
            df_NASDAQ = data.DataReader(name = "^IXIC", data_source = "yahoo", start = "2012-02-10")
            #Company
            df_comp = data.DataReader(name = stock, data_source = "yahoo", start = "2012-02-10")

            # Data Preparations:
            # Create all the balances
            df = all_balances(df_comp)
            ## Create balances NASDAQ:
            df = balances_NASDAQ(df,df_NASDAQ)
            ## Create Sell
            df = Sell_in_periods(df)
            ## Distance_mimim
            df = dist_min(df)
            ## Distance_maximum
            df = dist_max(df)

            ## Clean the df:
            rows = df["Close"].count()
            limit_top = rows - 92
            df_clean = df[750:limit_top]

            ## Create Groups
            df_clean = apply_groups(df_clean, stock )

              ## Target variable
            df_clean["15%_profits"] = np.where(df_clean["Sell_in_92_days"]>= 1.15, 1, 0)

            # ML Prediction:
            ## Apply the ML
            log_reg = modelo_LR(df_clean, stock)

            # Prediction of the actual day:
            df_today = df[-1:]
            df_buy = add_groups_today(df_today, stock)

            # Apply Logistic Regression actual day
            today_values = df_buy.values
            Buy_today = log_reg.predict(today_values)
            proba_each = log_reg.predict_proba(today_values)
            buy_today = Buy_today[0]

            # Decision modelo basico:
            if buy_today == 0:
                List_Not_buy_Basic.append(stock)
                print(stock, "Not predicted today")
            if  buy_today == 1:
                List_buy_Basic.append(stock)
                print(stock, "Predicted Today!!")
        except:
            print("Error in ", stock, "data model Basic")
            List_Not_buy_Basic.append(stock)


    return List_Not_buy_Basic, List_buy_Basic, List_buy_Acc_Basic, List_Not_buy_Acc_Basic
```

Figure 20 Model Basic function.

```
[ {
  "dividendYielTTM" : 0.02929313929313929,
  "dividendYielPercentageTTM" : 2.92931392931392900,
  "peRatioTTM" : 9.897119,
  "pegRatioTTM" : -5.625868719147189,
  "payoutRatioTTM" : 0.2885655828467888,
  "currentRatioTTM" : 2.1017405869929355,
  "quickRatioTTM" : 1.378996431432525,
  "cashRatioTTM" : 0.1757701551234433,
  "daysOfSalesOutstandingTTM" : 43.68046416278599,
  "daysOfInventoryOutstandingTTM" : 111.711210201937,
  "operatingCycleTTM" : 122.48173561193903,
  "daysOfPayablesOutstandingTTM" : 59.577238774177054,
  "cashConversionCycleTTM" : 11.648455301415318,
  "grossProfitMarginTTM" : 0.5544518121077141,
  "operatingProfitMarginTTM" : 0.2462036849564689,
  "pretaxProfitMarginTTM" : 0.2746380846325167,
  "netProfitMarginTTM" : 0.2514172909495849,
  "effectiveTaxRateTTM" : 0.08455052296917477,
  "returnOnAssetsTTM" : 0.11797679417597948,
  "returnOnEquityTTM" : 0.22674411969460073,
  "returnOnCapitalEmployedTTM" : 0.1380406402542854,
  "netIncomePerEBTTTM" : 0.9154494770308252,
  "ebtPerEbitTTM" : 1.115491365131579,
  "ebitPerRevenueTTM" : 0.2462036849564689,
  "debtRatioTTM" : 0.43356531239979573,
  "debtEquityRatioTTM" : 0.7654286043756748,
  "longTermDebtToCapitalizationTTM" : 0.2599669513812926,
  "totalDebtToCapitalizationTTM" : 0.2854178527552213,
  "interestCoverageTTM" : 32.58961474036851,
  "cashFlowToDebtRatioTTM" : 0.7871446943649772,
  "companyEquityMultiplierTTM" : 1.7654286043756748,
  "receivablesTurnoverTTM" : 8.356138310246378,
  "payablesTurnoverTTM" : 6.126500783017226,
  "inventoryTurnoverTTM" : 3.267353377876763,
  "fixedAssetTurnoverTTM" : 1.2494900782670566,
  "assetTurnoverTTM" : 0.4692469389451682,
  "operatingCashFlowPerShareTTM" : 7.3706070287539935,
  "freeCashFlowPerShareTTM" : 2.374539198820349,
  "cashPerShareTTM" : 6.982796755959695,
  "operatingCashFlowSalesRatioTTM" : 0.37951761490180197,
  "freeCashFlowOperatingCashFlowRatioTTM" : 0.3221633156613651,
  "cashFlowCoverageRatiosTTM" : 0.7871446943649772,
  "shortTermCoverageRatiosTTM" : 6.532563711609671,
  "capitalExpenditureCoverageRatioTTM" : 1.4752816173938708,
  "dividendPaidAndCapexCoverageRatioTTM" : 1.4752816172916192,
  "priceBookValueRatioTTM" : 2.0517543583776248,
  "priceToBookRatioTTM" : 2.0517543583776248,
  "priceToSalesRatioTTM" : 2.47670201457785,
  "priceEarningsRatioTTM" : 9.850961344876184,
  "priceToFreeCashFlowsRatioTTM" : 20.256561788449595,
  "priceToOperatingCashFlowsRatioTTM" : 6.525921109666234,
  "priceCashFlowRatioTTM" : 6.525921109666234,
  "priceEarningsToGrowthRatioTTM" : -5.625868719147189,
  "priceSalesRatioTTM" : 2.47670201457785,
  "dividendYieldTTM" : 0.02929313929313929,
  "enterpriseValueMultipleTTM" : 6.7169101255426495,
  "priceFairValueTTM" : 2.0517543583776248,
  "dividendPerShareTTM" : 1.4089999999999998
} ]
```


Figure 21 JSON file for intel

Group variable 3 month function:

```
def Balance_3meses (balance, company):

    v1 = lista_lista[company][6]

    v2 = lista_lista[company][7]


    if (balance >= v1):
        return 3
    if (balance < v1) and (balance >= v2 ):
        return 2
    else:
        return 1
```

Figure 22 Function to group the variable 3_months

Appendix 3: Viability of the project:

During the explanation of our project, we had been telling that our code is a new model of business based on information, this type of model is called data-driven business models. The main function of this kind of business is convert the raw data, or data provided by the companies that contract them to highly-value information. Due to principal asset of those business are the data, the initial investment is much lower than the average business. Meanwhile the initial investment of the average business is around 85.000€, in these types of business the initial inversion is around 7.500€. However, in our case the initial inversion is even lower around 2.200€ (1.250€ server + 700€ advertising + 250€ in FMP premium), due to we are geting the data for free. That budget would be the necessary to covert our ML model to a actual online service for a possible users. Now we are going to see which sources of income we could have.

There are two possible ways to obtain revenues from this program, deepening of the witch of both method we use our business model is going to change:

1. The first one is affiliating our code to an investment company or a professional's investors, in order to improve their services. This method, can create a great partnership with some experienced investors that help us in the decision-making of our code and improve the performance of our predictions, making a great deal for both sides.

2. The second option to obtain revenue is creating a subscription web page, where the users must pay for the use of the program. With this method our revenue is more uncertain, however the profit margin is much higher. The subscription would be annual, to reduce the low term risk, and the cost would be around 35 – 50 € / year. Getting a middle price of 42.5, with just 52 customers we would recover the initial investment. Using this method, the marketing and the advertise would be essential the first moths, nonetheless if the marketing strategy is adequate, we believe we can get a 20 users per month.

As we can see, our program has economic viability, and so a great business opportunity to be considered.