NOVA IMS Information Management School

MEGI

Mestrado em Estatística e Gestão de Informação

Master Program in Statistics and Information Management

Data Science for finance: automated investment recommendation with python

Marcio Jonavicius Rodrigues

Dissertation

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

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

DATA SCIENCE FOR FINANCE: AUTOMATED INVESTIMENT RECOMMENDATION WITH PYTHON

by

Marcio Jonavicius Rodrigues

Dissertation presented as partial requirement for obtaining the master's degree in Statistics and Information Management, with a specialization in Risk Analysis and Management

Advisor: Prof. Mauro Castelli

February 2022

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude to Nova IMS for all support. I would like to thank Professor Jorge Mendes, who was my tutor in the research methodologies class and since the very beginning, he helps us to develop the work, including given essential suggestions and comments for the development of the dissertation.

I am extremely grateful to my elder brother, Wagner Gomes Rodrigues Jr., for his support and encouragement since the beginning of my university studies.

Furthermore, I would like to thanks to my wife for all time and patience devoted to this dissertation. Your support in the final steps of this project was essential. Finally, my family and friends, who offered all the support I needed throughout the path.

ABSTRACT

Most investors have difficulty finding good companies to invest in the stock market. As there are several stock options, it is hard to monitor the companies' performance. So, this thesis aims to backtest to validate some investments strategies. The backtest includes fixed income and variable income investments, but specifically stock investments.

The analyses are made in the Brazilian market. For fixed income, we calculate the profitability obtained from investments in treasury bonds, using the inflation and the basic rate of the economy as indexers.

For variable income, we test some strategies of stock selection. We analyse Joel Greenblatt magic formula and Ben Graham's formula for choosing stocks. We also try to create a model to select stocks based on the quote and fundamental analyses.

This project aims to automate the selection of stocks based on historical data and fundamental indicators. It does not claim to be a generic model, as it would be unfeasible since there are several points of view according to different investor's profiles.

The project also shows where reads can get data to create a model. Then the reader can use their market knowledge to modify the model and thus create a model that suits the reader's preference. This research will be based on the Brazilian market but may be expanded by the reader of other markets.

This study is for long-term investors and not for day traders that need different tools and analysis.

KEYWORDS

Investment, Financial market prediction; Investment; Stock Market, Times Series; Data Science

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

EBIT: Earnings Before Interest and Taxes

D/E: Debt Per Net Equity

BVPS: Book Value Per Share

CL: Current Liquidity

PE: Price to Earnings

EPS: Earnings Per Share

ROE: Return on Equity

IV: Intrinsic Value

B3: Brazilian Stock Exchange

Ibovespa: Indicator for shares traded in the Brazilian stock market.

IPCA (*Índice de Preços ao Consumidor Amplo)*: Broad Consumer Price Index

Selic (*Sistema Especial de Liquidação e Custódia)*: Special System for Settlement and Custody

CVM *(Comissão de Valores Mobiliários)*: is an Autarchy linked to the Ministry of Economy, which has the objective of inspecting, regulating, disciplining, and developing the stock market.

IBGE *(Instituto Brasileiro de Geografia e Estatística)*: Brazilian Institute of Geography and Statistics.

1. INTRODUCTION

As there are several investment's modalities, it is hard for an investor to choose an investment strategy that is profitable and makes the investor comfortable. For example, the stock market trades several companies' stocks, and choosing the best stocks is not an easy task. However, with the internet's development and statistic techniques, all of us have more access to information that can help the process of choosing good companies to invest in. The following thesis explains how to use fundamental indicators to make company valuations. Then, based on these indicators, we can apply different techniques to select good companies. Some techniques were tested, and the profitability of the investments is presented in this project.

One of this project's goals is to facilitate the decision-making process for investors, since investing in the financial market is a way to preserve and increase the investor's capital. Thus, an investor, especially the small and novice investor, should not spend a lot of time with the task of choosing the investment, but spend time with his profession which is the real income generator. Only when the investor has more money, he should consider the small variations of the market.

Therefore, this work is important because it allows simplifying the selection and investment process.

1.1. BACKGROUND AND PROBLEM IDENTIFICATION

Often Investors, especially the new ones, face the question "how can I choose good companies to invest in?". Investors search on the internet for recommendations or pay for specialized companies to recommend some good stocks to invest in. Furthermore, the market has several companies, and it is hard to analyse all of them to select the best ones.

The objective of this work is to test different investment strategies and compare the performance over a period of 5 years. In addition, some techniques that can help in the selection of stocks will be presented.

We cannot expect the media to provide us tips for choosing good companies. That is because, in an efficient market, when recommendations show up on news the price is already adjusted, and it can be too late to invest.

The investor also needs to be careful with certain stock purchase recommendations. Nowadays, there are several YouTube channels making stock purchase recommendations. Recommendations can only be made by certified professionals. That is because there are malicious investment recommendations, not all of them, but investors need to be aware. For example, a Youtuber can buy a share of company X, then make a recommendation on his channel. The listeners who believe him will also buy the shares of company X. As a consequence, the demand for the stock will increase and the stock price will raise as well, generating profit for the Youtuber that bought the shares first.

1.2. 1.2 Study Objectives

The main goal of this study is to validate some investments strategies. Using a backtest with 5-year historical data, we evaluated fixed and variable income investment strategies. This helps the investor to understand the types of investments and their returns.

In fixed-income investments, we compare treasury bonds investments using the indexers Selic rate and IPCA index.

For variable income, we compare the performance of the magic formula for choosing stocks of Joel Greenblatt, the Ben Graham formula for choosing stocks, and a machine learning model created using times series and linear regression. These two techniques are widely used by investment recommendation companies. We also compare the model performance with some indexes.

The machine learning models were created based on the history of fundamental information and stock´s prices. We use a times series to predict the future price and use this prediction with fundamental indicators in a multiple linear regression that aims to predict the return of the investment.

Thus, the variable target of the model is the return, and the independent variables are the price forecasted, and fundamental metrics such as ROE, revenue, etc.

The model does not claim to be generic so that all investors use the same methodology because once an investment recommendation is widely followed, the company's stock price will quickly become expensive and unviable for purchase. So, readers can adapt the models based on their knowledge and experience.

The stock's prices were extracted from yahoo-finance using python. For the fundamental dataset, one part we extracted manually from https://fundamentus.com.br/ and the other part from the python package called fundamentos. To model, we used 1 year of history and 5 years for backtest. For the remaining strategies that do not require training, we make the 5 years of backtest.

Some companies offer investors models and investment recommendations, and this research aims to help investors as those companies do. For example, the company Smarttinvest (https://smarttinvest.com/carteiras-recomendadas/) works with portfolio recommendations. Recently, this company launched an artificial intelligence that selects stocks and analyses the portfolio's performance. Every 3 months, this artificial intelligence reviews and repurchases the shares. All the process is automated, and the investor does not have to worry about anything. Another company that has a model for investment is the GuiaInvest (https://site.guiainvest.com.br/). This company create the GI Score (Guia Invest Score) and based on fundamental analysis, gives a score for each Brazilian company that has shares on the stock exchange. Thus, investors can use this score to support their decision.

2. LITERATURE REVIEW

This chapter presents the most relevant literature for this work. There is a great amount of literature about investment which includes a lot of different approaches. During the writing of this thesis different studies were read and analysed to understand which approaches should be followed and which type of strategies should be used.

The first step to know how to invest is to understand how a company is managed and what is the main indicators the investors need to pay attention to. These topics are well covered in Epstein (2009), Zanini & Zani (2009), Neto (2017), and Diniz (2015).

After understanding how a company is managed, the investor has the difficult task that is choosing some stocks among several options. For this purpose, there are some methodologies to filter which companies have been better managed and could be a good investment option.

Using fundamental analysis, Graham (2003) and Greenblatt (2010) proposed a formula to select stocks. The Graham proposal is based on the fair share price, according to Graham. It is possible to verify if a stock is cheap or expensive using the Graham number that is calculated based on Earnings per Share (EPS) and Book Value per Share (BVPS). The Graham number is the maximum price that an investor should pay for a stock. Then, an investor only should buy a stock if it has a fair price. However, other great investors, like Warren Buffett, believe that in the long-run investors should not be worried about stock price but should choose companies with good fundamentals (Hagstrom, R. G. (2014)). Warren Buffett said investors should buy a stock because they believe on the business, not because they want the price to go up.

The Greenblatt Magic Formula consists of identifying, in a simple way, companies listed on the stock exchange that have high value and solid fundamentals, but that are being traded at lower prices in the market. The identification of these opportunities is obtained from the creation of a ranking. The raking is composed by the Return on Equity (ROE) and the Price Earnings Ratio (PER).

In addition to these 4 fundamentalist indicators mentioned, there are others that can be used to evaluate the companies. Debastiani & Russo (2017) presents several indicators. It shows how the indicators are calculated and how to interpret the results to make a proper company valuation. Furthermore, it gives a short explanation about macroeconomy and the investment sectors.

In order to test the efficiency of the methods to choose stocks, a backtest was done using python and a historic dataset. Tatsat & Lookabaugh (2020) is a great reference for machine learning for finance using python. It focuses on practice, gives a short explanation about the machine learning models, shows how to create a python code and how to interpret the results. It covers several models such as the autoregressive integrated moving average (ARIMA) and the Long short-term memory (LSTM). For those who do not know how to program in python, we recommend Bhasin (2019) and MacDonald & Blanchette (2019).

The ARIMA model was published by Box and Jenkins in 1976. This model emphasizes the analysis of probabilistic properties to make predictions and can be used to predict the stock price. This methodology consists of adjusting integrated autoregressive models of moving averages. This model was not developed based on economic theory, but statistical techniques were used to create the ARIMA methodology to predict the future values, Hull (2018).

This is not the only model used for price prediction, Tatsat & Lookabaugh (2020) have a deep learning approach to time series modelling, it approaches the LSTM that is a recurrent neural network model and can be used for time series forecasting. The recurrent neural networks are widely used for problems with unstructured sequential data, such as natural language processing and speech

recognition, but their characteristics are interesting for any problem with a sequence character, which is the stock price prediction case.

Another technique used for forecasting time series is the Prophet (Žunić, Korjenić, Hodžić, & Đonko (2020)). This is a procedure based on an additive model where non-linear trends are fit. It has a good performance with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to handle with outliers. For better understanding Lewinson (2020) And Korstanje (2021) can be consulted.

As one of the objectives of this project is to create a model to select stocks based on the return, a model that has a continuous target is required. Tatsat & Lookabaugh (2020) also address this type of model, the multivariate regression model. Multiple regression is a statistical technique to build models that reasonably describe relationships between various explanatory variables of a given process. This kind of model was important to build the final model. Other techniques can be consulted on Cao (2020).

3. Theoretical approach

Initially, it is important to briefly present the valuation and model concepts that will be used in this research. First of all, we explain about valuation and some accounting concepts that help understanding how companies are managed. The goal is not giving an extensive course about valuation, for those who want to want to dive deeper into the subject, we recommend consulting Debastiani & Russo (2017) and Antonik & Damodaram (2017).

Then we explain the formulas used to select shares by Greenblatt and Graham. Finally, we explain the statistical concepts used in the models that we created to select shares.

3.1. VALUATION

Valuation is a methodology that helps us calculate the companies' value. In this way, the entrepreneur can seek investments and even find out what a fair price would be if he ever puts his business up for sale. On the other hand, investors can project the value of the company's shares into the future, estimating the return that it may have if they invest in that company.

For the stock investor, it is important to compare the company's fair value and how much the market is paying for it. So, it is possible to know if the shares are expensive or not.

To understand how valuation works, it is necessary to know some accounting concepts like income statements and balance sheets.

3.1.1. Balance Sheet

A balance Sheet is a report that demonstrates clearly and accurately the financial situation of a company. For such, all business' assets and liabilities are considered, that is, its assets, debts, and profits.

In assets are included cash on hand and receivables, in addition to movable and immovable property. In liabilities we find accounts payable, labour obligations, and long-term debt, for example.

For a clearer understanding of how the Balance Sheet is composed, see an example structure Table 3.1

Balance Sheet			
Assets	Amount	Liabilities	Amount
Current Assets		**Current Liabilities**	
Cash	$	Accounts payable	$
Accounts receivables	$	Short-term notes	$
Temporary investment	$	Interest payable	$
Inventory	$	Accrued payable	$
Prepaid expenses	$	Taxes payable	$
Total Current Assets	$$	**Total Current Liabilities**	$$

Fixed Assets		**Long-term Liabilities**	
Long-term investments	$	Mortgage	$
Land	$	Other long-term liabilities	$
Buildings	$	**Total Long-Term Liabilities**	$$

Plant and equipment	$		
Furniture and fixtures	$	Capital stock	$
Total Net Fixed Assets	$$	Retained earnings	$
		Total Shareholders' Equity	$$

Table 3-1. Balance Sheet

3.1.2. Income Statement

An income Statement is a summary of the company's financial operations in a certain period (usually, it is considered the calendar of 1 year) to make clear whether the company made a profit or loss.

The income statement is basically composed by:

- **Gross Revenue** (or billing): is all entries, everything that was sold (products or services).
- **Net Revenue**: Gross Revenue minus taxes, discounts, and returns.
- **Gross Profit**: considers only variable costs related to production, such as raw material. It is calculated by Gross Revenue - production costs
- **Net Profit**: it considers the remaining costs such as food, rent or mortgage payment, employee salaries, and so on. It is calculated by Gross Profit - Business Costs.

We compare these amounts making the quarter against quarter or year against year to check whether the company is growing or not.

For a better understanding of how the Income Statement is composed, see an example structure Table 3.2:

Income Statement	
Gross revenue	**1,000,000**
(-) Deductions	1,000
(-) Returns	100
(-) Rebates and taxes	50
(-) Taxes	40,000
Net revenue	**958,850**
(-) Costs of goods sold	30,000
Gross profit	**928,850**
(-) Costs of goods sold	30,000
Net profit	**898,850**

Table 3-2. Income Statement

3.1.3. Fundamentalist Multiples

Fundamentalist multiples are indicators extracted from companies reports and market information that can be used as a parameter to identify stock purchase opportunities that are supposedly discounted. Remember that "discounted shares" are those in which the market is trading below the price it is fair value. There are several indicators, but they can be grouped into categories like:

- **profitability index**: in this group, we have for example the net margin, gross margin, and so on.
- **return index:** in this group, we have indicators like Return on equity, return on assets, and so on.

- **liquidity index:** in this, we have current liquidity, general liquidity, and so on.
- **debt index:** in this, we have general debt, third-party capital guarantee, and so on.

As there is an extensive list of indicators, in this research, we only detail the indicators that are used in the models. For those who want to go deeper into the subject, we recommend consulting Debastiani & Russo (2017) and Antonik & Damodaram (2017).

The multiples used in de models are as follows:

- **Net Margin**: that demonstrates whether a company is well managed**.** Thus, knowing the net margin is more than essential for any owner, director, or shareholder of a company. After all, it can be a factor that will improve business performance.
  Companies with a very tight net margin are in danger because the expenses to generate profit are high. Usually, a comparison is made in the same segment because the costs are different. Net Margin is calculated by

$$\textbf{Net Margin} = \frac{Net\ Income}{\text{Net Revenue } (annual)}$$

  As companies report the results quarterly, the annualized net revenue is calculated by summing up the actual net revenue and the 3 last net revenue.

- **Return on Equity (ROE)**: is an indicator that relates a company's profit to its net equity, the main market managers consider a good ROE to be above 15% per year. The formula to calculate the ROE is

$$ROE = \frac{Net\ Income\ (annual)}{net\ equity}$$

- **Earnings Per Share (EPS):** is a quotient that serves as an indicator of an organization's profitability by shares. The earnings per share indicator is one of the most important for determining the fair share price of a company.

$$EPS = \frac{Net\ Profity}{Share\ Quantity}$$

- **Price to Earnings (PE)**: is an indicator that relates the market value of a share to the profit presented. It is calculated using the formula:

$$PE = \frac{Stock\ Price}{EPS}$$

  Some market managers consider it as cheap if PE is lower than 10, as just price if PE is between 10 and 20, and expensive if PE is higher than 20.

- **Current Liquidity (CL)**: is an indicator used to measure a company's capacity, in short term, to pay all its obligations.

$$CL = \frac{Current\ Assets}{Current\ Liabilities}$$

- **Book Value Per Share (BVPS):** indicates the value of a company's equity distributed among the traded shares. The number allows the comparison between market and equity value. Analysing the BVPS, investors can verify whether the price charged in the market is within their expectations. In addition, negotiators have more clarity about the appropriate time to buy and sell the papers.

$$\text{BVPS} = \frac{Net\ Worth}{Shares\ Quantity}$$

- **Debt Per Net Equity (D/E):** calculates the percentage of debt over the company´s net equity. The main market managers consider a good D/E lower than 0.5%.

$$D/E = \frac{Debts}{Net\ Equity}$$

- **Earnings Before Interest and Taxes (EBIT):** is an operating result that encompasses all operating expenses and expenses, as well as, for example, also amortization and depreciation, which are operating costs (they differ from operating expenses because they do not imply cash outflows). This is an indicator widely used, but can be a trap, some analysts use it to assess the cash potential of companies, but the problem is that if the company has a high level of indebtedness and is paying a lot of interest, the EBITD may be good, but the net profit will be bad.
  In this research, we do not show how it is calculated because it is extensive, and companies report the indicator.

3.2. GREENBLATT'S MAGIC FORMULA TO SELECT STOCK

To have a magic formula for choosing the best stocks to invest is most investors around the world's desire. And that is exactly what Joel Greenblatt proposed, the so-called "Greenblatt Magic Formula", which ranks stocks using a set of fundamentalist indicators and allows investors to evaluate and identify the best stock options for investment.

The American Joel Greenblatt is a manager of Hedge Funds and a professor at Columbia Business School. Greenblatt. However, is best known for his book *The Little Book That Beats The Market,* Greenblatt (2010). The simple investment formula created by the academic and investor was introduced in this book.

This method is quite curious and worth knowing. We are going to explain how it works and how to apply this magic formula when choosing the assets that will make up your equity investment portfolio.

Greenblatt's Magic Formula consists of simply identifying companies listed on a stock exchange that have high value and solid fundamentals, but which are trading at the lowest prices in the market. The method creates a ranking to classify the companies.

The first step to run the formula is to delete the companies that give a negative net profit accumulated in 12 months. That is because Greenblatt believes we are not interested in investing in companies with negative net profit.

The formula uses ROE and PE that was explained in the previous chapter, and the formula is expressed by:

$$\textbf{Greenblatt} = \textbf{Score PE} + \textbf{Score ROE}$$

Where the score roe is calculated using the ROE in descent sorting and score PE is calculated using the PE in ascending order as explained below.

After having ROE and PE calculated for all companies, we need to calculate the Score PE and score ROE. And finally sum up these two scores.

Score PE is calculated by sorting the PE from largest to smallest and assigning the weight 1 to the highest PE, 2 to the second highest and so on. For example, we selected 10 stocks, and calculate the score PE as seen in Table 3.3. If the company is expensive, the PE will be higher. So, the technique seeks to remove companies to which the price has risen a lot.

On the other hand, the score ROE is similarly calculated but the weights are reversed, i.e., the smallest roe gets weight 1, the second smallest gets weight 2, and so on. See the example in Table 3.4.

Finally, we sum up the two scores and sort from the highest to the smallest. In our example, the better company is the VALE3, see the Table 3.5.

Stock	Price	PE	ROE	Score PE
MGLU3	18.9	161.79	10.7%	1
WEGE3	36.25	46.31	27.5%	2
YDUQ3	24.85	45.31	5.2%	3
VIVA3	32.27	35.55	17.1%	4
INTB3	28.54	19.64	26.2%	5
B3SA3	14.05	18.38	20.6%	6
ABEV3	16.56	17.89	18.5%	7
ENGI3	15.54	11.28	30.5%	8
VALE3	98.61	5.72	43.7%	9
PETR3	27.3	3.48	28.8%	10

Table 3-3. Score PE

Stock	Price	PE	ROE	Score PE	Score ROE
YDUQ3	24.85	45.31	5.2%	3	1
MGLU3	18.9	161.79	10.7%	1	2
VIVA3	32.27	35.55	17.1%	4	3
ABEV3	16.56	17.89	18.5%	7	4
B3SA3	14.05	18.38	20.6%	6	5
INTB3	28.54	19.64	26.2%	5	6
WEGE3	36.25	46.31	27.5%	2	7
PETR3	27.3	3.48	28.8%	10	8

ENGI3	15.54	11.28	30.5%	8	9
VALE3	98.61	5.72	43.7%	9	10

Table 3-4. Score ROE

Stock	Price	PE	ROE	Score PE	Score ROE	Score Joel
VALE3	98.61	5.72	43.7%	9	10	19
PETR3	27.3	3.48	28.8%	10	8	18
ENGI3	15.54	11.28	30.5%	8	9	17
ABEV3	16.56	17.89	18.5%	7	4	11
B3SA3	14.05	18.38	20.6%	6	5	11
INTB3	28.54	19.64	26.2%	5	6	11
WEGE3	36.25	46.31	27.5%	2	7	9
VIVA3	32.27	35.55	17.1%	4	3	7
YDUQ3	24.85	45.31	5.2%	3	1	4
MGLU3	18.9	161.79	10.7%	1	2	3

Table 3-5. Score Joel

In addition to these calculations, we must consider some important points. Before making the calculations, it is important to eliminate companies that did not make a profit. That is, PE must be greater than zero. And after applying the method and selecting the best companies, the investor should study the best among these companies and avoid investing in companies from the same sector. For the comparisons in this research, we considered just the top 10 companies.

3.3. BEN GRAHAM FORMULA

Benjamin Graham was born on May 9, 1894, in London. After graduating from Columbia University, Graham began working on Wall Street. In the beginning, he was responsible for small services, such as delivering documents and describing the issue of bonds.

However, in a short time he stood out and began to exercise the role of analysing the financial health of companies. Later, at the age of 26, Graham became a partner in the company. Three years later, he decided to leave the company along with Jerry Newman and open the Graham and Newman company in 1926. Two years later, he began teaching investment classes at Columbia University.

In partnership with a former student, David Dodd, Graham wrote the book Security Analysis, which has become a classic in the investment world. He later teamed up with Dodd again to write another classic: The Smart Investor.

Graham has developed a company valuation method based on the intrinsic value of a stock. Intrinsic value can be understood as the fair price of a share. Therefore, this price will not always be equal to the market quotation. In other words, a stock may be quoted at one price and have a different intrinsic value.

This is because this methodology focuses on Valuation, that is, on what the company is worth and not on how much the stock is being traded. As Warren Buffett would say:

"The price is what you pay, the value is what you get."

After calculating and finding the fair price of an asset, investors have on their hands a tool to help them build their investment portfolio. For this, the investor must choose assets to which the current price is lower than the one calculated by him.

However, the intrinsic value analysis of stocks tends to have a certain subjectivity, thus it is not a fixed or standard number. Therefore, it fluctuates within a range of values and over time. Stocks traded below their intrinsic value can offer good buying opportunities, as they offer less risk to the investor.

Benjamin Graham presents the flowing method to calculate the intrinsic value (IV) of a stock:

$$\boldsymbol{IV = \sqrt{22.5\ x\ EPS\ x\ BVPS}}$$

Benjamin Graham defined the criteria that a stock should not have the relation Price per Book Value Per Share (BVPS) higher than 15 and Price per Earnings Per Share (EPS) higher than 1.5. So, if we multiply 15 by 1.5, we have 22.5. Hence, he understands that the relation between EPS and BVPS should not be higher than 22.5.

To have IV formula, observe wthat Graham made:

$$IV = \sqrt{\frac{Price}{BVPS}\ x\ \frac{Price}{EPS}\ x\ EPS\ x\ BVPS} = \sqrt{15\ x\ 1.5\ x\ EPS\ x\ BVPS}$$

In this equation, Grahan is comparing the ideal relation (first part) with the real situation. It is necessary to apply the square root because the price is squared.

However, it is noteworthy that this method does not work for all types of companies, this calculation being more accurate for companies that have constant profits.

3.4. SELIC RATE

The Selic rate, also known as the "basic interest rate", is an acronym for the (*Sistema Especial de Liquidação e Custódia)* Special System for Settlement and Custody, a liquidity control mechanism that the Central Bank uses to determine the amount of money in the economy.

Through the Monetary Policy Committee (Copom), the Central Bank of Brazil decides on the new Selic index. The Copom is a group composed of eight members of the Central Bank that jointly decide the interest rate of the economy.

Currently, in a decision released on December 8, 2021, the Copom raised the Selic rate to 9.25% per year.

In short, the Selic is the rate that determines the cost of borrowing by banks with other institutions and with the Central Bank. Eventually, when there are more cash outflows than inflows, banks run the risk of closing the day with cash on the red. To avoid it, they need to take out loans with other financial institutions or even with the Central Bank – and the Selic rate is used for these operations. If Selic determines the cost that the banks will have with loans, it is easy to imagine that, when it gets higher, they tend to take less credit from other institutions or the Central Bank, since they are more expensive.

As a result, it is normal for banks to transfer this increase to loans or financing granted to individuals and companies.

Investors can also benefit from the increase in the Selic rate. The Brazilian government issues public debt bonds pegged to the Selic rate

3.5. IPCA INDEX

The IPCA (*Índice de Preços ao Consumidor Amplo* - Broad Consumer Price Index) is one of the most traditional and important inflation indexes in Brazil. Created in 1979, the indicator has a simple reason to exist: measuring the price variation of a set of products and services sold in retail and consumed by Brazilian families. The indicator aims to cover 90% of people living in urban areas in the country - and that is precisely why it is called "broad".

The result of the account indicates whether, on average, prices increased, decreased, or remained stable from one month to the next.

The target of the IPCA methodology is families with incomes from 1 to 40 minimum wages, whatever their source of income. To arrive at the inflation rate, prices are collected between the 1st and 30th of each month in stores and service establishments, public service concessionaires (such as water or electricity), in addition to the internet.

The basket of products and services surveyed monthly involves items of different natures. Rice and beans are accounted, of course, but also medical appointments, school fees, electronic devices, and leisure activities. Each one has a greater or lesser weight depending on its presence in the Brazilian's population average consumption basket. Thus, items related to food usually have a greater weight than, for example, communication or clothing.

The IPCA is part of an important monetary policy strategy in Brazil. It is the benchmark for the inflation targeting system, created in 1999. Under this system, the country is committed to adopting strategies to keep inflation within a range periodically fixed by the National Monetary Council (CMN).

In 2020, the target was 4% per year, with a tolerance range of 1.5 percentage points up or down. The target will be considered fulfilled, therefore, if at the end of the year the accumulated IPCA is within the range between 2.5% and 5.5%.

The main tool that the Central Bank must enforce to the inflation target is the interest rate. Therefore, the Selic (basic interest rate of the Brazilian economy) is increased when prices start to rise dangerously. Higher rates tend to make credit more expensive and curb consumption. When prices are under control, the Central Bank has more freedom to reduce interest rates and stimulate the economy.

If the measures are not sufficient and the IPCA ends the year at a level above the target system, the president of the Central Bank must explain himself to the Minister of Finance, indicating what actions will be taken to bring inflation back to the range of tolerance, and in how long.

As an investor, there are many different types of investments that track inflation, especially the IPCA.

3.6. IBOVESPA

The Ibovespa is the main performance indicator for shares traded on B3 and brings together the most important companies in the Brazilian capital market. It was created in 1968 and, over these 50 years, it has established itself as a reference for investors around the world.

Reassessed every four months, the index is the result of a theoretical asset portfolio. It comprises shares of companies listed on B3 that meet the criteria described in its methodology, corresponding to approximately 80% of the number of trades and financial volume in the Brazilian capital market. The Ibovespa index is represented in points of a theoretical portfolio. However, it is not possible to invest directly in Ibovespa points, but it is possible to buy ETF (Exchange Traded Funds). The ETFs are papers that reproduce a certain Index. In this case, the Ibovespa Index ETF is BOVA11, and it is possible to trade it as if it were a stock.

3.7. TIMES SERIES

A time series analysis focuses on extracting statistically significant information from the data, and time series forecasting is applied to create a model that predicts future values, using previously observed values as basis.

When we are working with time series analysis, some important steps must be followed. One of the first and most critical aspects, is plotting the data on a graph from the beginning. This way, some characteristics of the time series can be observed, and the researcher can understand more about the type of data under deliberation. From the graph of the data, one can detect the existence of a trend in the data, which means that the data must be observed, and downward or upward movements overtime must be discerned.

Another important aspect is the presence of seasonality in the data, which means that some patterns repeat themselves over time. This part is important for a variety of forecasting methods, as some of them are not capable of handling seasonality and must be removed before the forecasting procedure.

Outliers are other issues that must be analysed in a time series. The occurrence of outliers is particularly important in regression models and can even change the efficiency of the model. The variance must also be analysed to understand if the data varies over time, or if the variance is constant, which can sometimes be difficult to identify when considering only the plot of the data.

In this project, we decided to use a times series developed by Facebook called Facebook Prophet.

3.7.1. Facebook Prophet

In February 2017, Facebook Research launched an open-source tool for forecasting, the Facebook Prophet. The tool is available on GitHub for use in Python or R (https://github.com/facebook/prophet).

The community of Data Scientists around the world loved the idea, as they were also invited to contribute to the tool. And on top of that, they had the possibility to use a tool developed by Facebook based on their pains and needs that generated learning and a fantastic tool.

Prophet, an open-source software released by Facebook's Core Data Science team, is a procedure designed to predict time series data based on an additive model in which non-linear trends adjust for annual, weekly, and daily seasonality, in addition to the effects of holidays. Works best with time series with strong seasonal effects and multiple seasons of historical data. Prophet is robust to missing data and changes in trends and typically handles outliers well. According to Taylor & Letham (2018), Prophet is used in many apps on Facebook to produce reliable predictions and performs better than any other approach in most cases.

The standard implementation uses a univariate model, where only one variable, time, is used to forecast results. The forecast is achieved as below:

$$\boldsymbol{y(t)} = \boldsymbol{g(t)} + \boldsymbol{s(t)} + \boldsymbol{h(t)} + \varepsilon$$

- y(t) is the target variable, the value that is being predicted
- g(t) is the trend factor.
- s(t) is the seasonality component. It will vary depending upon the periodicity of the data (intra-daily, weekly, and yearly seasonality).
- h(t) is the holidays component. Prophet allows for custom holidays that may impact the model.
- ε is our error term, these are assumed to be normally random distributed variables.

The times series was implemented in python, and we can use it installing the package Prophet using the command pip install Prophet. For those interested in understanding more about the math behind the model, we recommend reading the article Taylor & Letham (2017).

3.8. BACKTEST

Backtest is the testing of a model using historical data.

In the financial market, backtesting refers to testing an investment strategy using past market data, and its objective is to estimate how that strategy would have behaved in a given period in the past. To best achieve this goal, the backtest needs to simulate past market conditions with as much detail as possible.

The analysis of the results of a backtesting must be done with great care, as a superficial analysis can end up choosing a strategy that will not perform well in the future. First, it should always be kept in mind that the objective is to find a good investment strategy for whatever situation, not to select the best strategy in the tested period.

4. METHODOLOGY

As previously described, the main objective of the study is to analyse different types of investment strategies and try to create a strategy using times series, that is, combine a times series model with a linear regression with the goal of scoring companies. To validate the strategies' performances, we made a backtest using a five-year historical dataset.

The strategies tested in this work are: fixed income investment using the Brazilian basic economic rate (Selic) and using the IPCA; variable income investment selecting stocks using the magic formula of Joel Greenblatt, the Ben Graham formula, and investing on the index. Finally, trying to create a model for choosing stocks using machine learning techniques.

All analyses are based on the Brazilian market and the methodology will follow these steps:

- Problem definition
- Explanation of the mathematics concepts
- Dataset preparation
- Modelling
- Results evaluation

4.1. 3.1 PROBLEM DEFINITION

The problem definition is an essential point. It helps to define all the steps of the study. The model and analysis of this study are relevant to the stock market. It is also important for investors and other researchers that are trying to understand better the stock market and trying to have greater return on investments. Additionally, the problem definition also helps delimitate the basis of this work. As there are so many investments strategies in the market, the investors have difficulty to select the best opportunities to invest. Joel Greenblatt and Ben Graham proposed a mathematical equation to help investors to choose stocks, but they were created based on the American market. Then, we are going to validate these methods in the Brazilian market.

4.2. 3.2 EXPLANATION OF CONCEPTS

We present a brief explanation of the main financial and statistical concepts used in this work. We present fundamental concepts of Balance Sheet, Income Statement, indicators such as ROE, PER, etc. Furthermore, we present the magic formula of Joel Greenblatt and the Ben Graham formula which are deterministic models. In the statistical part, we explain the fakebook's prophet algorithm that we use to predict the future price that will be one variable of the model. We also give a brief explanation of linear regression used to create a return model.

4.3. 3.3 DATASET PREPARATION

The dataset preparation is one of the hardest parts of the project because there are few data sources to extract the historical fundamental dataset. Since the goal is to make backtests to analyse the company's performance, we need to extract the historical data. For this purpose, we extract 6 years (Mar/2014 until Mar/2021) of fundamentalist data and 6 years of price. We are going to use 5 years for backtest and 1 year to train models.

First, we extract from the site https://fundamentus.com.br a list of all companies that have stocks in the Brazilian market. We have a total of 506 codes and 350 companies. That is because each company can sell a different type of stock as described in table 4.1. So, the same company can sell more than 1 type of stock. For example, the company Gerdau, has the codes GGBR4 and GGBR3, each paper has a different price in the market, but it is the same company with the same fundamental indicators.

Code	Type	Example
3	Common	VALE3
4	Preferred	GGBR4
5	Preferred Class A	USIM5
6	Preferred Class B	ELET6
11	BDRs, ETF, and Units	BOVA11

Table 4-1. Type of stocks

After having the company´s codes, we try to extract the historical fundamental dataset from the python package *fundamentos* (https://pypi.org/project/fundamentos). However, there are several missing dates. So, we only used the variable quantity of share. The code used to extract the fundamental indicators using the python *fundamentos* package are presented in the appendix 9.2 and 9.3.

For the other variables, we extract all balance sheet and income statement from https://fundamentus.com.br and create the indicators needed to create the models. The extraction was made one by one manually. So, there is no code to reproduce process.

As some companies were not found on the site, it was not possible to do the test using all companies listed on the stock exchanges market.

Furthermore, some companies have few trading on the stock exchange and that is why they were removed from the study because, as will be explained later, every 3 months, the portfolio is rebalanced and these companies do not have trading during the rebalancing period, making it difficult to performance the analysis. The list of companies deleted for having few trading are in the appendix.

At the end, our dataset remains with 429 codes and 312 companies. The final dataset is available in the GitHub (https://github.com/jonavicius-marcio/master).

We also mapped the sector and subsector using the website https://fundamentus.com.br and it is shown in the Table 4.2 and Table 4.3.

Sector	Quantity
Basic materials	48
Industrial goods	68
Communications	7
Cyclic consumption	121
Financial	88
Health	13
Information Technology	11

Non-cyclical consumption	38
Oil, gas, and biofuels	11
Public utility	86

Table 4-2. Shares Sector

Sector	Quantity
Agriculture	6
Cars and motorcycles	3
Chemicals	13
Clothing and footwear fabrics	31
Commerce	23
Commerce and Distribution	13
Computers and equipment	2
Construction	26
Construction and engineering	14
Drinks	1
Electricity	76
financial intermediaries	49
Gas	3
Hotels and Restaurants	3
Housewares	10
Machines and equipment	16
Medical-hospital services analysis and diagnosis	9
Medicines and other products	4
Mining	7
Miscellaneous	17
Miscellaneous financial services	5
Miscellaneous services	7
Oil, gas, and biofuels	11
Others	5
packaging	2
pension and insurance	14
Personal use and cleaning products	5
processed food	13
Programs and Services	8
real estate exploration	16
Steel and metallurgy	16
Telecommunications	7
Transport	19
transport material	12
Travel and leisure	8
Water and sanitation	7
wood and paper	10

Table 4-3. Shares Subsector

The sector is import to diversification. If investor buys 10 stocks from the same sector, and this factor is negatively impacted by some uncontrolled factor, them all the 10 stocks will be impacted as well. In this project we only try to use this variable for the model we try to create. For Grahan formula and Joel formula, we won´t consider the sector to simulation, but as these formulas show the best stocks, the investors should consider taking a look in the sector as well to diversify its stock portfolio. That means, if the investor buy a stock of oil, them he should consider not buying other stocks that depends on the oils as well, like transport.

For extract quotes, we use the python package yfinance. It gives the price by working day of the stocks that we choose, and we can select the range's period. It gives us until 10 years of historical date. We also used this package to extract the quotes of indexes. The code used in this process is available in the appendix 9.1 and 9.4.

For the extraction of the basic economic rate (Selic), we used the Brazilian central bank site, https://www.bcb.gov.br/controleinflacao/taxaselic. In this site we extracted Selic rate since 1996, period when COPOM started to report the target for the SELIC rate for monetary policy purposes. This website has a lot of information about Brazilian monetary policy. We also made it available to the database used in this project on GitHub (https://github.com/jonavicius-marcio/master).

Regarding to the IPCA, we extract from IBGE website. It is the most import Brazilian institute that reports statistics about Brazil. https://www.ibge.gov.br/estatisticas/economicas/precos-e-custos/9256-indice-nacional-de-precos-ao-consumidor-amplo.html?=&t=series-historicas. We also made it available to the database on GitHub (https://github.com/jonavicius-marcio/master).

4.4. 3.4 MODELLING

Modelling process starts with an exploratory analysis. This previous analysis is required to clean the dataset and correct possible mistakes in the process. During this process, we analyse missing data, outliers and understand the behaviour of the continuous and discrete variables.

Then the dataset will be divided between development, training, and testing. This division will be made based on timeline, that is, for the model development we use oldest information, and the most recent information we use validates in order have a backtest.

After, we try to predict the stock price using fakebook's prophet algorithm, it gives us a variable that will be combined with other fundamentalist variables. All these variables will be used in a multiple linear regression model to try to predict the profitability, our target. To validate the performance of the model, we use the square error.

4.5. 3.5 RESULTS EVALUATION

The results are the comparation of investments strategies. That is, we compare the return of investments in fixed income with 2 different techniques (the magic formula of Joel Greenblatt and the Benjamin Graham formula), and investments in index.

To compare the investments strategies, we make a backtest using 5 year of data information. That is, we compare the return of investments in fixed income with 3 the different techniques (the new model created, the magic formula of Joel Greenblatt and the Ben Graham formula). Still in the variable

income, we calculated the return of investments in indexes. We used the Ibovespa, BOVA11 and the IVVB11, the ETF of American stocks, SP&500.

We also compared the return of the same amount invested in a risk-free investment, for this proposal we used 2 types of investments in Brazilian treasure bond. The first one is indexed on the basic economic rate (Selic) and the second is indexed to inflation indicator (IPCA).

5. RESULTS AND DISCUSSION

The result and assumptions of the analyses will be presented in this chapter. Firstly, we will present the results of investments with lower risk, then investments in stocks and, finally, the test result using time series and linear regression.

Before showing the first results, it is important to introduce the concept of portfolio rebalancing that will be used in all strategies. As companies publish the Income Statement and Balance Sheet report every 3 months as a rule imposed by CVM. Then, we set the rule of rebalancing the portfolio every 3 months, after reporting their results. Besides that, we want to simulate an ordinary investor, that is, a worker who saves part of his income and wants to preserve his goods. Then, we assume that every 3 months the investor saves 10 thousand reais, and at every rebalancing of the portfolio, the investor will invest another 10 thousand.

Our simulation starts in Mar/2015, date of reporting of company results, and finish in Mar/2021. Then, during this period, 25 contributions, the investor could save 250 thousand.

Our first strategy was the risk-free investment, the investment treasury bonds using Selic rate, and the second one is also the investment on treasury bonds but using IPCA rate as indexer.

As we can see in Figure 5.1, investing in Selic was more profitable than investing. That happens because Selic rate is higher than IPCA as presented in Figure 5.2. However, investors need to remember that there are investment modalities on the market that pay Selic or IPCA plus a percentage, for example, treasury bond IPCA + 5%.

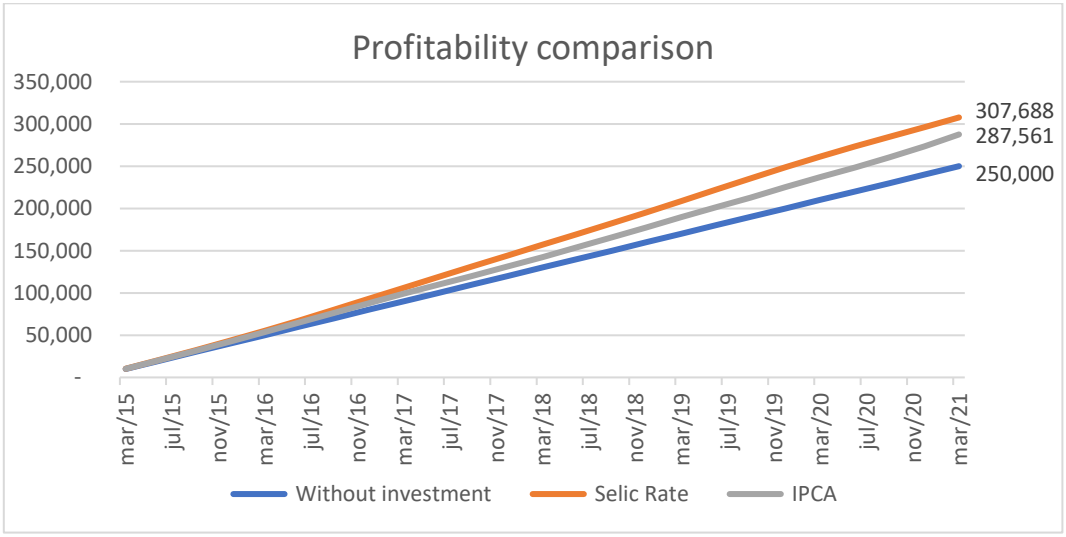

Figure 5-1. Treasury Bonds Strategies

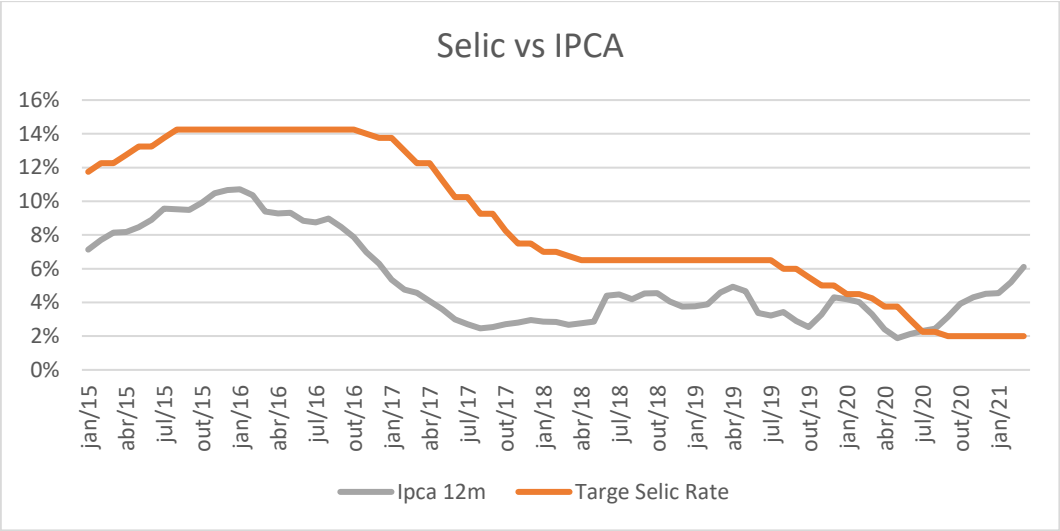

Figure 5-2. IPCA and Selic rates

The third strategy is the Greenblatt's Magic Formula. In this strategy, we will run the formula and make a new investment, a contribution, every 3 months. So, perform the flowing steps:

1. in Mar/2015, delete the companies that have negative net profit accumulated in 12 months because we are not interested in companies with negative net profit.
2. apply the formula explained in the chapter 3.2.
3. select the 10 first companies, that means, the 10 companies with higher score.
4. invest the 10 thousand reais proportionally, in this case, 1 thousand reais in each company.
5. after 3 months, in Jun/2015, we check the actual price of the 10 companies, and calculate the profitability, it is like selling the stocks in market.
6. add to our profitability more 10 thousand reais, and run the process again, but instead of investing 10 thousand in 10 companies proportionally, invest our previous profitability plus 10 thousand reais in the new 10 companies that the model will give us.

Note that some companies can stay in our portfolio. However, companies that have more than one paper can be selected just once. For example, the company Gerdau have the paper GGBR3 and GGBR4. In this case we select just the first paper because, although the price of paper may vary, the company and the ROE and PE are the same.

After applying this method, we have profitability presented in the Figure 5.3.

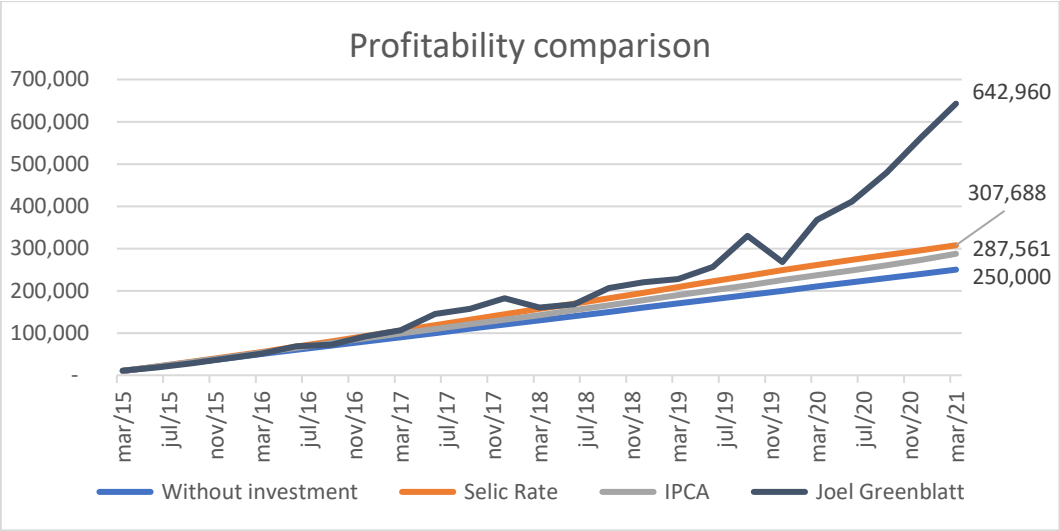

Figure 5-3. Greenblatt's Magic Formula to Select Stock

We can see that the performance of this strategy was better than investments in treasury bonds. However, most of the time, investing in stocks using the Greenblatt's Magic Formula was not much better than investing in treasury bonds. The scenery started to change after Nov/2019. But that was the period in which the Brazilian government started to lower the Selic rate, and several investors started to invest in stock market because investments in treasury bonds were not attractive. The Figure 5.4 shows the comparison between the Selic rate and the Joel's profitability.

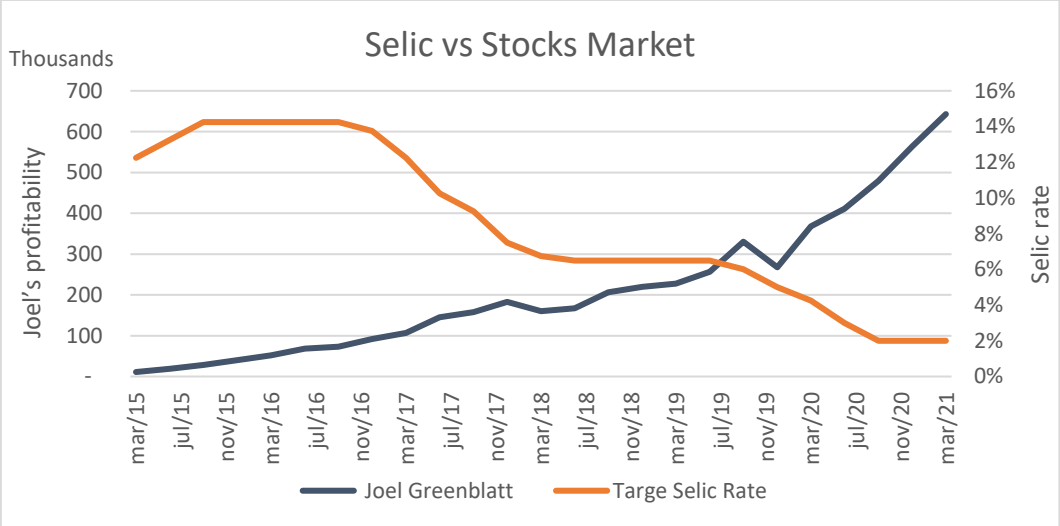

Figure 5-4. Selic rate and Joel's profitability.

Although we ran model 25, and therefore selected 250 companies, as companies may be repeated in each round, during this period only 38 companies entered the model.

The list of companies is shown on Table 5.1 and the python code in the appendix 9.1 and 9.5.

code	company	Amount invested	Profit & Loss	Profit & Loss %
CSNA3	CSN	42,131	39,195	93%
SUZB3	Suzano Papel	35,991	19,841	55%
PTBL3	PORTOBELLO S/A	371,366	120,519	32%
BEEF3	Minerva	61,777	18,451	30%
ATOM3	ATOMPAR	360,787	104,115	29%
WEGE3	WEG SA	161,864	41,367	26%
ASAI3	ASSAI	57,363	9,866	17%
SQIA3	SINQIA	32,042	4,458	14%
AZUL4	AZUL	70,741	9,523	13%
LAME3	LOJAS AMERICANAS S.A.	12,135	1,559	13%
EQPA3	EQTL PARA	482,061	45,488	9%
VULC3	VULCABRAS S/A.	98,663	8,183	8%
ECOR3	ECORODOVIAS	208,440	15,657	8%
CAMB4	PENALTY	3,099	218	7%
GUAR3	GUARARAPES CONFECÇÕES	95,006	6,355	7%
WHRL3	WHIRLPOOL S.A.	443,350	28,515	6%
GRND3	GRENDENE SA	34,750	1,323	4%
ENGI11	ENERGISA	220,726	8,155	4%
BAZA3	BANCO DA AMAZONIA S.A.	7,689	272	4%
LREN3	RENNER	6,149	202	3%
ALPA3	ALPARGATAS	6,149	125	2%
KLBN11	KLABIN	140,185	1,781	1%
DTCY3	DTCOM	88,255	56	0%
ODPV3	ODONTOPREV	488,210	-53	0%
REDE3	REDE EMPRESAS DE ENERGIA ELÉTRICA S.A.	312,184	-1,943	-1%
CURY3	CURY S/A	148,403	-1,720	-1%
CIEL3	CIELO	73,359	-957	-1%
FRIO3	Metalfrio	26,300	-1,548	-6%
TCNO3	TECNOSOLO S/A	55,487	-3,626	-7%
PMAM3	PARANAPANEMA S.A.	33,330	-2,401	-7%
WIZS3	WIZ S.A.	282,507	-26,233	-9%
ABEV3	AMBEV S/A	139,561	-13,280	-10%
PLPL3	PLANOEPLANO	148,403	-16,257	-11%
MRFG3	Marfrig	34,022	-3,792	-11%
TAEE11	TAESA	34,022	-5,849	-17%
SLED4	EDITORA SARAIVA	8,796	-1,619	-18%
POMO3	MARCOPOLO	7,830	-1,477	-19%
CCRO3	COMPANHIA DE CONCESSÕES RODOVIÁRIAS	48,967	-11,510	-24%

Table 5-1. Companies selected using the Joel Formula and its Profit and Loss

The fourth strategy uses the Benjamin Graham formula. The method of making the portfolio rebalancing is like Joel's. However, Graham's formula does not give us a score, it calculates the intrinsic value. So, we calculate the percentual difference between the market value and intrinsic value. Companies that have lower (more negative) percentual difference are cheaper. As a result, we selected the 10 cheapest companies. Another difference is that we only change companies if the percentual difference is higher than -30%.

1. In Mar/2015, delete the companies that have negative net equity and eps.
2. apply the formula explained in the chapter 3.3.
3. calculate the percentual difference between the market value and intrinsic value
4. select the 10 companies that have lower percentual difference.
5. invest the 10 thousand reais proportionally, in this case, 1 thousand reais in each company.
6. After 3 months, in Jun/2015, check the actual price of the 10 companies, and calculate the profitability, it is like selling the stocks in market.
7. add to our profitability more 10 thousand reais, and run the process again, but only change companies if the percentual difference is higher than -30%.
8. instead of investing 10 thousand in 10 companies proportionally, invest our previous profitability plus 10 thousand reais in the new 10 companies that the model will give us.

Note: The method was created in an industrial period and the companies had great patrimonies. Currently, technology companies can have low equity and high profitability. Therefore, the method does not work for companies that have a very small equity value and, consequently, generate a lot of cash.

As the method does not work for every type of company, we only select the shares that have net equity and eps positive.

As we can see in Figure 5.5, the Graham method so far has shown the best results. In addition, from Sep/16 this strategy started to show better results than an investment in fixed income.

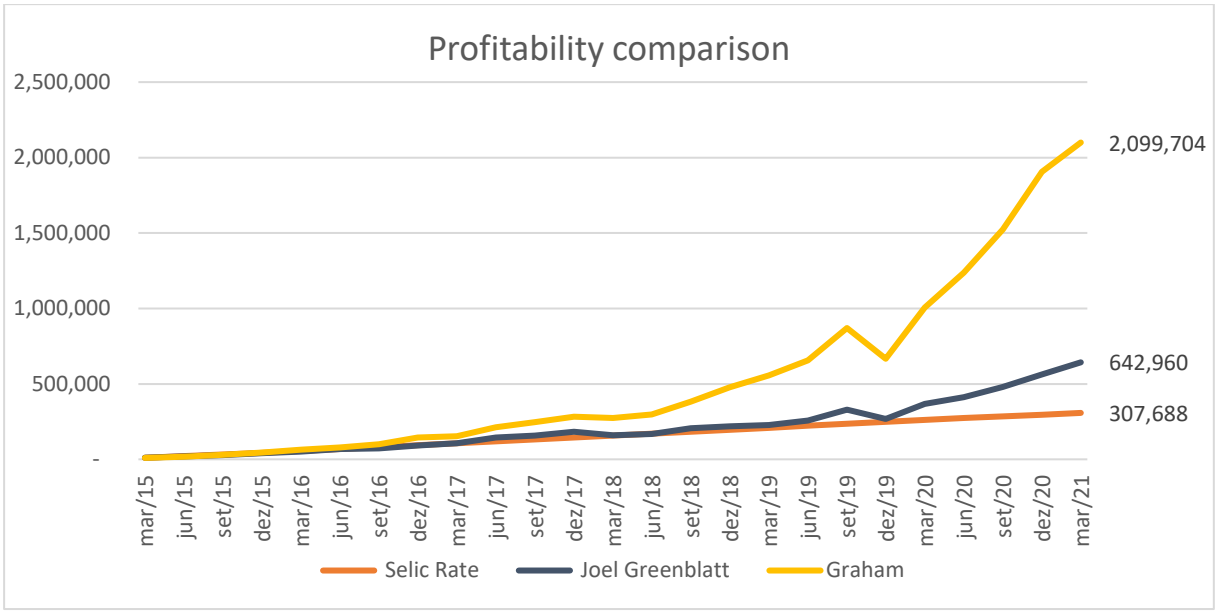

Figure 5-5. Grahan's profitability compared with other strategies.

This method had a smaller change of companies than the Joel method. During the 5 years, only 22 companies were selected. The list of companies and their profits and loss are presented on Table 5.2. The python code used added in the appendix 9.5.

code	company	Amount invested	Profit & Loss	Profit & Loss %
PCAR3	PÃO DE AÇÚCAR	344,978	390,448	113%
NUTR3	NUTRIPLANT	381,883	273,611	72%
LCAM3	LOCAMERICA	121,997	41,347	34%
PRIO3	PETRORIO	1,142,895	357,117	31%
BAHI3	BAHEMA	35,320	7,034	20%
EMAE4	EMAE	418,783	78,258	19%
MGLU3	MAGAZ LUIZA	1,096,414	161,783	15%
SHUL4	SCHULZ	1,148,611	161,462	14%
CAMB3	PENALTY	638,833	85,434	13%
ENEV3	ENEVA	938,016	124,359	13%
GUAR3	GUARARAPES CONFECÇÕES	210,595	26,327	13%
REDE3	REDE EMPRESAS DE ENERGIA ELÉTRICA S.A.	15,162	1,738	11%
BRGE12	CONSORCIO ALFA	16,876	1,817	11%
TRPL3	TRANSMISSÃO PAULISTA	1,126,018	52,836	5%
EQTL3	EQUATORIAL ENERGIA S.A.	1,147,611	51,388	4%
CSMG3	COPASA MG	5,716	240	4%
PNVL3	PANVEL FARMÁCIAS	1,148,611	41,340	4%
CPLE3	COPEL	1,148,611	39,743	3%
EALT4	ELECTRO AÇO ALTONA S/A	8,430	-459	-5%
OIBR3	OI	127,795	-12,838	-10%
SGPS3	Springs	259,985	-32,659	-13%
CLSC4	CELESC	2,966	-621	-21%

Table 5-2. Companies selected using the Graham Formula and its Profit and Loss

Another strategy is to invest in index, in this work we make a backtest using the 2 main index, the Ibovespa represented by code BOVA11 and the S&P500, the index of the most important stocks in the USA. In Brazil, it's possible to invest in the S&P500 through an ETF, the IVVB11.

We can see in Figure 5.6 that IVVB11 had a performance close to the Joel strategy and both Joel and Graham strategies are better than the Ibovespa index. The python code used to this simulation is presented in the appendix 9.5.

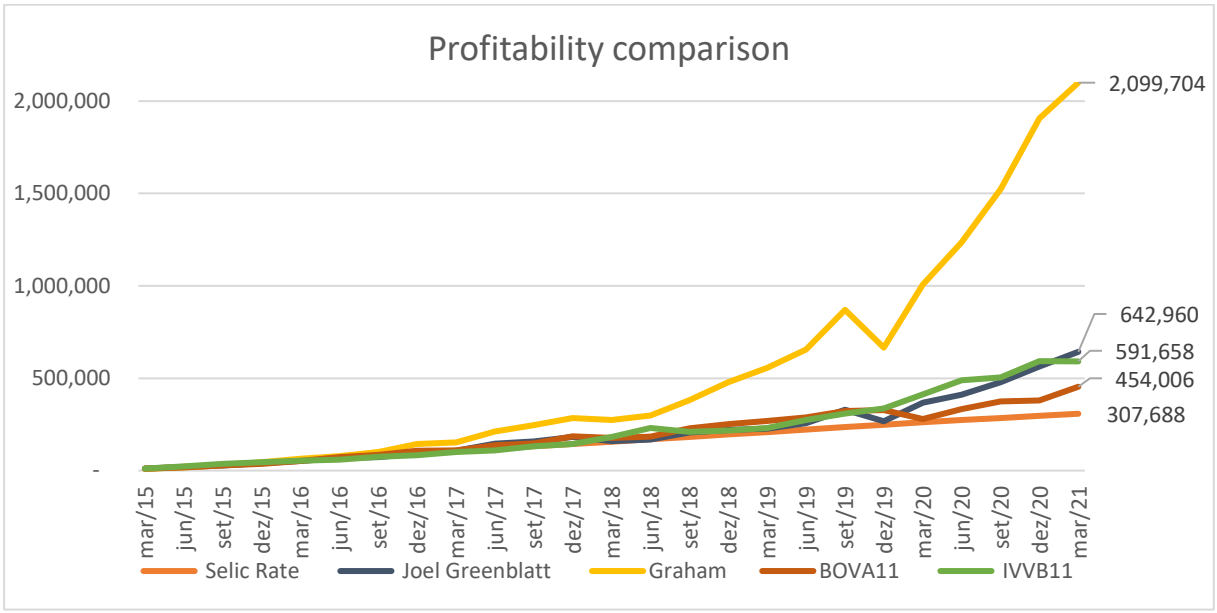

Figure 5-6. Index's profitability compared with other strategies.

If we delete the Graham line, Figure 5.7 we can see better the comparison of performances. We can observe that in Mar/2020, there was a drop in the Ibovespa index caused by covid. Although American stocks also fell, the IVVB11 did not fall because there was a devaluation of the real against the dollar.

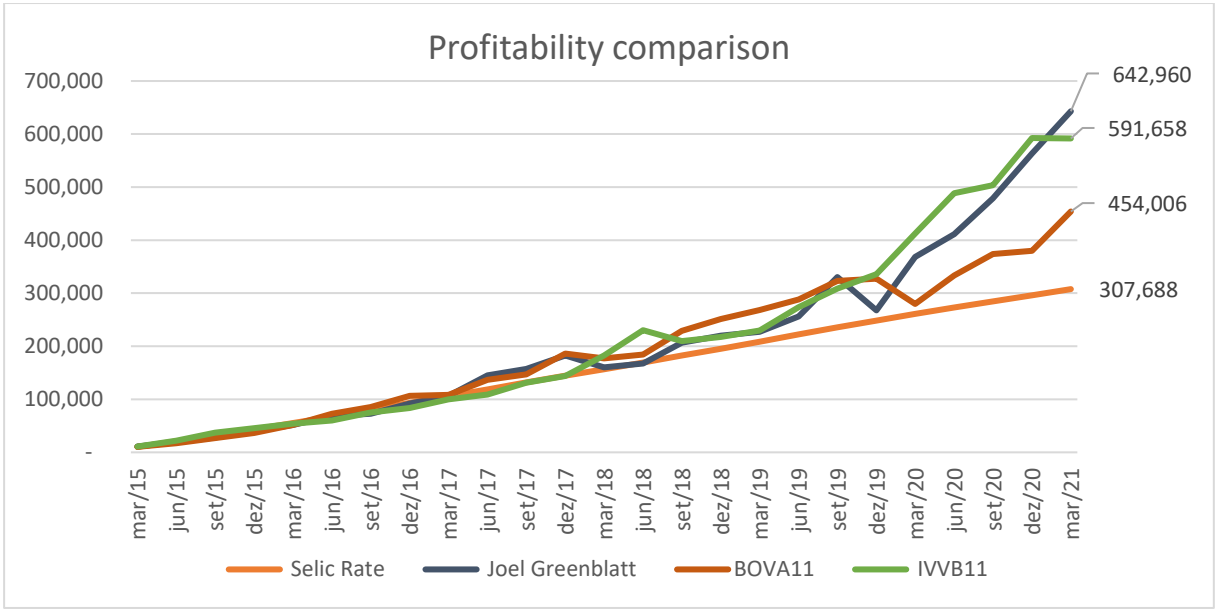

Figure 5-7. Index's profitability compared to other strategies.

The last strategy used in this project is the stock selection model using machine learning. In this model, we used the Facebook time series to forecast the stock price. For example, in Mar/2015, we needed to decide in which stock we wanted to invest. So, we used the data from Mar/2014 until Mar/2015 to predict the stocks prices on Jun/2015, data of portfolio rebalancing. This prediction along with fundamentalist variables of Mar/2015 is used in a linear regression that aims to predict the return. Then, we select the 10 first companies to invest our money as soon as the Joel and Graham simulation was made. At each round of contribution, the model's betas are updated with the information we have so far.

The first model was trained using data of 2014 and as we can see on the Table 5.3, only the variable net_profit and net_margin was significant to predict the return. Unfortunately, the variable prediction created using the Facebook Profet was not good to predict the return. The code and more results of this model is shown in the appendix 9.6 and 9.7.

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.1480	0.451	0.328	0.743	-0.737	1.033
total_assets	0.0310	0.134	0.232	0.817	-0.232	0.294
shares	0.0540	0.122	0.443	0.658	-0.185	0.294
net_profit	0.8468	0.432	1.959	0.050	-0.002	1.695
growth_profit	0.1273	0.243	0.524	0.601	-0.350	0.605
net_equity	-0.1457	0.296	-0.492	0.623	-0.727	0.435
net_revenue_12	0.2826	0.256	1.102	0.271	-0.221	0.786
net_profit_12m	0.0540	0.315	0.171	0.864	-0.564	0.672
cl	-0.0137	0.106	-0.129	0.897	-0.221	0.194
roe	0.1174	0.299	0.392	0.695	-0.470	0.705
net_margin	-1.0904	0.244	-4.470	0.000	-1.569	-0.612
eps	0.0664	0.170	0.390	0.697	-0.268	0.401
bvps	0.0506	0.132	0.383	0.702	-0.209	0.310
d_e	-0.1135	0.256	-0.443	0.658	-0.616	0.389
close	0.3292	0.356	0.926	0.355	-0.369	1.027
volume	0.0515	0.149	0.346	0.730	-0.241	0.344
prediction	-0.5689	0.452	-1.260	0.208	-1.455	0.318

Table 5-3. OLS Linear Regression

After selecting the stocks and running the backtest simulation of 5 years, the model selected 37 companies, as presented in the Table 5.4. The final performance is presented in the Figure 5.8, unfortunately, the model did not defeat the formula of Graham and Joel, it was very close to BOVA11's performance.

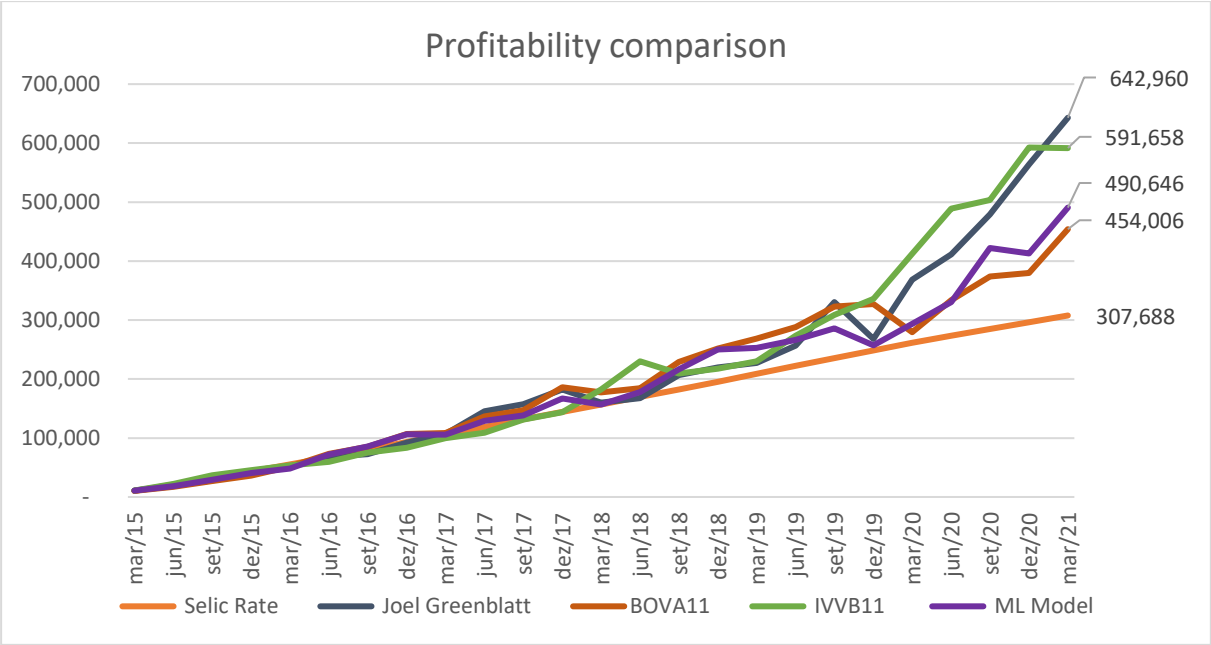

Figure 5-8. Machine Learning profitability compared with other strategies.

Code	Company	Amount invested	Profit & Loss	Profit & Loss %
B3SA3	B3	29,547	7,302	25%
CYRE3	CYRELA BRAZIL REALTY	34,017	8,124	24%
MRFG3	Marfrig	52,973	11,584	22%
GGBR3	GERDAU S.A.	42,294	7,729	18%
CPFE3	CPFL ENERGIA S.A.	26,727	4,824	18%
BRAP3	BRADESPAR S/A	14,814	2,402	16%
ELET3	ELETROBRÁS	169,926	27,345	16%
SUZB3	Suzano Papel	93,270	12,935	14%
SULA11	Sul America	34,017	4,029	12%
GPAR3	CELGPAR	9,563	1,063	11%
SBSP3	SABESP	41,538	4,375	11%
JBSS3	JBS	223,122	23,491	11%
VALE3	VALE	384,484	40,370	10%
BRKM3	BRASKEM	179,171	16,357	9%
LIGT3	LIGHT SA	27,612	2,157	8%
BBAS3	BANCO DO BRASIL S.A.	449,499	21,743	5%
SANB11	SANTANDER	451,203	20,329	5%
CMIN3	CSNMINERACAO	42,294	1,887	4%
TRPL3	TRANSMISSÃO PAULISTA	8,179	281	3%
KLBN11	KLABIN	10,897	290	3%
CSNA3	CSN	173,841	4,513	3%
ITUB3	ITAUUNIBANCO	502,268	11,857	2%
ITSA3	ITAÚSA	379,565	8,662	2%
CPLE3	COPEL	30,345	582	2%
ABEV3	AMBEV S/A	407,932	4,971	1%
BBDC3	BANCO BRADESCO S.A.	502,268	4,250	1%
HYPE3	HYPERA	5,060	13	0%
VIVT3	TELEF BRASIL	151,907	-1,178	-1%
BPAC11	BTGP BANCO	59,919	-1,056	-2%
CIEL3	CIELO	50,224	-1,738	-3%
CMIG3	CEMIG	27,259	-1,043	-4%
BRFS3	BRF Foods	3,910	-176	-4%
OIBR3	OI	17,713	-899	-5%
PETR3	PETROBRAS	300,694	-16,310	-5%
CCRO3	COMPANHIA DE CONCESSÕES RODOVIÁRIAS	19,749	-1,197	-6%
USIM3	USIMINAS	50,065	-7,047	-14%
PDGR3	PDG REALT	14,814	-2,545	-17%

Table 5-4. Companies selected using the Machine Learning and its Profit and Loss

6. CONCLUSIONS

After analysing the results obtained with the backtests, we can conclude that investments in variable income, although they are riskier, in the long run they give a greater return in comparison to the investments in fixed income.

Although Graham's methodology is considered outdated by some analysts, we could observe that for the Brazilian market, the model still performs very well. Furthermore, this model has the best performance.

Joel's model and indexes had similar results most of the time. The results became different after covid when the scenario changed. There was a rise in the dollar exchange rate, fall in the quote of some companies and rise in others.

Regarding to the fixed income investments, in the long run, it had a worse performance in comparison to the variable income. The great difficulty in variable investments is the selection of assets and the backtest showed that the Joel and Graham model are good for the selection of assets.

Unfortunately, the time series model was not efficient for forecasting stock prices, and the machine Learning model did not defeat the formula of graham and Joel. But as described in the next chapter, there is still opportunity for future work.

The ideal is to build a balanced portfolio, that is, a portfolio that contains a little fixed income strategy and a little variable income. Although fixed income has a lower income, it also has a lower risk and, therefore, it would be interesting to keep part of the assets in fixed income.

Nowadays, Brazil is in a scenario of high inflation (high IPCA) and the government is raising the Selic rate to control it. This has caused the share price to fall.

It is worth noting that past earnings are no guarantee of future earnings and that this work is not intended to indicate any type of asset purchase.

7. LIMITATIONS AND RECOMMENDATIONS FOR FUTURE WORKS

There are several types of investment strategies and predictive models. In this work, only a few were tested. To find intrinsic value, other models can be tested like:

- Weighted average cost of capital (WACC)
- Discounted Cash Flow Analysis (DCF)
- Gordon Model
- Valuation based on the company's assets and liabilities.

We work just with the Brazilian market, and this work could be extended to other countries' markets, then we could avoid the country risk using a diversification strategy in other markets.

For price prediction, we could test other models like Arima, AutoArima, LSTM, AdaBoost, CastBoost and so on.

Instead of using the regression multiple, we could try using other techniques like decision tree, xgboost, neural network and so on.

8. BIBLIOGRAPHY

Tatsat, H., Puri, S., & Lookabaugh, B. (2020). Machine Learning & Data Science Blueprints for Finance from Building Trading Strategies to Robo-Advisors Using Python. First edition.

Graham, B. (2003). *O investidor Inteligente. O Guia Prático de como ganhar dinheiro na Bolsa, Comentários de Jason Zweig*. *Rev Ed. pp 445.*

Debastiani, C. A., & Russo, F. A. (2017). *Avaliando empresas, investindo em ações: a aplicação prática da análise fundamentalista na avaliação de empresas.* First edition.

Hull, J. C. (2018). Risk Management and Financial Institutions. Fifth edition

Kobori, J. (2018). *Análise fundamentalista Como Obter uma performance superior e consistente no mercado de ações*. Second edition

Antonik, L. R., Müller, A. N., & Damodaram, P. A., (2017*). Avaliação de Empresas (Valuation) para leigos.* First edition.

Žunić, E., Korjenić, K., Hodžić, K., & Đonko, D. (2020). Application of facebook's prophet algorithm for successful sales forecasting based on real-world data. https://arxiv.org/ftp/arxiv/papers/2005/2005.07575.pdf

Cao, L. (2020). AI in Finance: A Review. SSRN https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3647625

Greenblatt, J. (2010). The Little Book That Still Beats the Market. First edition.

Wrenn, C., & Pecaut, P. (2019). Into the Minds of Warren Buffett and Charlie Munger. First edition.

Zanini, F. & Zani J., (2009). *Curso Básico de Finanças*. First edition.

Epstein L., (2009). Reading Financial Reports for Dummies. First edition.

Diniz, N., (2015). *Análise das demonstrações financeira*. First edition.

Neto, A. A., (2017). *Valuation, medidas de criação de valor, gestão baseada em valor, avaliação de empresas*. Second edition.

MacDonald, B. & Blanchette, M. (2014). Python for Finance. First Edition.

Bhasin, A., (2019). Python Basics: A Self-Teaching Introduction

Lewinson, E. (2020). Python for Finance Cookbook. Over 50 recipes for applying modern Python libraries to financial data analysis. First Edition, pp 84-89.

Korstanje J. (2021). Advanced Forecasting with Python, With State-of-the-Art-Models Including, LSTMs, Facebook's Prophet, and Amazon's DeepAR. First Edition, pp 253 – 272.

Hagstrom, R. G. (2014) – The warren Buffet way. Second edition, pp 45-70.

Taylor, S. J., & Letham, B. (2018). Forecasting at scale. The American Statistician, 37-45. https://doi.org/10.1080/00031305.2017.1380080

Taylor, S. J., & Letham, B. (2017). Forecasting at scale. https://peerj.com/preprints/3190.pdf

9. APPENDIX

9.1. LIST OF COMPANIES DELETED

AFLT3	BRGE7	CEPE3	EEEL4	LINX3	NAFG3	PTNT3	SOND3	VSPT3
AMIL3	BSLI3	COCE6	ELEK3	LUXM3	NAFG4	PTNT4	SPRI3	VSPT4
APTI4	BSLI3	CORR3	ELEK4	MEND5	NATU3	RANI4	SPRI5	
BAHI11	BSLI4	CORR4	ELPL3	MGEL3	ODER3	RLOG3	SPRI6	
BBSE3	BTTL3	CPRE3	FIGE3	MMAQ3	ODER4	SAPR11	TIET11	
BMGB11	CALI3	CRUZ3	FIGE4	MMAQ4	PCAR4	SAPR3	TIET3	
BOBR3	CALI4	CTSA8	IDVL3	MRSA5B	PINE3	SAPR4	TIET4	
BPAR3	CCXC3	EEEL3	IDVL4	MRSA6B	POWE3	SMLS3	VIVT4	

9.2. IMPORT USED IN MOST OF THE CODES.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import time
```

In [2]:
```
pd.set_option('display.max_rows', 200)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
```

In [3]:
```
pd.set_option('display.float_format', '{:.2f}'.format)
```

In [4]:
```
#! pip install fundamentos
```

In [5]:
```
#!pip install -q yfinance
```

In [6]:
```
import fundamentos as ftos
```

In [7]:
```
from pandas_datareader import data
import plotly.express as px
import yfinance as yf
```

9.3. CODES USED TO EXTRACT DATA FROM PYTHON FUNDAMENTOS PACKAGE

This code extract data of several companies and it takes so long to run.

1) extract the companies results

In [8]:
```
companies= pd.read_csv("companies.csv", sep=";", encoding = 'latin_1')
```

In [9]:
```
len(companies)
```

Out[9]:
```
506
```

In [7]:
```
date_set= pd.DataFrame()
for company in range(len(companies)):
    year_min=2011
    year_max=2021
    while year_min <= year_max:
        for quarter in range(4):
            try:
                df = ftos.get_fundamentos(companies['empresa'].loc[company], year=year_min, quarter=quarter+1)
                df['company']=companies['empresa'].loc[company]
                date_set = pd.concat([date_set, df])
            except:
                pass
        year_min=year_min+1
    print(company)
```

In [8]:
```
date_set.head()
```

Out[8]:

			Caixa		Dividendos									Dívida	
	Aplicações Financeiras	Dinheiro em Caixa	Disponibilidades	DY	Dividendos e JCP	Payout	DB/PL	DL/EBITDA	Dívida Bruta	Dívida Líquida	Dívida em Moeda Estrangeira	EV	EV/EBIT	Endividamento Financeiro	F
Data															
2019-12-31	-90760.0	2158	-88602	NaN	NaN	NaN	2.49	NaN	238706	327308	310884.0	NaN	NaN	0.7134	2838
2020-03-31	2158.0	8669	10827	NaN	NaN	NaN	0.77	NaN	89048	78221	88731.0	NaN	NaN	0.4337	2526
2020-06-30	2158.0	15340	17498	NaN	NaN	NaN	0.61	NaN	88708	71210	88732.0	NaN	NaN	0.3792	1731
2020-09-30	-46284.0	54880	8596	NaN	NaN	NaN	1.11	NaN	238842	230246	275674.0	NaN	NaN	0.5270	5045
2020-12-31	-58038.0	30376	-27662	NaN	NaN	NaN	0.26	NaN	60228	87890	339915.0	NaN	NaN	0.2074	-559

In [10]:
```
date_set.to_csv('fundamentalist_dataset.csv')
```

9.4. PRICE EXTRACTION

1) Price

In [8]:
```
companies= pd.read_csv("companies.csv", sep=";", encoding = 'latin_1')
```

In [29]:
```
date_quote= pd.DataFrame()
for company in range(len(companies)):
    try:
        comp=companies['empresa'].loc[company]
        df =yf.download(comp+".SA", start='2015-01-01')
        df['company']=companies['empresa'].loc[company]
        date_quote = pd.concat([date_quote, df])
    except:
        pass
    print(company)
```

In [32]:
```
date_quote.to_csv('date_quote.csv')
```

9.5. MODELS

1) clean the date quote data set

```
date_quote= pd.read_csv("date_quote.csv", sep=";", encoding = 'latin_1')
```

```
# Adjust date
```

```
date_quote['date_month']=date_quote['date'].str[3:5]
date_quote['date_year']=date_quote['date'].str[6:10]
date_quote['date']=date_quote['date'].str[6:10]+date_quote['date'].str[3:5]+date_quote['date'].str[0:2]
```

```
date_quote_clean=date_quote[['date', 'company' , 'date_month', 'date_year', 'close' ]]
```

```
date_quote_clean = date_quote_clean.reset_index()
```

```
date_quote_clean=date_quote_clean.drop(columns=['index'])
```

```
date_quote_clean['date_month']=date_quote_clean['date_month'].astype(int)
date_quote_clean['date_year']=date_quote_clean['date_year'].astype(int)
```

```
# add
date_quote_manual= pd.read_csv("date_quote_manual.csv", sep=";", encoding = 'latin_1')
```

```
date_quote_end=pd.concat([date_quote_clean, date_quote_manual])
```

```
date_quote_end['date']=date_quote_end['date'].astype(int)
```

2) Contribution

```
date_quote_end=date_quote_end.sort_values(by=['company', 'date'], ascending=False)
```

```
#dropping ALL duplicate values
date_quote_end_1=date_quote_end.drop_duplicates(subset=["company", 'date_month','date_year'], keep = 'first').reset_index()
```

```
def create_contributions(df):
    df['contributions']=0
    contributions=0
    for year in range(2015,2022):
        for month in range(3, 13, 3):
            contributions=contributions+1
            # Condition
            conditions = [
                (df['date_year'] == year) & (df['date_month'] == month)]
            choices = [contributions]
            df['contributions'] = np.select(conditions, choices, default=df['contributions'])
    return df
```

```
date_quote_end_2=create_contributions(date_quote_end_1)
```

```
date_quote_end_2.to_csv('date_quote_end_2.csv')
```

3) Create EPS (earning per share)

```
df_results= pd.read_csv("dateset_fundamentalist.csv", sep=";", encoding = 'latin_1')
```

```
df_results['eps']=df_results['eps'].astype('float')
```

```
# create the quarter
```

```
date_quote_filter=date_quote_end.sort_values(by='date', ascending=False)
```

```
date_quote_filter=date_quote_filter.drop_duplicates(subset=['company', 'date_month', 'date_year'], keep='first')
```

```
date_quote_filter=date_quote_filter[['company', 'date_month', 'date_year', 'close']]
```

```
df_results_price=pd.merge(df_results,date_quote_filter,how='left',
                          left_on=['company','date_year','date_month'],right_on=['company','date_year','date_month'])
```

```
#  MERGE indicators with price.
# Keep the companies we found the price
```

```
df_results_price_2=df_results_price[df_results_price['close'].notna()].reset_index()
```

```
df_results_price_2['pe']=df_results_price_2['close']/df_results_price_2['eps']
df_results_price_2['pe'] = df_results_price_2['eps'].apply(lambda x: 0 if x == 0 else x )
```

```
# delete duplicated companies (eg CGRA3 and CGRA4)
df_results_price_2=df_results_price_2.sort_values(by=['company'])
df_results_price_clean=df_results_price_2.drop_duplicates(subset=['name', 'date', ], keep='first')
```

9.5.1. Magic Formula of Joel

4) Magic Formula Of Joel

```
def scoring(df, metric, ascend):
    df=df.sort_values(by=[metric], ascending=ascend).reset_index()
    df=df.drop(columns=['index'])
    var_score='score_'+metric
    df[var_score]=df.index
    return df
```

```
def formula_joel():
    date_joel= pd.DataFrame()
    for year in range(2015,2022):
        for month in range(3, 13, 3):
            df_selec=df_results_price_clean[(df_results_price_clean['date_year']==year) &
                                            (df_results_price_clean['date_month']==month)]
            # we select only positive net_profit
            df_selec=df_selec[df_selec['net_profit_12m']>0]
            #score_pe
            df_selec=scoring(df=df_selec, metric='pe', ascend=False)
            # score_roe
            df_selec=scoring(df=df_selec, metric='roe', ascend=True)
            # score_joel
            df_selec['score_joel']=df_selec['score_roe']*df_selec['score_pe']
            df_selec=df_selec.sort_values(by=['score_joel'], ascending=False).reset_index()
            ##
            df_selec=df_selec.drop(columns=['index','level_0' ])
            df_selec=scoring(df=df_selec, metric='score_joel', ascend=False)
            df_selec=df_selec.rename(columns={'score_score_joel':'chosen'})
            date_joel = pd.concat([date_joel, df_selec])
    return date_joel
```

```
df_joel=formula_joel()
```

```
df_joel_1=create_contributions(df_joel)
```

Calcute the profit

```
date_quote_end_3=date_quote_end_2.rename(columns={'close':'close_end'})
date_quote_end_3['contributions']=date_quote_end_3['contributions']-1
date_quote_end_3=date_quote_end_3[['company', 'close_end','contributions']]
```

Joel all the cicle

```
joel_performance = pd.DataFrame({'contributions': range(1,26), 'profit_total':range(1,26)})
date_detail_joel= pd.DataFrame()

investiment_month=10000
investiment=0
```

```
for contributions in range(26):
    contributions=contributions+1

    investiment=investiment+investiment_month
    investiment_per_share=investiment/10

    df_joel_aux=df_joel[(df_joel['contributions']==contributions)]
    df_joel_aux_1=pd.merge(df_joel_aux, date_quote_end_3, how='left', left_on=['company', 'contributions'],
                           right_on=['company','contributions'])

    df_joel_aux_1['no_shares']=investiment_per_share/df_joel_aux_1['close']
    df_joel_aux_1['profit']=df_joel_aux_1['no_shares']*(df_joel_aux_1['close_end'])

    # selecet the 10 TOP companies
    top_selected=df_joel_aux_1[df_joel_aux_1['chosen'].isin([0,1,2,3,4,5,6,7,8,9])]
    date_detail_joel= pd.concat([date_detail_joel, top_selected])
    investiment=top_selected['profit'].sum()

        # Condition
    conditions = [
        joel_performance['contributions']==contributions]
    choices = [investiment]
    joel_performance['profit_total'] = np.select(conditions, choices, default=joel_performance['profit_total'])
```

```
date_detail_joel.to_csv('date_detail_joel.csv')
```

9.5.2. Graham Formula

5) Graham Formula

```
df_graham=df_results_price_clean.copy()
```

```
df_graham=create_contributions(df_graham)
```

```
df_graham['bvps']=df_graham['bvps'].astype(float)
```

```
df_graham=df_graham[df_graham['net_equity']>0]
df_graham=df_graham[df_graham['eps']>0].reset_index()
```

```
df_graham['vi']=(22.5*df_graham['eps']*df_graham['bvps'] )**(1/2)
```

```
df_graham['vi_percent']=df_graham['close']/df_graham['vi']-1
```

```
df_graham=df_graham.drop(columns=['index'])
df_graham=df_graham.drop(columns=['level_0'])
```

```
# Calculate grahan performance
```

```
grahan_performance = pd.DataFrame({'contributions': range(1,26), 'profit_total':range(1,26)})
date_detail_grahan= pd.DataFrame()
companies_keeped = pd.DataFrame({'company': list('a')})

investiment_month=10000
investiment=0
top=10
```

```
for contributions in range(26):
    investiment=investiment+investiment_month
    investiment_per_share=investiment/10

    df_graham_selected=df_graham[(df_graham['contributions']==contributions)].reset_index(drop=True)
    df_graham_selected=scoring(df=df_graham_selected, metric='vi_percent', ascend=True)
    df_keeped= df_graham_selected[df_graham_selected['company'].isin(companies_keeped['company'])]

    df_graham_selected_clean = df_graham_selected[~df_graham_selected['company'].isin(companies_keeped['company'])]
    .reset_index(drop=True)
    df_new = df_graham_selected_clean[df_graham_selected_clean.index.isin(range(top))]

    df_graham_selected_1=pd.concat([df_new, df_keeped]).reset_index(drop=True)
    # profit
    df_graham_selected_2=pd.merge(df_graham_selected_1,date_quote_end_3,how='left',left_on=['company','contributions'] ,
                                  right_on=['company','contributions'])
    df_graham_selected_2['no_shares']=investiment_per_share/df_graham_selected_2['close']
    df_graham_selected_2['profit']=df_graham_selected_2['no_shares']*(df_graham_selected_2['close_end'])

    investiment=df_graham_selected_2['profit'].sum()


    # selecet the 10 TOP companies
    date_detail_grahan= pd.concat([date_detail_grahan, df_graham_selected_2])

    # Condition
    conditions = [
        grahan_performance['contributions']==contributions]
    choices = [investiment]
    grahan_performance['profit_total'] = np.select(conditions, choices, default=grahan_performance['profit_total'])

    contributions_next=contributions+1
    df_graham_next=df_graham[df_graham['contributions']==contributions_next].reset_index()


    df_graham_next['contributions']=df_graham_next['contributions']-1
    df_graham_next=df_graham_next.rename(columns={'vi':'vi_end', 'vi_percent':'vi_percent_end'})
    df_graham_next=df_graham_next[['company', 'contributions', 'vi_end', 'vi_percent_end']]

    df_graham_selected_next=pd.merge(df_graham_selected_2,df_graham_next,how='left',left_on=['company','contributions'],
                                     right_on=['company','contributions'])

    df_graham_selected_next_1=df_graham_selected_next[df_graham_selected_next['vi_percent_end']<=-0.3]

    companies_keeped=df_graham_selected_next_1[['company']]

    top=10-len(df_graham_selected_next_1)
```

```
date_detail_grahan.to_csv('date_detail_grahan.csv')
```

9.5.3. Selic

6) Investment in Fixed income

6.1) Selic

```
meta_selic= pd.read_csv("meta_selic.csv", sep=";")
```

```
def change_date_format(df):
    df['date_month']=df['date'].str[3:5]
    df['date_year']=df['date'].str[6:10]
    df['date']=df['date'].str[6:10]+df['date'].str[3:5]+df['date'].str[0:2]
    df['date_month']=df['date_month'].astype(int)
    df['date_year']=df['date_year'].astype(int)

    return df
```

```
def annual_to_monthly_tax(df, annual_tax):
    df[annual_tax]=df[annual_tax].astype(float)
    df['monthly_tax']= ( (1 + df[annual_tax]/100)**(1/12) - 1)*100

    df_2=change_date_format(df)

    return df_2
```

```
meta_selic=annual_to_monthly_tax(meta_selic, 'meta_taxa_selic')
```

```
def apply_contribution_fixed(df):
    df=df.sort_values(by=['date'], ascending=False)
    df=create_contributions(df)
    df=df.drop_duplicates(subset=['date_month','date_year'], keep = 'last')
    df=df[df['contributions']!=0].reset_index(drop=True)
    return df
```

```
meta_selic_2=apply_contribution_fixed(meta_selic)
```

```
def calculate_compound_interest(df):
    investiment_month=10000
    investiment=0
    df_final=pd.DataFrame()

    for contributions in range(26):
        contributions=contributions+1
        investiment=investiment+investiment_month
        df_month=df[df['contributions']==contributions].reset_index(drop=True)
        df_month['pnl']=investiment*(1+df_month['monthly_tax']/100)**3
        investiment=df_month['pnl'].values
        df_final = pd.concat([df_final, df_month])
    return df_final
```

```
meta_selic_final=calculate_compound_interest(meta_selic_2, )
```

6.2) ipca

```
ipca= pd.read_csv("ipca_historic.csv", sep=";")
```

```
ipca_1=annual_to_monthly_tax(ipca.copy(), 'ipca_12m')
```

```
ipca_2=apply_contribution_fixed(ipca_1)
```

```
ipca_final=calculate_compound_interest(ipca_2)
```

7) Index

```
def calculate_stock_profit(df):
    investiment_month=10000
    investiment=0
    df_final=pd.DataFrame()

    for contributions in range(26):
        contributions=contributions+1
        investiment=investiment+investiment_month
        df_month=df[df['contributions']==contributions].reset_index(drop=True)
        df_month['pnl']=investiment*(1+df_month['monthly_tax']/100)
        investiment=df_month['pnl'].values
        df_final = pd.concat([df_final, df_month])
    return df_final
```

7.1) Bova11

```
index= pd.read_csv("index_bova.csv", sep=";")
```

```
index_2=change_date_format(index)
```

```
index_c=apply_contribution_fixed(index_2)
```

```
# Create a monthly tax
```

```
index_result=index_c[[ 'contributions', 'close']].reset_index(drop=True)
```

```
index_result['contributions']=index_result['contributions']-1
```

```
index_result=index_result.rename(columns={"close": "close_end"})
```

```
index_result_2=pd.merge(index_c,index_result,how='left',left_on=['contributions'],right_on=['contributions'])
```

```
index_result_2['monthly_tax']=((index_result_2['close_end']-index_result_2['close'] )/index_result_2['close']  )*100
```

```
index_result_bova=calculate_stock_profit(index_result_2)
```

7.2) IVVB11

```
index= pd.read_csv("index_ivvb.csv", sep=";")
```

```
index_2=change_date_format(index)
```

```
index_c=apply_contribution_fixed(index_2)
```

```
index_result=index_c[[ 'contributions', 'close']].reset_index(drop=True)
```

```
index_result['contributions']=index_result['contributions']-1
```

```
index_result=index_result.rename(columns={"close": "close_end"})
```

```
index_result_2=pd.merge(index_c,index_result,how='left',left_on=['contributions'],right_on=['contributions'])
```

```
index_result_2['monthly_tax']=((index_result_2['close_end']-index_result_2['close'] )/index_result_2['close']  )*100
```

```
index_result_ivvb=calculate_stock_profit(index_result_2)
```

9.6. TIMES SERIES: FACEBOOK PROFET

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import datetime
from statsmodels.tsa.seasonal import seasonal_decompose
```

```
!pip install fbprophet
```

```
from fbprophet import Prophet
```

```
date_quote=pd.read_csv('date_quote_complete.csv', sep=';')
```

```
date_quote['date_month']=date_quote['date'].str[3:5]
date_quote['date_year']=date_quote['date'].str[6:10]
date_quote['date_2']=date_quote['date'].str[6:10]+date_quote['date'].str[3:5]+date_quote['date'].str[0:2]
```

```
date_quote.head()
```

	date	open	high	low	close	adj close	volume	company	date_month	date_year	date_2
0	02/01/2015	11.613673	11.857728	11.285460	11.319123	8.387552	42539.0	CMIG3	01	2015	20150102
1	05/01/2015	11.319123	11.706246	11.117146	11.117146	8.237885	106705.0	CMIG3	01	2015	20150105
2	06/01/2015	11.150809	11.580010	11.108731	11.108731	8.231650	113359.0	CMIG3	01	2015	20150106
3	07/01/2015	11.243382	11.428527	11.201303	11.234966	8.325192	148294.0	CMIG3	01	2015	20150107
4	08/01/2015	11.361202	11.546347	11.058236	11.058236	8.194233	46579.0	CMIG3	01	2015	20150108

```
date_quote['date_month']=date_quote['date_month'].astype(int)
date_quote['date_year']=date_quote['date_year'].astype(int)
```

```
def create_contributions(df):
    df['contributions']=0
    contributions=0
    for year in range(2014,2022):
        for month in range(3, 13, 3):
            contributions=contributions+1
            # Condition
            conditions = [
                (df['date_year'] == year) & (df['date_month'] == month)]
            choices = [contributions]
            df['contributions'] = np.select(conditions, choices, default=df['contributions'])
    return df
```

```
date_quote_2=create_contributions(date_quote)
```

```
date_quote_2=date_quote_2.sort_values(by=['company', 'date_2'], ascending=True)
```

```
date_quote_contributions=date_quote_2.drop_duplicates(subset=['company', 'contributions'], keep='first')
date_quote_contributions=date_quote_contributions[date_quote_contributions['contributions']!=0]
```

```
# Add contributions
date_quote_contributions_clean=date_quote_contributions[['date_2', 'company', 'contributions']]
```

```
date_quote=date_quote.rename(columns={'contributions': 'contributions_old'})
```

```
date_quote_cont=date_quote.merge(date_quote_contributions_clean, left_on=['date_2', 'company'],
                                 right_on=['date_2', 'company'], how='left')
```

```
date_quote_cont=date_quote_cont.sort_values(by=['company', 'date_2'], ascending=True)
```

```
# Add final value and difference in days
date_quote_contributions_clean_2=date_quote_contributions[['date', 'company','close','contributions']].reset_index(drop=True)
date_quote_contributions_clean_2['contributions']=date_quote_contributions_clean_2['contributions']-1


date_quote_contributions_clean_2 = date_quote_contributions_clean_2.rename(columns={'date': 'date_fim', 'close': 'close_fim'})
```

```
date_quote_cont_2=date_quote_cont.merge(date_quote_contributions_clean_2, left_on=['contributions', 'company'],
                                        right_on=['contributions', 'company'], how='left')
```

```
date_quote_cont_2['date']=pd.to_datetime(date_quote_cont_2['date'], format='%d/%m/%Y')
date_quote_cont_2['date_fim']=pd.to_datetime(date_quote_cont_2['date_fim'], format='%d/%m/%Y')
date_quote_cont_2['days'] = (date_quote_cont_2['date_fim'] - date_quote_cont_2['date']).dt.days
```

```
companies_list=date_quote_contributions.groupby("company")["date"].count().reset_index()
companies_list=companies_list['company'].values.tolist()
```

Predictions

```
def applied_model(df):
  dataset = df[['date', 'close']].rename(columns = {'date': 'ds', 'close': 'y'})
  modelo = Prophet()
  modelo.fit(dataset)

  futuro = modelo.make_future_dataframe(periods=int(days_ref))
  prediction = modelo.predict(futuro)
  prediction = prediction [['ds', 'yhat']].rename(columns = {'ds': 'date', 'yhat': 'prediction'})

  return prediction
```

```
date_quote_prediction=pd.DataFrame()
for i in range(len(companies_list)):
  date_quote_cont_company = date_quote_cont_2[date_quote_cont_2['company']==companies_list[i]]
  for counter in range(1, len(all_contributions)):
    try:
      date_ref = date_quote_cont_company[date_quote_cont_company['contributions']==counter].iloc[0]['date_2']
      days_ref = date_quote_cont_company[date_quote_cont_company['contributions']==counter].iloc[0]['days']

      date_quote_cont_company_1 = date_quote_cont_company[date_quote_cont_company['date_2']<=date_ref]

      prediction=applied_model(date_quote_cont_company_1)
      prediction=prediction.rename(columns = {'date': 'date_fim'})

      date_quote_prediction_add=date_quote_cont_company_1.merge(prediction, left_on=['date_fim'], right_on=['date_fim'],
                                                how='left')
      date_quote_prediction_add = date_quote_prediction_add[~date_quote_prediction_add['prediction'].isna()]
      date_quote_prediction_add=date_quote_prediction_add.sort_values(by=['contributions'])
      date_quote_prediction_add=date_quote_prediction_add.drop_duplicates(subset=['company'], keep='last')

      date_quote_prediction = pd.concat([ date_quote_prediction, date_quote_prediction_add])
    except:
        pass
  print (i)
  date_quote_prediction.to_csv('prediction.csv')
```

```
date_quote_prediction=pd.DataFrame()
```

9.7. MACHINE LEARNING MODEL

Modeling

```
from sklearn.preprocessing import MinMaxScaler
import statsmodels.api as sm
```

```
X_train = prepare_dataset_clean_correlation.drop(columns=['real_profit_loss'])
```

```
y_train = prepare_dataset_clean_correlation['real_profit_loss']
```

```
features_x=list(X_train.columns)
const_name=['const']
features= const_name + features_x
```

```
scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
# train MLR model
X_train = sm.add_constant(X_train)
regressor = sm.OLS(y_train, X_train).fit()
regressor.summary(xname=features)
```

OLS Regression Results

Dep. Variable:	real_profit_loss	R-squared:	0.044
Model:	OLS	Adj. R-squared:	0.026
Method:	Least Squares	F-statistic:	2.445
Date:	Sun, 16 Jan 2022	Prob (F-statistic):	0.00126
Time:	19:38:31	Log-Likelihood:	3.7225
No. Observations:	871	AIC:	26.55
Df Residuals:	854	BIC:	107.6
Df Model:	16		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.1480	0.451	0.328	0.743	-0.737	1.033
total_assets	0.0310	0.134	0.232	0.817	-0.232	0.294
shares	0.0540	0.122	0.443	0.658	-0.185	0.294
net_profit	0.8468	0.432	1.959	0.050	-0.002	1.695
growth_profit	0.1273	0.243	0.524	0.601	-0.350	0.605
net_equity	-0.1457	0.296	-0.492	0.623	-0.727	0.435
net_revenue_12	0.2826	0.256	1.102	0.271	-0.221	0.786
net_profit_12m	0.0540	0.315	0.171	0.864	-0.564	0.672
cl	-0.0137	0.106	-0.129	0.897	-0.221	0.194
roe	0.1174	0.299	0.392	0.695	-0.470	0.705
net_margin	-1.0904	0.244	-4.470	0.000	-1.569	-0.612
eps	0.0664	0.170	0.390	0.697	-0.268	0.401
bvps	0.0506	0.132	0.383	0.702	-0.209	0.310
d_e	-0.1135	0.256	-0.443	0.658	-0.616	0.389
close	0.3292	0.356	0.926	0.355	-0.369	1.027
volume	0.0515	0.149	0.346	0.730	-0.241	0.344
prediction	-0.5689	0.452	-1.260	0.208	-1.455	0.318

Omnibus:	632.248	Durbin-Watson:	2.137
Prob(Omnibus):	0.000	Jarque-Bera (JB):	22363.712
Skew:	2.855	Prob(JB):	0.00
Kurtosis:	27.158	Cond. No.	151.

Apply only prediction and close variables

```
X_2 = prepare_dataset_clean_correlation.drop(columns=['real_profit_loss'])
```

```
# Keep the signficant variables
X_2=X_2[[ 'net_margin', 'net_profit']]
```

```
features_x=list(X_2.columns)
const_name=['const']
features_2= const_name + features_x
```

```
scaler = MinMaxScaler()
X_2 = scaler.fit_transform(X_2)
# train MLR model
X_2 = sm.add_constant(X_2)
regressor = sm.OLS(y_train, X_2).fit()
regressor.summary(xname=features_2)
```

OLS Regression Results

Dep. Variable:	real_profit_loss	R-squared:	0.036
Model:	OLS	Adj. R-squared:	0.034
Method:	Least Squares	F-statistic:	16.13
Date:	Sun, 16 Jan 2022	Prob (F-statistic):	1.32e-07
Time:	20:57:22	Log-Likelihood:	0.11118
No. Observations:	871	AIC:	5.778
Df Residuals:	868	BIC:	20.09
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.4022	0.309	1.304	0.193	-0.203	1.008
net_margin	-1.0924	0.243	-4.500	0.000	-1.569	-0.616
net_profit	0.7951	0.231	3.442	0.001	0.342	1.249

Omnibus:	621.263	Durbin-Watson:	2.132
Prob(Omnibus):	0.000	Jarque-Bera (JB):	21264.228
Skew:	2.790	Prob(JB):	0.00
Kurtosis:	26.554	Cond. No.	77.7

Apply the model

```
# Modeling
```

```
from sklearn.linear_model import LinearRegression
```

```
multiple_price_target = multiple_price_target[~multiple_price_target['net_profit'].isna()]
```

```
date_detail_new_model= pd.DataFrame()
investiment_month=10000
investiment=0
# 4 contribution to train the model. After, the model will learn with new information
contribution_filter=[1,2,3,4]
```

```
for contributions in range(5,31):

    investiment=investiment+investiment_month
    investiment_per_share=investiment/10

    # train the model
    df_model=multiple_price_target[multiple_price_target['contributions'].isin(contribution_filter)].reset_index(drop=True)
    df_model_x=df_model[[ 'net_margin', 'net_profit']]
    df_model_y=df_model[[ 'real_profit_loss']]


    regressor = LinearRegression()
    regressor.fit(df_model_x, df_model_y)
    score = regressor.score(df_model_x, df_model_y)

    # apply the model
    df_next=multiple_price_target[multiple_price_target['contributions'].isin([contributions])].reset_index(drop=True)
    df_next_x=df_next[[ 'net_margin', 'net_profit']]
    prediction = regressor.predict(df_next_x)

    df_next['result_prediction']=df_next['net_margin']*regressor.coef_[0,0] + df_next['net_profit']*regressor.coef_[0, 1]
    + regressor.intercept_[0]

    # select the stocks, olny profitable
    df_next_clean= df_next[df_next['net_profit']>0]
    selected_next=df_next_clean.sort_values(by=['result_prediction'], ascending=False).head(10)
    selected_next['final_profit_loss']=(1+selected_next['real_profit_loss'])*investiment_per_share

    date_detail_new_model= pd.concat([date_detail_new_model, selected_next])

    investiment=selected_next['final_profit_loss'].sum()
    contributions_add=[contributions]
    contribution_filter=contribution_filter + contributions_add
```

```
date_detail_new_model.to_csv('date_detail_new_model.csv')
```