



**Kaunas University of Technology**  
Faculty of Electrical and Electronics Engineering

# **Application of Machine Learning to Stock Market Index Prediction**

Bachelor's Final Degree Project

---

**Martin Xavier Gomez Salazar**

Project author

**Prof. Dr. Renaldas Urniežius**

Supervisor

---

**Kaunas, 2024**



**Kaunas University of Technology**  
Faculty of Electrical and Electronics Engineering

# **Application of Machine Learning to Stock Market Index Prediction**

Bachelor's Final Degree Project  
Intelligent Robotic Systems (6121EX013)

---

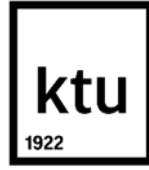
**Martin Xavier Gomez Salazar**  
Project author

**Prof. Dr. Renaldas Urniežius**  
Supervisor

**Assoc. Prof. Dr. Leonas Balaševičius**  
Reviewer

---

**Kaunas, 2024**



**Kaunas University of Technology**

Faculty of Electrical and Electronics Engineering

Martin Xavier Gomez Salazar

## **Application of Machine Learning to Stock Market Index Prediction**

### **Declaration of Academic Integrity**

I confirm that the final project of mine, Martin Xavier Gomez Salazar, on the topic „Application of Machine Learning to Stock Market Index Prediction“ is written completely by myself; all the provided data and research results are correct and have been obtained honestly. None of the parts of this thesis have been plagiarised from any printed, Internet-based, or otherwise recorded sources. All direct and indirect quotations from external resources are indicated in the list of references. No monetary funds (unless required by Law) have been paid to anyone for any contribution to this project.

I fully and completely understand that any discovery of any manifestations/case/facts of dishonesty inevitably results in me incurring a penalty according to the procedure(s) effective at Kaunas University of Technology.

---

(name and surname filled in by hand)

---

(signature)

2024 ..... ..

### TASK OF FINAL PROJECT OF UNDERGRADUATE (BACHELOR) STUDIES

**Issued to the Student:** \_\_\_\_\_ **Martin Xavier Gomez Salazar** \_\_\_\_\_ **Group** E RBU – 0

**1. Project Subject:**

Lithuanian Language: \_\_\_\_\_ *Mašininio mokymo taikymas akcijų rinkos indekso prognozavimui* \_\_\_\_\_

English Language: \_\_\_\_\_ *Application of Machine Learning to Stock Market Index Prediction* \_\_\_\_\_

Approved 2024 yy. \_May\_ mm. \_6\_ dd. Decree of Dean Nr. V25-03-9

**2. Goal of the Project:** \_\_\_\_\_ *To develop a stock market forecasting algorithm based on machine learning* \_\_\_\_\_

**3. Specification of Final Project:** \_\_\_\_\_ *The research must be done with real market data* \_\_\_\_\_

**4. Project's Structure.** *The content is concretized together with supervisor, considering the format of the final project, which is listed in 14 and 15 points of Combined Description of Preparation, Defence and Keeping of Final Projects Methodical Requirements*

*1. Perform literature analysis, analyze what the stock market and its index are, what ETFs replicate market index and can be traded. 2. what machine learning methods are used for index price prediction. 3. Download real market index or ETF trade data, preprocess them, prepare a selected type of machine learning model, train it with training data and optimize model parameters. 4. Perform a research on the effectiveness of the developed algorithm and make final conclusions*

**5. Economical Part.** *If economical substantiation is needed; content and scope is concretized together with supervisor during preparation of final projects*

**6. Graphic Part.** *If necessary, the following schemes, algorithms, and assembly drawings; content and scope is concretized together with supervisor during preparation of final projects*

**5. This Task is Integral Part of Final Project of Undergraduate (Bachelor) Studies**

**6. The Term of Final Project Submission to Defense Work at a Public Session of Qualification Commission.** \_\_\_\_\_ *until 2024-05-12* \_\_\_\_\_  
(date)

I received this task: \_\_\_\_\_ *Martin Xavier Gomez Salazar* \_\_\_\_\_ *2024-02-07* \_\_\_\_\_

(student's name, surname, signature)

(date)

Supervisor: \_\_\_\_\_ *Prof. Dr. Renaldas Urniežius* \_\_\_\_\_ *2024-02-07* \_\_\_\_\_

(position, name, surname, signature)

(date)

Martin Xavier Gomez Salazar. Application of machine learning to stock market index prediction. Bachelor's Final Degree Project. supervisor Prof. Renaldas Urniežius; Faculty of Electrical and Electronics Engineering, Kaunas University of Technology.

Study field and area (study field group): 6121EX013 Intelligent Robotics Systems / E RBU – 0.

Keywords: Stock Market, Artificial Intelligence, Machine learning, SPY, S&P500, Decision tree, Random Forest.

Kaunas, 2024. Number of pages 41.

### **Summary**

This project is an investigation on how Machine learning can be applied to predict the upward trends in the stock market to make investment decisions and gain profits. This investigation examines what is the stock market, factors that influence it, indicators and trading tools used for investment. It also provides an overview about Artificial Intelligence and its subfield Machine learning, and how these two technologies have been applied to the market for the same purpose of finding trends and predict when a is a good time to buy or sell stock market assets. For the practical part of this project Random Forest ML model was implemented using python programming language and all the required libraries for Machine learning modeling. Within the Random Forest model two trading strategies were compared; the “Buy-and-hold” strategy and “Daily-trading” strategy by giving the model an initial investment budget of one thousand euros. The results showed that the daily trading strategy is slightly more profitable. The Exchange-Trust-Fund (ETF) in question was the SPY and historical data was extracted from the last 30 years through the online platform “Yahoo Finance”. This work presents the implementation of the Random Forest model, the extraction and preprocessing of the data, the creation of the targets to predict, and overall analysis of the benefits of using machine learning for profit gain.

Martin Xavier Gomez Salazar. Mašininio mokymo taikymas akcijų rinkos indekso prognozavimui. Bakalauro baigiamasis projektas / vadovas Prof. Dr. Renaldas Urniežius; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): 6121EX013 Intelligent Robotics Systems / E RBU - 0

Reikšminiai žodžiai: Akcijų rinka, dirbtinis intelektas, mašininis mokymasis, SPY, atsitiktinis miškas.

Kaunas, 2024. Puslapių sk. p. 41

## **Santrauka**

Šis projektas yra tyrimas, kaip mašininį mokymąsi galima pritaikyti prognozuojant akcijų rinkos augimo tendencijas, siekiant priimti investicinius sprendimus ir gauti pelną. Šiame tyrime nagrinėjama, kas yra akcijų rinka, ją įtakojantys veiksniai, investavimui naudojami rodikliai ir prekybos įrankiai. Jame taip pat pateikiama dirbtinio intelekto ir jo polaukio Mašininio mokymosi apžvalga ir tai, kaip šios dvi technologijos buvo pritaikytos rinkoje, siekiant rasti tendencijas ir numatyti, kada tinkamas laikas pirkti ar parduoti akcijų rinkos turtą. Praktinėje šio projekto dalyje Random Forest ML modelis buvo įdiegtas naudojant python programavimo kalbą ir visas mašininio mokymosi modeliavimui reikalingas bibliotekas. Random Forest modelyje buvo palygintos dvi prekybos strategijos; „Buy-and-hold“ strategija „Daily-trading“ strategija, modeliui suteikiant pradinį tūkstančio eurų investicijų biudžetą. Rezultatai parodė, kad kasdienė prekybos strategija yra šiek tiek pelningesnė. Minėtas „Exchange-Trust-Fund“ (ETF) buvo SPY, o pastarųjų 30 metų istoriniai duomenys buvo gauti per internetinę platformą „Yahoo Finance“. Šiame darbe pristatomas atsitiktinio miško modelio įgyvendinimas, duomenų išgavimas ir išankstinis apdorojimas, numatomų tikslų kūrimas ir bendra mašininio mokymosi naudos siekiant pelno analizė.

## Table of contents

<b>List of figures .....</b>	<b>9</b>
<b>List of tables .....</b>	<b>11</b>
<b>List of abbreviations and terms.....</b>	<b>12</b>
<b>Introduction .....</b>	<b>13</b>
<b>1. Analytical Investigation .....</b>	<b>14</b>
1.1. About the Stock Market and Trading .....	15
1.1.1. What is the Stock Market .....	15
1.1.2. Stock Market Tools and Data .....	17
1.1.3. Trading and Its Strategies .....	21
1.2. Machine Learning and Artificial Intelligence .....	21
1.3. ML and AI applied to trading .....	24
1.3.1. Current AI-based trading tools: .....	24
<b>2. Methodology Selection .....</b>	<b>25</b>
2.1. Libraries included. ....	27
2.2. Model Parameters .....	28
<b>3. Experimental Investigation.....</b>	<b>31</b>
3.1. Research Roadmap .....	31
3.2. Model Algorithm to Code .....	34
3.2.1. Data Acquisition & Cleansing.....	34
3.2.2. Visualization and Analysis .....	34
3.2.3. Data Splitting for model training.....	37
3.2.4. Optimal Parameters selection for ML model. ....	38
3.2.5. Training the model .....	40
3.2.6. Compare Target with Predictions .....	41
3.2.7. Model Backtesting.....	42

3.2.8. Adding more data to improve accuracy.....	44
3.2.9. Predictions Result Comparison with “Buy and Hold” Strategy.....	46
<b>Conclusions .....</b>	<b>51</b>
<b>List of references.....</b>	<b>52</b>
<b>Appendices .....</b>	<b>55</b>



## List of figures

<b>Fig 1:</b> Economic Cycle <sup>[5]</sup> .....	15
<b>Fig 2:</b> Screenshot Extracted from Personal Broker Account (ETORO) .....	17
<b>Fig 3:</b> Mountain Graph <sup>[24]</sup> . ....	19
<b>Fig 4:</b> Candlestick Graph <sup>[24]</sup> .....	19
<b>Fig 5:</b> Candlestick Anatomy <sup>[32]</sup> .....	20
<b>Fig 6:</b> Linear regression example <sup>[30]</sup> . ....	23
<b>Fig 7:</b> Decision tree example <sup>[14]</sup> . ....	25
<b>Fig 8:</b> Random Forest Flowchart <sup>[20]</sup> .....	26
<b>Fig 9:</b> Python libraries Imported .....	28
<b>Fig 10:</b> Sci-kit learn library integration.....	28
<b>Fig 11:</b> Research Roadmap Flowchart .....	32
<b>Fig 12:</b> Code to request data from Yfinance API.....	34
<b>Fig 13:</b> Code written to obtain a visualization graph .....	35
<b>Fig 14:</b> Graph plotted on Jnotebook to compare SPY with S&P500.....	36
<b>Fig 15:</b> Code to set the target of the investigation. ....	36
<b>Fig 16:</b> Code to split the dataset into train & test, predictors also shown.....	38
<b>Fig 17:</b> Loops determine optimal parameters, n_estimators and min_sample_split iterations. ....	38
<b>Fig 18:</b> Precision Scores results table.....	39
<b>Fig 19:</b> Precision scores results graph. ....	40
<b>Fig 20:</b> Optimized model Training.....	41
<b>Fig 21:</b> Target vs Predictions concatenation. ....	41
<b>Fig 22:</b> Target vs Predictions concatenation graph. ....	42
<b>Fig 23:</b> Prediction and backtesting Functions. ....	43
<b>Fig 24:</b> Back Testing Results .....	44
<b>Fig 25:</b> Addition of new features to dataset .....	45

<b>Fig 26: Backtesting after features addition .....</b>	<b>46</b>
--	-----------

## List of tables

<b>Table 1:</b> Online Brokers vs Bank investing fees comparison <sup>[11,13,15]</sup> .....	16
<b>Table 2:</b> Comparison of different ML Models <sup>[3]</sup> .....	23
<b>Table 3:</b> Model Data Input Example .....	29
<b>Table 4:</b> Random Forest model parameter selection .....	30
<b>Table 5</b> Addition of the target column to the dataset.....	37
<b>Table 6:</b> Confidence scores table for the different models trained. ....	48
<b>Table 7:</b> Daily Traiding with ML profitability .....	48
<b>Table 8:</b> Strategies Comparison.....	49

## **List of abbreviations and terms**

### **Abbreviations:**

AI – Artificial Intelligence

API – Application Programming Interface

CNN – Convolutional Neural Networks

EPS – Earnings per Share

ETF's – Exchange Trust Funds

GDP – Gross Domestic Product

ML – Machine Learning

P/E – Price to Earnings

P/L – Profit & Loss

REIT's – Real Estate Investment Trusts

TNN – Traditional Neural Networks

NLP – Natural Language Processing

### **Terms:**

**Trades** – The activity of buying and selling, or exchanging, goods and/or services between people or countries<sup>[27]</sup>. It is also applied in the Stock Market.

**Assets** – something valuable belonging to a person or organization that can be used for the payment of debts<sup>[9]</sup>.

**Shares** – A part of something that has been divided between several people<sup>[22]</sup>.

**Tickers** – The letters that are used to refer to a company's shares on a stock market, this name originates from the electronic screens that show share prices on a particular stock market<sup>[26]</sup>.

## Introduction

With the innovative discoveries and evolution of artificial intelligence the world is evolving and changing at a rate that we haven't seen before. This new technology accompanied by machine learning and deep learning has offered us new tools to approach problems and challenges in a different way, achieving surprising results. It is true that these instruments are reshaping the way we predict and forecast events that are about to happen, letting us act way ahead of the occurrence of the event.

Money plays an essential role on everybody's life, and it is important to always have backup plans in case of financial emergencies. For most people this backup plan could be a financial safety pillow or having some savings. No matter in what way we are saving money, there is a big flaw on regular saving strategies; it is what differentiates people that actively work to get money, with people that make their money work for them to get more money in a passively way. This flaw relies on a phenomenon known as inflation, that is one of the economic phenomena that are caused by offer and demand in the market. This phenomenon, makes our money lose value over time, meaning that the more we keep our money static on our bank accounts, the power that it has to buy us food, products, services and other necessities, decreases; meaning that what we can buy now with the money that we have in the bank, will only buy us a fraction of the same in the future.

To counteract this effect that is why that money should be invested, to keep it growing over time and counteract the inflation rate at which it devalues, this work delves into ways of investing funds and how machine learning, a subset of artificial intelligence, can help those investments be profitable.

The main scope of this investigation is to determine how far prediction algorithms have advanced and if they can predict the market with accurate results. As a big risk in investments is the volatility of the market and world events which influence it, machine learning can help mitigate those risks by analyzing historical data and its trends. The research question lies on how machine learning methods can predict the market to aid in profitable investments. To answer the research question, Random Forest ML model was utilized to predict SPY ETF future share prices, and different trading strategies were tested in the model using python programming language and 30-year-old stock market data.

The aim of the project is to create a machine learning tool to predict the SPY ETF behavior in the market and aid investing decisions.

### Project tasks:

- 1. Perform literature analysis, analyze what the stock market and its index are, what ETFs replicate market index and can be traded.*
- 2. what machine learning methods are used for index price prediction.*
- 3. Download real market index or ETF trade data, preprocess them, prepare a selected type of machine learning model, train it with training data and optimize model parameters.*
- 4. Perform a research on the effectiveness of the developed algorithm and make final conclusions*

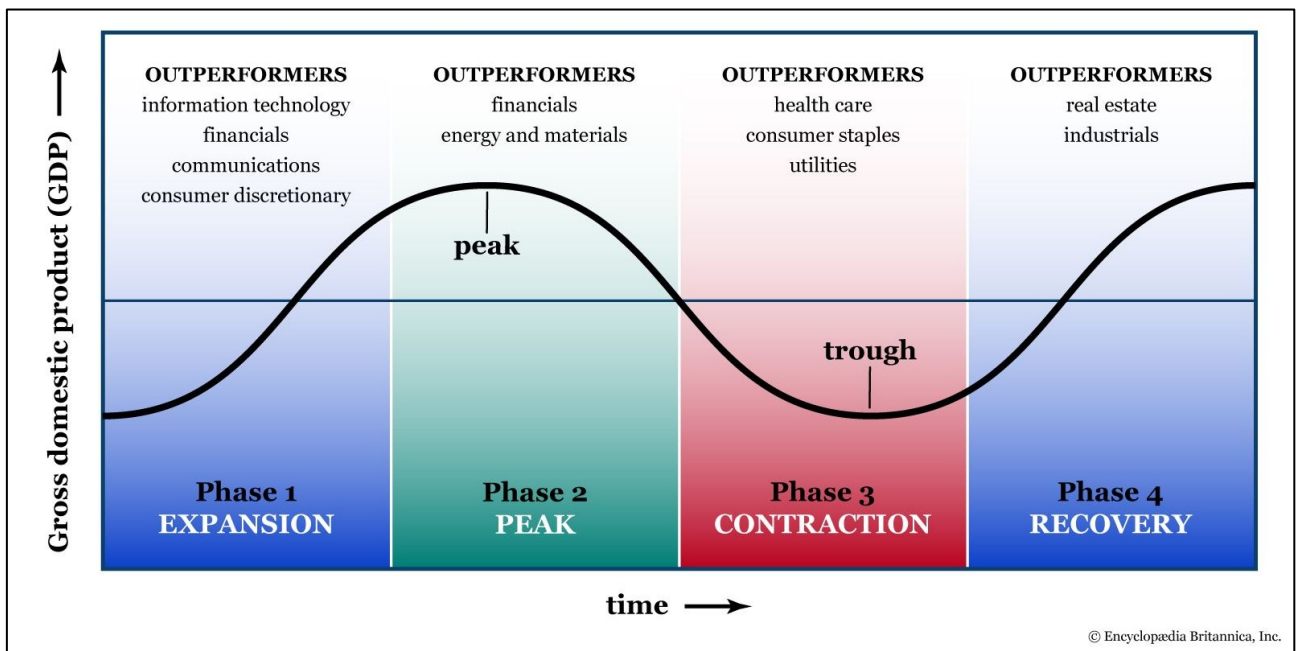
## 1. Analytical Investigation

The scope of this project is built on the understanding of basic economic principles and how money circulates throughout markets and global economies. To begin understanding these economic phenomena, a good starting point is with supply and demand. Demand refers to the consumers inclination to purchase a good or service. Factors like product quality, popularity, trends, and current income influence the consumers judgment when deciding where to spend their money. Supply on the other hand, refers to the ability of producers to provide the goods for a certain price throughout time. These economic phenomena are what determine the price of products and explain why there is not a fixed price to every good all the time. When there is a scarce amount of a product or service, it means that there exists a limited supply; if many people want to have said product or service, there is a high demand. Prices are thus affected; as an owner of a limited supply product, one can substantially increase the price because of high demand <sup>[19]</sup>.

These dynamics greatly influence inflation and deflation. Inflation happens when there is a sudden increase in prices, and it is because there is a higher demand than the actual supply of the service or product. Money loses its value, and suddenly one would need more amounts of money so that it has the same value as before. If there is more money circulating, it devalues. Deflation, on the contrary, happens when there is less demand for a highly supplied product or service. There are more goods available than means to buy it, therefore, to keep the economy flowing, vendors must reduce prices to make business. <sup>[19]</sup>.

Understanding these terms makes it easier to understand the Economic Cycle which is recurring in economies and important to consider when investing in stocks. Encyclopedia Britannica defines an economic cycle as “a recurrent boom-and-bust phases” which come around many times in markets and they typically have four defined stages<sup>[5]</sup>. A visual depiction of said cycle is shown in figure 1. First it starts with the expansion phase, economy starts to grow, people are spending money, its easier to invest as interest rates are low, and gross domestic product (GDP) is high. Then the peak phase, the rising of the previous cycle hits its climax. After the contraction phase, unemployment rate increases, investments are not as common and falling GDP are present. The economy is the lowest it has been and will be at this cycle, another word used for this part is recession. Finally, the recovery phase is when businesses start to thrive again. It is the calm after the storm, consumers’ demand rises, higher income, and employment rates are also present <sup>[5]</sup>.

As figure 1 shows, this cycle acts as a wave, changing through time. Each cycle can vary in duration, from as little as two months up to decades. Therefore, the knowledge of the economic cycle alone is not enough to know when and where to invest, although it may seem intuitive. Now that the basics of economics have been covered, the stock market can be defined, and strategies used in trading to make educated decisions can be presented.



**Fig 1: Economic Cycle** <sup>[5]</sup>

## 1.1. About the Stock Market and Trading

The stock market has been around for centuries, and it has evolved accordingly. It is a complex market with many terms and tools to understand. It is based on core economic phenomena like supply and demand, inflation & deflation, and it is influenced by historical, current world events, and several other derived factors. To provide a good overview and understanding of it, one must learn about what the market is, what is being commercialized in the market, the different areas, and the data that is being handled.

### 1.1.1. What is the Stock Market

First, the stock market is defined as "... a regulated environment where sellers and buyers commercialize shares of a company. It is not a secret that companies divide themselves into multiple shares that could be sold to the public at a price per share, making buyers of these shares part 'owners' of the company." <sup>[36]</sup>. In simpler terms, it is like a big playground where people play different kinds of games, the playground is the stock market. Following this analogy, as in any playground, there are different game stations, in the same way there are different markets inside the Stock Market, these are referred to as the 'stock exchanges'. Geography is one differentiation for different stock

exchanges, companies in specific countries trade inside specific ones. Inside stock exchanges people can trade shares of companies, commodities (gold, silver, or oil), currencies and cryptocurrencies. Other ones like Real-estate Investment trusts (REIT's) or bonds can also be acquired. There also are Exchange Traded Fund (ETF's) which are a share of many companies together, so when one buys a piece of an ETF, a small percentage of many different companies is acquired simultaneously <sup>[36]</sup>.

Individuals cannot access the stock market directly because of stakeholder's regulations. In this scenario the stakeholders are the companies, because their assets are on the line, government entities because they regulate the companies, and investors themselves. These regulations ensure traceability of the shares for safety reasons. In consequence, as intermediaries between a physical person and the stock market, brokers exist; they are third parties who are authorized to buy and sell stocks on behalf of a natural person. Back in the day, brokers were people sitting behind desks who had access to market and stocks information, and one had to call them to place an order. This was a lengthy process and gave way to human error and scams because the buying person did not always have the expertise on the topic. Today, the process of trading stocks with a broker is easier because one can sign up with an online broker and start trading, all from a personal smartphone or computer. There is also the possibility to access real-time data and learning material on investing, to allow more individuals to make educated decisions on trades. Investing with an online broker is safer nowadays because this brokers are regulated by companies and local governments <sup>[31]</sup>.

In the past a person would have to pay a broker fee before an order was placed in the stock market. Today, one can easily start trading without any initial fee, and brokers only take a small commission of the investment once they are made <sup>[31]</sup>. Brokers who do not gain money from investor's investments, gain money through interests like regular banks do by investing the static funds in the person's account. Popular brokers that operate in the European union are Interactive Brokers, eToro, XTB, and Trading 212<sup>[11]</sup>. These brokers have online platforms, and a person can open an account in a day or less. Local banks can also act as a broker between the investors and the stock market with a higher incurred fee, table 1 shows the comparison between online brokers and Lithuanian-based banks.






**Table 1:** Online Brokers vs Bank investing fees comparison<sup>[11,13,15]</sup>

Broker	Comission per trade	Min. Deposit	Comment/Extra fees
Interactive Brokers	\$3.2 if trade < \$6200/ 0.05% of trade value if trade > \$ 6200	-	-
eToro	\$0	\$50	Withdrawal fee \$5 + Inactivity fee after a year (\$10/month)
XTB	0.2% of trade	-	Inactivity fee after a year (\$10/month)
Trading 212	\$0	\$1	-



Swedbank (LT)	0.25% of trade	€8	-
Luminor Bank (LT)	0.1% of trade	-	3 investment risk levels

In figure 2, an example of an online trading platform from eToro broker can be observed. The figure shows the assets in which orders were placed, they the five icons representing one ETF (SPY) and four companies' stocks (Google, Apple, Continental, and Nvidia), under the price column, the price for one individual share is presented, followed by the Units that represent the amount of shares bought, the Avg. Open column is the average price at which the value of the asset opens each day and the last column P/L represent the profit or loss on the investment. At the bottom we can see the cash available for investment, total amount invested, the total for P/L and the total value of the portfolio, that represents the final value of the money invested.

My Portfolio ▾					⌵	
<div> Orders Manual Trades Market Open Stocks ETFs </div>						
Asset (5)	Price	Units	Avg. Open	P/L		
 <b>SPY</b> SPDR S&P 500 ETF	509.04 0.15% (0.79)	3.32971 Long	497.8708	\$37.19	⋮	
 <b>GOOG</b> Alphabet	168.25 -3.13% (-5.43)	1.70884 Long	117.038	\$87.96	⋮	
 <b>AAPL</b> Apple	174.32 2.97% (5.03)	1.20753 Long	163.93	\$12.87	⋮	
 <b>CON.DE</b> Continental AG	61.08 -2.57% (-1.61)	0.41 Long	67.64	-\$3.23	⋮	
 <b>NVDA</b> NVIDIA Corporation	868.19 -4.22% (-3.65)	0.02191 Long	912.8673	-\$0.98	⋮	
<div> <div>\$0.04 Cash Available</div> <div>+</div> <div>\$2,095.72 Total Invested</div> <div>+</div> <div>\$133.81 Profit/Loss</div> <div>=</div> <div>\$2,229.57 Portfolio Value</div> </div>						

**Fig 2:** Screenshot Extracted from Personal Broker Account (ETORO)

### 1.1.2. Stock Market Tools and Data

Now that it has been established what the Stock Market and its derivatives are, the data handled in the market and tools used to analyze and use that data can be discussed. The main data analyzed in the stock exchanges are company's stock price over time. Apart from basic information like the company name, the stock exchange provides information like the company's statistics, previous closing price, daily range, a 52-week range, average volume, one-year return, market capitalization, P/E ratio, revenue, EPS, and dividends if offered by company. All of them offer the investor key

information about the risks of investment in specific assets, and they give the following insights about the stocks:

- The previous closing price refers to the price at which the market closed on the previous day.
- Daily range is the variation of the price throughout the day.
- 52-week range is the variation of the price every 52 weeks.
- Average volume is the average number of shares of a stock traded within that day.
- One-year return is the value increment percentage expected over each year.
- Market capitalization represents the total value of the company and determines its size in the market.
- The P/E ratio is the Price to Earnings ratio in other words the value of the stock to the earnings of the company, this reflects the interest of investors in a company.
- Revenue is the money generated by a company or business on a regular basis.
- EPS is the Earnings Per Share is the amount of money that corresponds to each shareholder after subtracting the operational costs from the revenue of the company.
- Dividends is a repartition of earnings that is given by companies to their shareholders regularly four times per year, however it varies depending on the company.<sup>[35]</sup>

Data visualization is an essential feature of the stock market and trading, it helps traders make educated decisions based on the stock's prices over time. There are many types of graphs to visualize

data, however they all show more-or-less the same information. For this specific work, mainly two graphs will be used to visualize data, they are mountain graphs and candlestick graphs seen in figure 3 and 4 respectively [32].



Fig 3: Mountain Graph<sup>[24]</sup>.

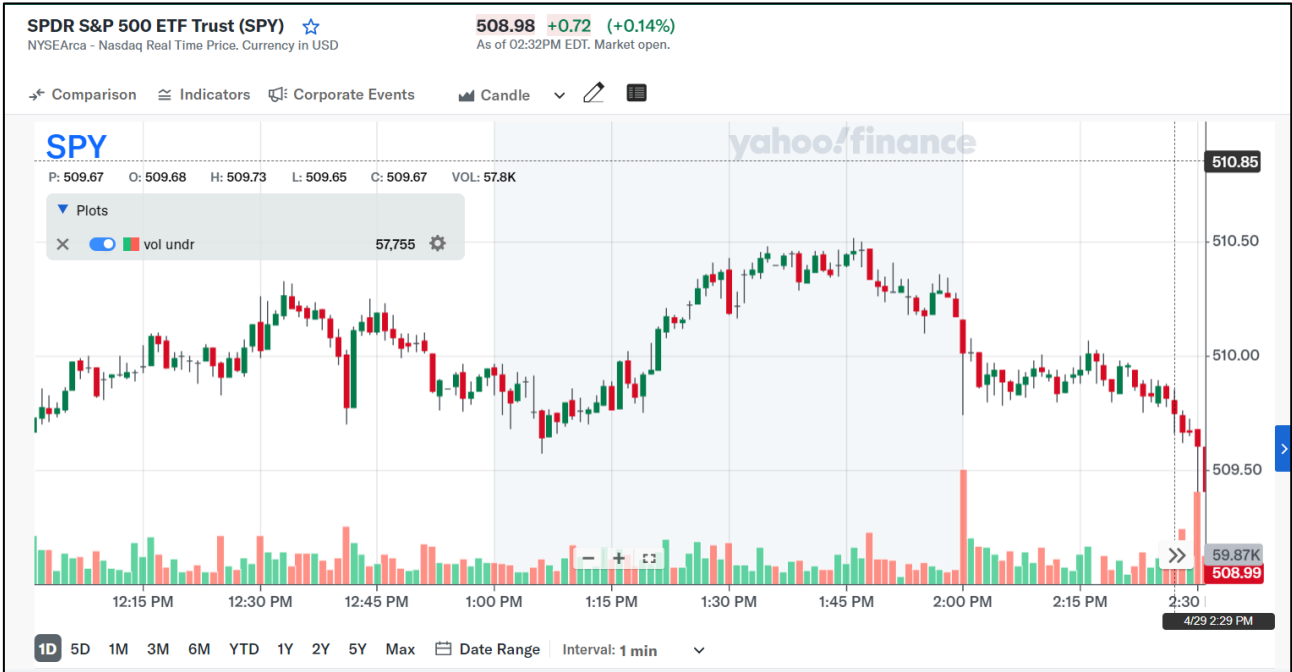
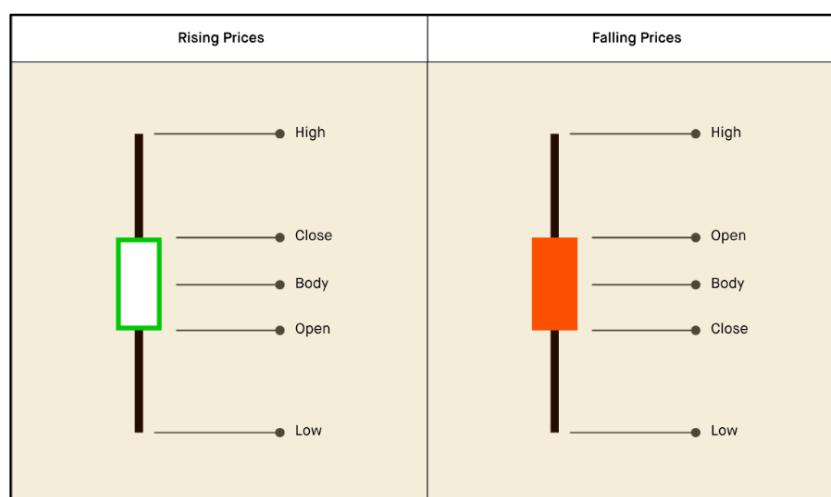


Fig 4: Candlestick Graph<sup>[24]</sup>.

As seen in both figures, the data plotted is price of the ETF’s share plotted over time, presented on the y axis and time on the x axis. This data is shown for the SPY ETF. One can choose to visualize on

which time frame to see the data; from as soon as one day or as broad as the ETF stock's behavior over the last 5 years.

The most common graph used worldwide is the Candlestick as it gives the most information visually easy to digest. According to a popular broker for cryptocurrencies' learning platform, Robbinhood Learn, a candlestick is defined as "In stock trading, a candlestick is a charting tool that quickly conveys a stock's opening, closing, high, and low price for the day." [32]. Figure 5 shows the anatomy of a single candlestick where one can see the different parts of the graph. The market has regular working hours as any job, once the market opens, that stock has a starting price, and the same when it closes. The 'Close' and 'Open' sections in the candlestick define these prices. Since the price of the stock fluctuates throughout the day, it reaches a maximum and minimum price, this is shown by the 'high' and 'low' parts of the candlestick on the specific date. Finally, the body of the candlestick based on its color shows whether the stock price shifts upward (green) or downward (red) in comparison to the previous day. This helps understand the trend the stock has taken better, rather than just seeing numbers on the screen. [32]



**Fig 5:** Candlestick Anatomy<sup>[32]</sup>.

Along with data visualization tools, forecasting methods are equally as important to ensure successful trading. Forecasting in simple terms, is a technique that predicts the output of a system based on certain inputs like previous historical data, it estimates the trend the data will follow; nonetheless, the results are not always accurate as sudden events can influence the results. For businesses, they use forecasting to decide where to spend their future funds based on supply and demand, as initially discussed in the analytical investigation [16]. In a trading environment, the method is used for the same purpose. As a trader, forecasting helps in predicting future trends of the stock. Its also useful to estimate when the price will fall or rise depending on previous data. With this new information one can choose what the best time to sell or buy a stock is.

As in any discipline, there are different types of forecasting methods used for different purposes, these can be classified into two main categories: quantitative and qualitative. Quantitative methods are solely based on quantifiable, tangible data such as statistics. It solely focuses on the numbers, the stocks' prices over time. Examples of quantitative forecasting methods are the Time Series Method, Discounting, and Analysis of Leading/Lagging indicators. Qualitative methods rely on expert financial analysts' opinions, and on market research. It is a more abstract approach useful for a short-term investment.

Time Series Analysis extrapolates the relationships of different variables in the past. Its outcome has some degree of confidence level in which the scope falls, therefore this is a quantitative method.

### **1.1.3. Trading and Its Strategies**

Different trading strategies output different results, however which strategy to use is mostly selected by the purpose behind trading itself. First of all, a 'trading strategy' can be defined as a structured plan for purchasing or selling stocks/other stock market assets that is proved to generate profit in relation to the investment. This strategy must be proved and consistent in order to be considered one; it's built up according to careful market research and technical and fundamental analysis <sup>[29]</sup>.

Factors which influence different trading strategies include technical indicators, industry, portfolio diversification, timeline, risk tolerance, tax considered, and leverage. If it is objective data, it can be used to formulate a strategy and constantly updated with feedback data. The main two trading strategies used in this work are; *'Buy-and-Hold'* and *'Daily Trading'*. The first one is very self explanatory, the trader first analyzes the stocks condition, past trends, current world events that influences the stock, market open price and decides to place the order on the trade. Once the stock is confirmed, the investor keeps the stock there until it grows to a desired amount. The latter has also an intuitive name <sup>[29]</sup>. Advantages of buy and hold: its a low-risk investment because its usually made on a stock exchange or ETF one knows it will always increase. Its a long-term strategy. Disadvantage, it takes a lot of time to see the returns on the investment. Advantages of daily trading is seeing rapid gains and disadvantage is its a higher-risk strategy. One is subject to volatility of the stock and market.

A trading discipline which aims to study a stock 's price over time using different statistical and geometric tools is referred to as a technical analysis <sup>[25]</sup> It helps traders identify patterns in the stocks's price charts and trends. Some indicators tell the trader when a stock price will raise, therefore it is a good moment to buy. Some indicators, like a peak in the price, a valley, or stock price oscillations are informative and aid in technical analysis.

## **1.2. Machine Learning and Artificial Intelligence**

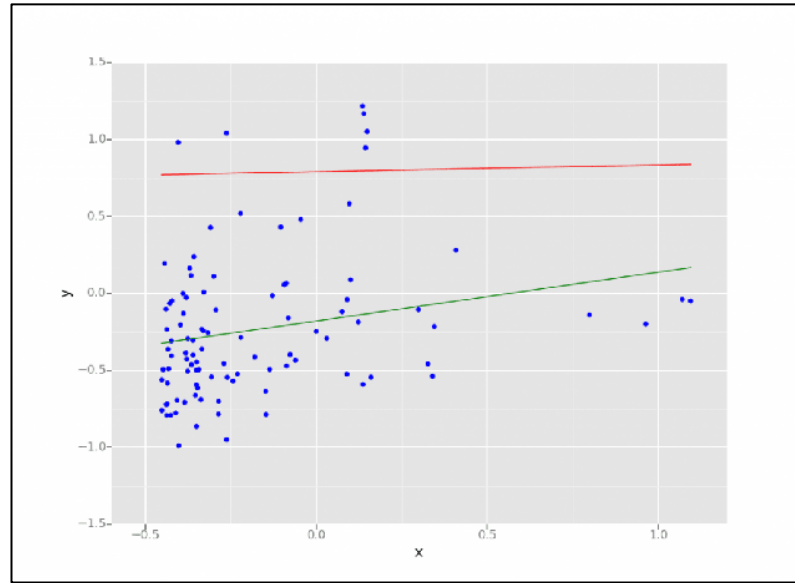
Artificial intelligence (AI) is the field of engineering that research ways to develop human-like solutions. The objective of AI is to develop computer systems that can learn, solve problems, and sustain themselves without human interaction. AI is a versatile discipline and can be applied to any industry. Today, humans interact with it basically on the daily. Speech recognition is an example of applied AI, any human with a smartphone experiences it with products such as Apple's built-in voice-

control assistant *Siri*, *Hey! Google* on android phones or the Amazon's *Alexa* assistant. These devices learn to recognize their owner's voice and perform actions according to their owners' commands. Businesses use of AI has been rising over the last decade, customer service is another industry where AI is indisputably popular, from businesses' pages having AI-powered chat bots, to fully AI-driven phone call assistance. For these reasons, it is often argued that Artificial Intelligence aims to replace humans in any industry; because of the power and versatility of AI [8].

Machine Learning (ML) is a subset of Artificial Intelligence, and is often used in computer science, due to the fact that ML specializes in data and specific algorithms that enable a computer or robot to perform human activities. It allows the computer or robot to learn based on patterns, just like a human does, and encourages the system to improve its learning with time. There are different ways in which a system learns and they can be classified into supervised learning, unsupervised, and reinforcement learning [1]. During unsupervised learning, the algorithm is given a set of data and it must recognize features and patterns by itself. K-means and hierarchical clustering are examples. During supervised learning, the data has already some labels which facilitate the algorithm to detect patterns or features. Decision trees, neural networks, and support vector machines are examples. And reinforcement learning is classified as semi-supervised learning. The algorithm is fed non-complete data and is forced to generate the missing data itself, kind of like its guessing. The reinforcement part comes when the algorithm is told whether or not their guess was right. Examples of this type of ML include Expectation-maximisation and support vector machines [2]

Machine learning models explained in more detail:

- Neural Networks: Net of interconnected processing 'Neurons' with multiple layers which aim to mimic a human-neuron. This algorithm can be used to recognize patterns in vast amounts of data like image or speech recognition. There exist Traditional Neural Networks (TNN) in which each node is connected to the next one, and each input passes through every filter on each layer. There's also Convolutional Neural Networks (CNN) which are more useful when dealing with lots of data as grid-like data because CNN can assign a hierarchical order to the sequence of layers through which the data passes. TNN and CNN have different architectures but overall serve the same purpose [33].
- Linear regression: Is an algorithm highly useful when dealing with numerical data, this technique predicts future values based on the linear relationship of a given set of data. The objective of linear regression is to identify the best-fitting between the given and predicted data [30].



**Fig 6:** Linear regression example<sup>[30]</sup>.

- Clustering: this is an unsupervised model, the algorithm is trained to identified patterns on its own and output the group of related data. It is helfult compare and contrast data more reliably as clustering scrutinized the data, contrary to that of a human data analyst<sup>[30]</sup>.

**Table 2:** Comparison of different ML Models<sup>[3]</sup>

ML Model	Type of learning	Type of data	Application Example
Neural Networks	Unsupervised	Raw abstract, unstructured	Image or speech recognition
Linear Regression	Supervised	numerical	Stock prices prediction
Clustering	Unsupervised	Unlabled, patterned	Feature recognition
Decision trees	Supervised	Numerical/categotical	House-price prediction
Random forrests	Supervised	Numerical/categorical	Medical diagnostics

### 1.3. ML and AI applied to trading

Machine learning has been rising over the last couple of years, making experiments with different models applied to the stock market more commonly. It can be used as an investigative tool to identify patterns based on past data to invest in the future, aim to predict future stock prices, or even just be AI-powered tools that make investing easier. Sentiment analysis is a common application, Natural Language Processing (NLP) is used for these purposes. It essentially means that a computer is able to compare different sources of information like news, articles, social media, advertisements and extract the overall ‘sentiment’ or common attitude towards a particular stock<sup>[6]</sup>. Financial analysts use this to evaluate their future investments. A recent real-world example was the change of Facebook’s name to Meta along with its change in company structure. As a result in 2021, its stock, per single share decreased by 61% <sup>[37]</sup>.

Data forecasting live is another application. There are currently many ML algorithms that can be fed real-time data. For example, real time weather conditions influence the price of commodities like oil and gas. Political disputes and wars can affect supply chain and that would decrease the stock of certain companies. Feeding this info into a ML algorithm will adjust its predictions accordingly. Apart from making stock prices predictions, ML can be used to predict world conflicts based on past and current data. ML can also automate the trading process, High-frequency trading machines are an example. These AI-based machines can make transactions for you based on input interests, values and short advantageous time changes<sup>[6]</sup>.

#### 1.3.1. Current AI-based trading tools:

Focused on stock market investing aid, AI powered tools can perform three main tasks; automated trading in which either the bot makes the transaction for you or tells you when to do it. Educate based on real-time data; these platforms mimic a broker and offer a simulation platform for beginners. And finally, provide analytical tools like NLP, Sentiment Analysis, Forecasting and other methods to predict future stock prices and events that will most likely influence the stock/ETF. Below some examples of such tools will be discussed.

*Trade Ideas* and *Trend Spider* are automating trading platforms which offer trading bots, data-visualization tools, and market alerts to help investments.<sup>[28]</sup> *Composer trade* offers a simulation platform in which beginner clients can mimic trading with real world and real time prices. They also offer AI-powered financial advice and analytics. Their trading algorithms are customizable, and no coding experience is required. <sup>[12]</sup> *EquiBot* is an analytics FinTech company which uses AI to plan client’s investments and use ML to align the stock prices <sup>[7]</sup>

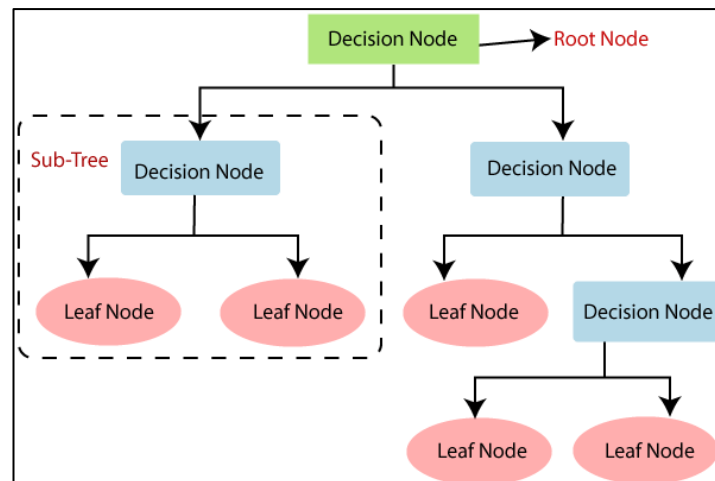


## 2. Methodology Selection

The aim of this work is to use machine learning to predict the Stock market to maximize the profit on investments. To do so, Random Forest model was used, this is essentially a machine learning model based on the ensemble learning principle, it means that is made up of several instances of another model which in this case is decision trees.

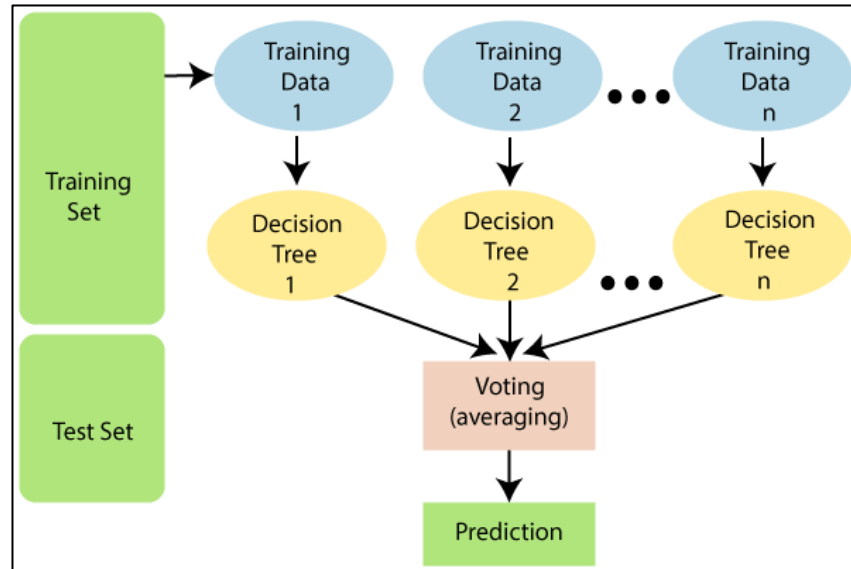
Decision trees: It is basically a model that consists on several true or false statements that check if relations between the predictors of a dataset have some kind of relationship or not, it can be visually represented by a flowchart in which each square or so called decision nodes ask a question that has a true or false output and based on this result it branches out into leaf nodes that are the final outcome of the classification or regression performed<sup>[14]</sup>.

On figure 7 a desition tree is presented where we can see the initial decision node or also called the root node and it splits into other decision nodes until all the algorithm has iterated over all the classifiers in the dataset, giving as a final output the Leaf nodes.



**Fig 7:** Decision tree example<sup>[14]</sup>.

Random Forest: This kind of classifier is a set of desition trees that trains the model using subsets of data on the given dataset, when individual values from each desiton tree is obtained they are averaged together to obtain a final prediction or in other words the output, with this classifier the higher number of trees used provides a better accuracy and decresses the overfitting the predictions. Some of the advantages of this model is that it can be trained in a shorter time when comparing it to other ML models, and it also has a good accuracy on the predictions even when the dataset is missing some information<sup>[20,34]</sup>.



**Fig 8:** Random Forest Flowchart<sup>[20]</sup>.

Random forests are especially useful for classification and regression; therefore it was the most suitable for the application of predicting the prices of the S&P500 and SPY ETF within the stock market. The tools used to carry out the project were Python programming language, interactive online tool Jupyter Notebooks, and code editor Visual Studio because of its versatile plug-ins. Python programming language was selected rather than other languages like C, C#, or C++ because of its simplicity, multipurpose application, and massive documentation and support platforms tailored for machine learning.

The starting point, the code to implement the ML model, train it, and generate results was adapted from Vik Paruchuri, data engineer at Dataquest company based in San Francisco, California USA <sup>[21]</sup>. Dataquest is an educational online platform which enables anyone to learn data science from scratch with many courses focused on python programming language. Along with their courses, data engineers and educators like Paruchuri share their projects and repositories online via Github with open access for public use. For this work the code from his 'sp\_500' project was modified and adapted for the project's needs .

The SPY ETF's data was extracted from Yahoo Finance. It is a website which contains real-time data on the stock market and relevant news which influences it. The data for S&P 500 and SPY was extracted from Yahoo Finance using its Application Programming Interface (API) compatible with python <sup>[4]</sup> . All the historical data for the S&P500 and SPY was downloaded from which the last 30 years were used for the modeling process of this prediction tool. The S&P500 dataset was utilized to show its relationship with the SPY , as the SPY is derived from it.

Scikit-learn is another open-source library for using ML in python language, this provides a huge set of tools to train AI, ML, and deep learning. It also gives an overview of all the available ML models and algorithms as well some examples of what areas and projects they are used for. Random forest algorithm was taken from this library and a precision score tool on scikit-learn was also used to compare the desired target against the predictions made by the model. This is referred to as a 'confidence score'.

Comparison between "Buy & Hold" vs. "Daily Trading" strategies was also done using the random forest classifier predictions to mimic the "Daily Trading" strategy as discussed in the analytical investigation. The reason why this model was used to mimic the daily trading strategy is because as mentioned in section 1, by daily trading a share of an ETF according to the price's fluctuations, the profit is proved to be higher than when just buying a share and leaving it to fluctuate with time. Still, this daily trading strategy incurs more risk and one must know and carefully use technical analysis when the time to buy or sell is optimal. This is where machine learning jumps in, to prove this, the model was given initially an 'investment budget of 1000 euros to spend in the SPY ETF starting in 1994. The number of shares obtained in 1994 with 1000 euros was 26.036106873011494 shares. Then, this number was compared to the value of those shares in 2024 to determine the increment and profit gained over that period.

## **2.1. Libraries included.**

NumPy: It's a python library that allows data array manipulation and has different tools to manage multi-dimensional vectors and matrices along with mathematical functions.

Pandas: Python library to manipulate and analyze huge data sets conformed by arrays. Also allows data structures and numerical table manipulation with time series.

Operating System: Library that allows python to communicate with computer terminal to read and write files on the system. It also offers the possibility to control and automate actions on system.

Matplotlib: library that allows the user to plot numpy and pandas' data into graphs for 2D data visualization and facilitate analysis. Outputs many plots available in different file formats.

Plotly: it's a library built over matplotlib that allows to plot interactive graphs for better analysis of data. Allows the user to zoom, pitch, and hover over the graph for easier data extraction. Some dependencies are necessary like *cufflinks* and *chart\_studio.plotly*. For Jupyter notebooks integration, the nbFormat library has to be included.

Sklearn: it's the actual library that has different sections of which one is RandomForestClassifier (ensemble) to be able to train the model and Precision\_score (metrics) to evaluate accuracy of the model. Machine learning library with plenty of algorithms.

Yfinance: API provided by yahoo finance that allows python to access tickers (different assets on the market) From Yfinance all data was extracted.

The following figures 9 and 10 show the code that was written to import the previously mentioned libraries. As a common practice to speed up the workflow when programming, they are imported with shortened names for convinience.

```
import yfinance as yf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os

%matplotlib inline

#Libraries for interactive vizualisation

import mplfinance as mpf
import cufflinks as cf #plotly dependency
import chart_studio.plotly as py #plotly dependency
from plotly.offline import download_plotlyjs,init_notebook_mode,plot,iplot

import plotly.graph_objects as go
from datetime import datetime
init_notebook_mode(connected=True)
cf.go_offline()

✓ 0.0s
```

**Fig 9:** Python libraries Imported

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import precision_score
```

**Fig 10:** Sci-kit learn library integration.

## 2.2. Model Parameters

The historical data extracted from the SPY ETF looks as depicted in table 3. Similarly, as it was shown in the etoro platform figure, it can be observed that the given data set provides five main features unique to each date that will be used by the classifier to identify relations and patterns. This is the data which will be the input to the model. For each day starting from 1994, the ETF share has an Open , Low , High , Close , and Volume value, all of which are numerical data, in currency. Volume refers the average number of shares of the ETF traded on that date. These terms have been presented and described in section 1.1.2.

**Table 3:** Model Data Input Example

Date	Open	Low	High	Close	Volume
1994-05-02	X0	Y0	Z0	T0	H0
1994-05-03	X1	Y1	Z1	T1	H1
...	...	...	...	...	...
2023-04-26	Xi	Yi	Zi	Ti	Hi

Pandas library allows modifying the original data set, to add, move, extract, delete or perform mathematical operations among columns. In this way, new columns were created to determine which is the target that is desired for prediction (to know whether the following day the price will increase or decrease); later add prediction column to be compared with target.

These input features (open, low, high, close etc) provided to the model are also referred as predictors that will influence the model's accuracy. Finding relationships and patterns among the predictors is a good practice, because by identifying which predictors relate to the other, later these new predictors can be fed to the model for training and in that way, the model can improve its predictions. To improve the initial model, backtesting the model itself was necessary to increase accuracy for better predictions. This consists on retraining the model on fewer data, then try to predict future data. For example you train the data for 10 years to predict the 11th year. You retrain the data of 11 years and predict the 12th and so on and so forth. This increases the model's accuracy<sup>[10]</sup>.

The random forest model has three main input parameters which influence its predictions. These are `n_estimators`, which is the number of decision trees the random forest is based on. The second one is `min_sample_splits` whis determines the over-fitting factor of the model, the higher it is, the less accurate model is but less prone it is to over-fit. And finally, `random_state`, this parameter only ensures that randomness is kept on each iteration of the trials, to ensure that the output changes based on parameters themselves and not the random data generated.

To determine which parameters, provide the optimal trained model the previous mentioned `n_estimators`, and `min_sample_splits` are being modified, performing different combinations to find the highest precision score. Once this precision score is determined, those parameter are chosed to train the Random Forest Classifier. On table 4 we can visualize how are this values going to be presented.

**Table 4:** Random Forest model parameter selection

	<b>Estimators (Decision trees instances)</b>				
<b>Min. sample splits</b>	100	200	300	400	500
100	PS(100,100)	PS(200,100)	PS(300,100)	PS(400,100)	PS(500,100)
150	PS(100,150)	PS(200,150)	PS(300,150)	PS(400,150)	PS(500,150)
200	PS(100,200)	PS(200,200)	PS(300,200)	PS(400,200)	PS(500,200)
250	PS(100,250)	PS(200,250)	PS(300,250)	PS(400,250)	PS(500,250)
300	PS(100,300)	PS(200,300)	PS(300,300)	PS(400,300)	PS(500,300)

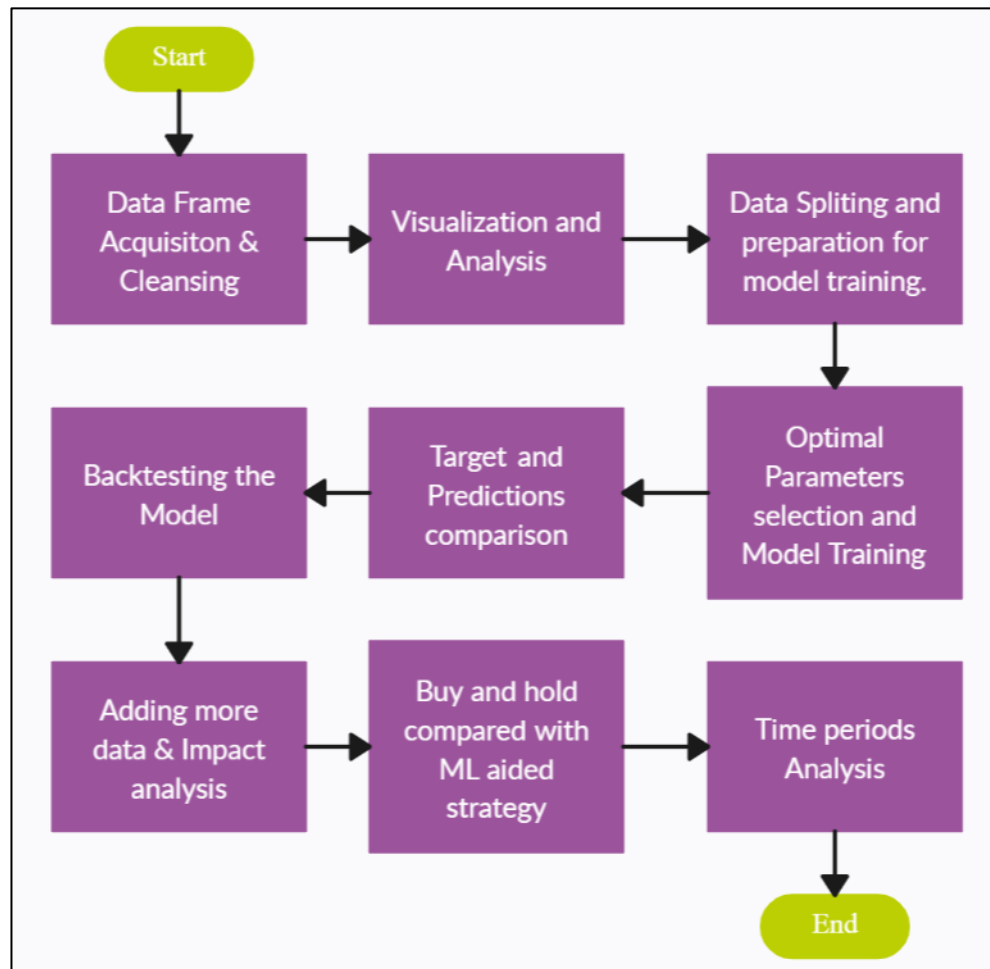
### **3. Experimental Investigation**

To proceed with the experimental part of this research, an algorithm was created to address the flow of the program. This algorithm describes the step-by-step on how the code was modified to predict the SPY price. Overall, nine steps were necessary to complete the task.

Starting by extracting the datasets from Yahoo Finance, afterwards data not relevant to this investigation was removed from the dataset and to visualize the information acquired the matplotlib and ipplot libraries were used to graphical illustrate the data and perform a visual inspection on the relationship between the two datasets acquired, data is then prepare by splitting it into a training and testing, and the initial model is then built and iterations over itself are done providing different parameters on each iteration to show the different precision scores obtained. Once the optimal parameters were selected, the model was trained once again and the predictions obtained were compared with the desired targets. A backtesting method is used to improve the prediction confidence of the model and then new features are created to further improve the predictions. Lastly a comparison between a buy and hold trading strategy with the model mimicking a daily trade strategy is provided, to determine the profitability of these two strategies.

#### **3.1. Research Roadmap**

Figure 11 shows an overview of the steps taken to complete the aim. In total, the project completion can be divided into 9 sequential steps that were followed in the same way in Jupyter Notebook. Each step is shortlry named and displayed as a block in the roadmap, the roadmap is for better visualization for the reader.



**Fig 11:** Research Roadmap Flowchart

The first step was the data acquisition and cleansing, in order to use a machine learning model vast amount of data are necessary. The data in subject was S&P 500 and SPY historical data from their start until the present year. The data was extracted from Yahoo finance and it included the shares' open, close, high, low and volume prices. It also included adj. average and dividends for both ETF's however, they were excluded as they were not needed for the scope of the project.

Next, the acquired data was plotted in order to get better visualization, deeper comprehension about the relationship between S&P 500 and SPY, and according to this make further analysis.

After analyzing the relationship between both ETF's data splitting was completed. This means that the whole data was separated into training data that was fed to the machine learning model, and testing data that was used to validate the model's accuracy. Its important to not feed all the data into the model to be able to evaluate its performance with real existing data, in this way we do not blindly trust the model's predictions.

Once data was fed, the parameters for the model training could be selected to have the best performing model. This step is important because different combinations of parameters result in different



predictions and confidence scores. Many combinations had to be performed in order to choose the optimal one which would output best results.

After the optimal parameters were selected and model was trained, evaluation of the results was next step. During this step, the set targets were compared to the specific targets. It is important to remark that the kind of predictions being done is the change of the value of the asset from one day to the next, resulting on a target of one or zero, one being a price increase and zero a price drop regardless of the numerical value. The model is not meant to predict numerical data because stocks are greatly influenced by external day-to-day events so its not viable for a model to predict exact value. What can be predicted from such volatile data is whether the trend will shift from upwards to downward and vice versa.

The following step is backtesting the model. It is essential to backtest the model to make sure that the capability of predicting these trends is accurate. This is done to evaluate the model's performance in real world scenarios. For example, one retrains the model with ten years data and expects it to predict the eleventh year's data; after, the model is retrained with the eleven years data for it to predict the twelfth year and so on and add these results to the dataset for an improved confidence level at the end.

The next block is adding more data and conducting impact analysis to determine the impact of the added data to the confidence score of the model. To have more data for the model, relations and trends can be found between the already given data that can be given to the model for learning. It is worth noting that it is not always beneficial to add more data to the model; for example, in stock market context, economy can drastically change from one day the other and that could make the previous data given to the model for training completely invalid. However, for this project adding more data is good to consider as it may influence the model positively, as the data of trends from the previous two, ten, and thirty days was used. Since the behavior of stock market data is not linear and follows different trends in short periods of time.

Finally, the best obtained model can be used to predict which trading strategy could be better; comparing a buy and hold strategy, for which the amount of shares bought on 1994 were 26.036106873011494 with a value of 38.4082 Euros, and daily trading strategy aided by the trained machine learning model for which the same amount of shares were bough on the same initial date and were traded based on preceding days to expected price rise.

To conclude, time period analysis was conducted. During this step, gains predicted were compared to real gains per year. Where there were clear discrepancies between predicted and actual gains, analysis of real-world events was done to explain the influence on the changing trend.

### 3.2. Model Algorithm to Code

Translating the algorithm into python programming language went as follows:

#### 3.2.1. Data Acquisition & Cleansing

First, data for the machine learning model must be available. Figure 12 shows the code to extract the SPY and S&P500 ETF's from yahoo finance. There's two main steps for this and its coded using an if/else statement.

If/else statement. It checks whether there is already a path in the computer which contains the data. Otherwise, it retrieves data from Yahoo finance and stores it in a .csv file. Two files were donloaded, one file containing all the available data of the SPY an S&P500, and another file extracting just the last 30 years. It is necessary to extract all the historical data for better understanding and comprehensive visualization. This data then is read and interpreted as the dataframe that is going to be used with the pandas library.

```
if (os.path.exists("sp500.csv") & os.path.exists("sp500all.csv")
    & os.path.exists("spy.csv") & os.path.exists("spyall.csv")):

    sp500 = pd.read_csv("sp500.csv", index_col=0)
    sp500all = pd.read_csv("sp500all.csv", index_col=0)
    spy = pd.read_csv("spy.csv", index_col=0)
    spyall = pd.read_csv("spyall.csv", index_col=0)
else:

    sp500 = yf.download("^GSPC",period='30y')
    sp500all = yf.download("^GSPC",period='max')
    sp500all.to_csv("sp500all.csv")
    sp500.to_csv("sp500.csv")

    spy = yf.download("SPY",period='30y')
    spyall = yf.download("SPY",period='max')
    spyall.to_csv("spyall.csv")
    spy.to_csv("spy.csv")
```

✓ 0.7s

Fig 12: Code to request data from Yfinance API

#### 3.2.2. Visualization and Analysis

The datasets acquired were assigned as a panda's dataframe; for data processing and analysis visualization is necessary. Matplotlib library aided in plotting the mountain graphs shown in figure 14, where the y-axis is the closing price of the share in any currency, and the x-axis is is the time in

days. Since the SPY is based on the S&P 500, both exchange traded funds (ETF's) were plotted on the same graph. This showed that SPY is a scaled down version of the S&P 500, price wise.

```
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12, 6))

sp500all.plot.line(y="Close", use_index=True, ax=axes[0,0], color='Blue', label='S&P500')
spyall.plot.line(y="Close", use_index=True, ax=axes[0,0], color='Orange', label='SPY')
axes[0,0].set_title("S&P500 and SPY All Historical Data")
axes[0,0].set_ylabel('Closing Price')
axes[0,0].set_xlabel('Date')

sp500.plot.line(y="Close", use_index=True, ax=axes[0,1], color='Blue', label='S&P500')
spy.plot.line(y="Close", use_index=True, ax=axes[0,1], color='Orange', label='SPY')
axes[0,1].set_title("S&P500 and SPY over the last 30 Years")
axes[0,1].set_ylabel('Closing Price')
axes[0,1].set_xlabel('Date')

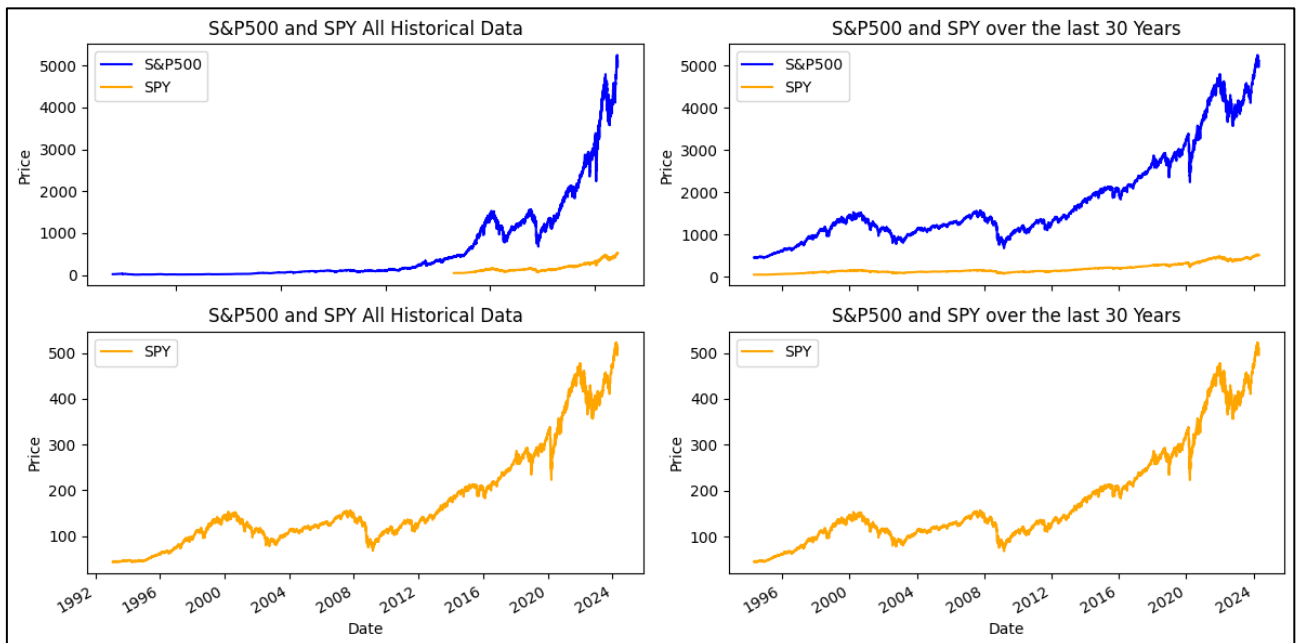
spyall.plot.line(y="Close", use_index=True, ax=axes[1,0], color='Orange', label='SPY')
axes[1,0].set_title("S&P500 and SPY All Historical Data")
axes[1,0].set_ylabel('Closing Price')
axes[1,0].set_xlabel('Date')

spy.plot.line(y="Close", use_index=True, ax=axes[1,1], color='Orange', label='SPY')
axes[1,1].set_title("S&P500 and SPY over the last 30 Years")
axes[1,1].set_ylabel('Closing Price')
axes[1,1].set_xlabel('Date')

fig.autofmt_xdate()
plt.tight_layout()
plt.show()
```

**Fig 13:** Code written to obtain a visualization graph

Figure 14 shows the data that was extracted to compare the S&P500 with the SPY. In this figure one can appreciate the relationship between these two assets of the stock market. The two graphs on the right side of the figure present all the historical data available to these assets; on the right the other two graphs present the historical data for this assets for the last 30 years only. The two graphs at the top present a clear comparison between the values of these assets showing that the S&P500 is bigger than the SPY by an approximate factor of ten. On the bottom, the two images show only the SPY on an appropriate scale, this allows us to appreciate the shape of the data on the graph in a better way.



**Fig 14:** Graph plotted on Jnotebook to compare SPY with S&P500

According to the information obtained from this analysis we can determine that the target aimed to predict is the value of the asset on the next day. The kind of event that we are looking for, is if the value of the asset is going to be higher tomorrow compared with today.

Based on this inference the following code was used to compare this condition and the result is assigned to a new column which is called *target*, if the condition is met the value returned is 1, otherwise if the condition is not met the value is going to be 0. Therefore, regardless value increment of the shares, the target being predicted will be if there is a positive change in the value.

On the figure 15 provided we can see the code that was written to achieve this action and later it can be seen how it is reflected on table 5.

```
#adding new column to the dataset
#(.shift -shifts all the data on the column one position upwards)
spy["Tomorrow"] = spy["Close"].shift(-1)

#adding a new colum as target
#(Target is the value of tomorrow only if the closing of previous day is lower than tomorrow)
spy["Target"] = (spy["Tomorrow"] > spy["Close"]).astype(int)
spy
```

**Fig 15:** Code to set the target of the investigation.

**Table 5** Addition of the target column to the dataset

	Open	High	Low	Close	Adj Close	Volume	Tomorrow	Target
Date								
1994-05-02	45.093750	45.718750	44.937500	45.375000	26.367529	275000	45.328125	0
1994-05-03	45.390625	45.406250	45.062500	45.328125	26.340273	183400	45.250000	0
1994-05-04	45.421875	45.421875	45.078125	45.250000	26.294888	401900	45.187500	0
1994-05-05	45.296875	45.375000	45.187500	45.187500	26.258574	659800	44.750000	0
1994-05-06	44.968750	44.968750	44.593750	44.750000	26.004328	216300	44.359375	0
...	...	...	...	...	...	...	...	...
2024-04-23	501.779999	506.089996	499.529999	505.649994	505.649994	64633600	505.410004	0
2024-04-24	506.559998	507.369995	503.130005	505.410004	505.410004	55928100	503.489990	0
2024-04-25	499.179993	504.269989	497.489990	503.489990	503.489990	69122400	508.260010	1
2024-04-26	506.350006	509.880005	505.700012	508.260010	508.260010	63283200	510.059998	1
2024-04-29	510.089996	510.750000	507.250000	510.059998	510.059998	45916800	NaN	0

Table 5 shows the updated data, the only difference between the initial data and this one is the two added columns of *Tomorrow* and *Target*. The *Tomorrow* column is added so the comparison between the previous day closing price and is made, and the *Target* column can be created.

### 3.2.3. Data Splitting for model training.

To predict future events, the machine learning model has to be trained with data. Now that the input data has been extracted and cleansed, we can move on with the data preparation in order to test the model after the training stage. Some data from the initial dataset was excluded for testing the predictions of the model. The initial data is from 30 years, however as the market is only open during working days, in one year there is data for 200 working days.

In this case the data distribution was as follows;

Total data: 6000 days (30 yrs)

Training data: 5700 days

Testing data: last 300 days

```
#Dataset Splitting & Predictors
train = spy.iloc[:-300]
test = spy.iloc[-300:]
predictors = ["Close", "Volume", "Open", "High", "Low"]
```

**Fig 16:** Code to split the dataset into train & test, predictors also shown

### 3.2.4. Optimal Parameters selection for ML model.

Now that the data was classified and split into training and testing data, we can focus on the actual random forest's parameter selection. The input parameters of the function are *n\_estimators*, *min\_sample\_split*, and *random\_state*. Estimators refer to the number of decision trees which the model will use to make the predictions. The sample splits prevent over-fitting of the model. Over-fitting happens when the model becomes too good at predicting past data (like the training data) but not efficient or flexible enough to predict future prices. And random state controls the randomness of the decisions done by the classifier. Figure 17 shows the code used for this step. There were five different values for estimators and sample splits to evaluate their influence of the model.

```
estimators = [100,200,300,400,500]
samples_splits = [100,150,200,250,300]
scores = []

for i in estimators:
    samples = []
    for j in samples_splits:
        model = RandomForestClassifier(n_estimators=i, min_samples_split=j, random_state=1)
        model.fit(train[predictors], train["Target"])

        preds = model.predict(test[predictors])
        preds = pd.Series(preds, index=test.index)
        accuracy = precision_score(test['Target'], preds)

        samples.append(accuracy)

    scores.append(samples)
```

**Fig 17:** Loops determine optimal parameters, *n\_estimators* and *min\_sample\_split* iterations.

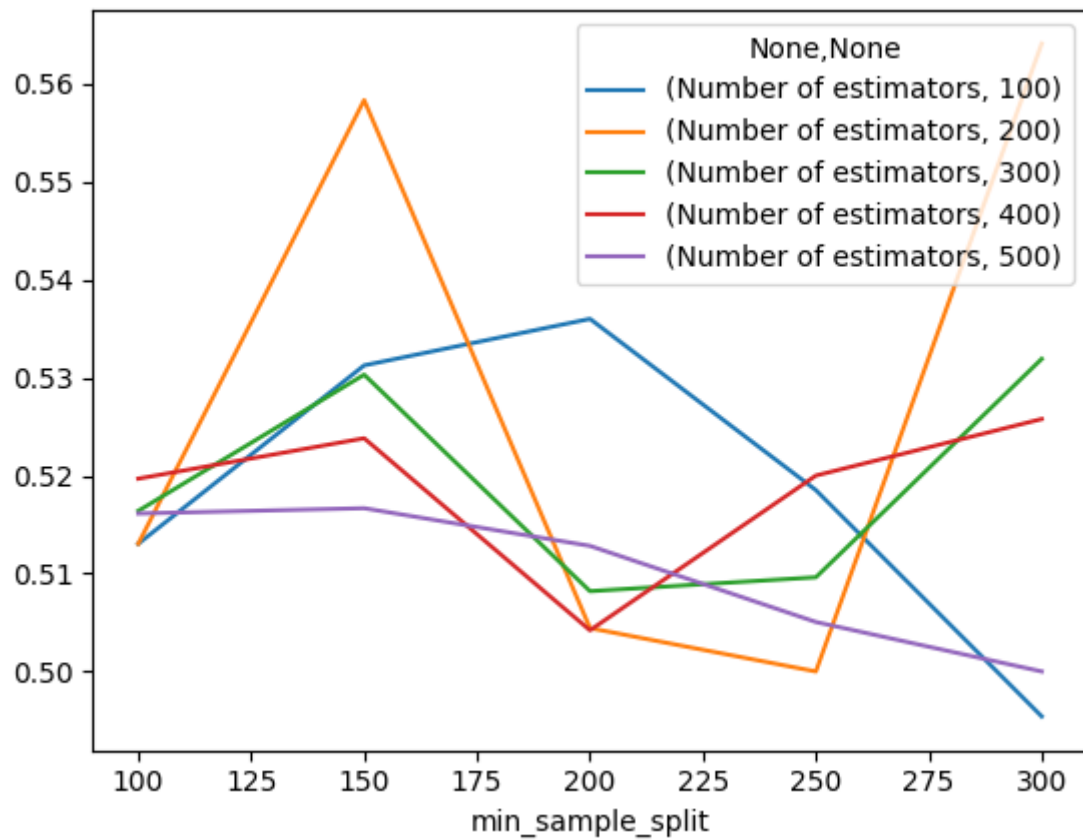
Figure 18 shows the results of each combination of parameters. The numerical data seen in the graph is the model's confidence score. The confidence score is the score that determines the accuracy in the prediction of future classifiers, as a percentage. An ideal confidence score would be 100%, however that is impossible to do. So the aim is to obtain the highest confidence score. As its seen from the values, most of the combinations are around 52%, quite a low confidence score; hence, the graph in figure 19 was plotted based on the data obtained from figure 18.

```
scores_comparison = pd.DataFrame(scores,samples_splits,estimators)
scores_comparison
```

	100	200	300	400	500
100	0.539823	0.540984	0.512821	0.532110	0.545455
150	0.522124	0.523077	0.504202	0.519231	0.504673
200	0.530435	0.543307	0.508333	0.509615	0.505051
250	0.517241	0.531250	0.512397	0.504673	0.510000
300	0.529412	0.531250	0.495798	0.504673	0.504950

**Fig 18:** Precision Scores results table.

When selecting 200 estimators, the highest results are obtained for all the minimum sample split instances therefore the recommended combination of parameters would be 200 estimators and 200 minimum sample splits. However, the data shows that using 500 estimators and 100 minimum sample splits, gives the highest confidence score. It is worth mentioning that this was the longest step of the algorithm as it took 15 minutes to run.



**Fig 19:** Presicion scores results graph.

### 3.2.5. Training the model

Now that the optimal parameters for the random forest classifiers were determined, they are inputted into the function and the model is trained. The results are shown in figure 20, the total confidence score of the model was 0.5641, meaning a confidence score of about 56%.



```
model = RandomForestClassifier(n_estimators=200, min_samples_split=300, random_state=1)
model.fit(train[predictors], train["Target"])
✓ 11.3s

RandomForestClassifier
RandomForestClassifier(min_samples_split=300, n_estimators=200, random_state=1)

preds = model.predict(test[predictors])
preds = pd.Series(preds, index=test.index)
precision_score(test['Target'], preds)
✓ 0.0s

0.5641025641025641
```

**Fig 20:** Optimized model Training.

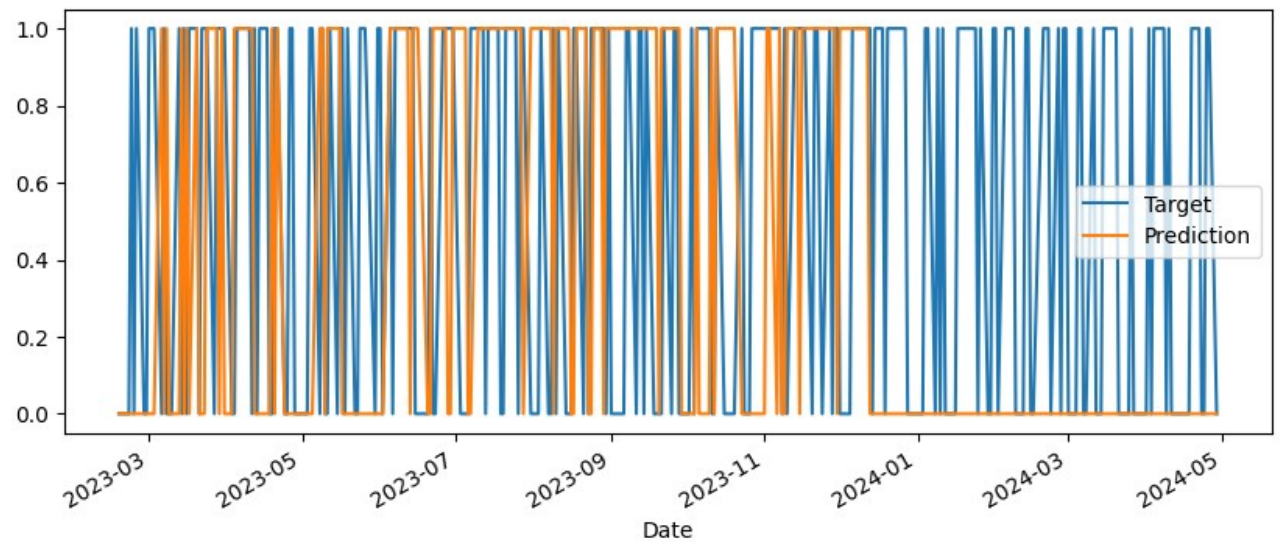
### 3.2.6. Compare Target with Predictions

The predictions results obtained are compared with the targets previously defined, and plotted to show when did the prediction matched the target. The code written to achieve this is shown in figure 21 where a concatenation function was utilized to contrast this data.

```
combined = pd.concat([test["Target"], preds], axis=1)
combined.rename(columns = {0:'Prediction'},inplace= True)
combined.plot(figsize=(10,4))
```

**Fig 21:** Target vs Predictions concatenation.

This code provided the following graph in figure 22, the way the information in this graph is being interpreted, is where the orange line matches the blue line at the top or at the bottom, being 1 the days in which it predicted a growth in the price and 0 the days that the price decreased.



**Fig 22:**Target vs Predictions concatenation graph.

### 3.2.7. Model Backtesting

To better predict the model two main functions were defined, the first one is a function that expects a already splitted dataset, the predictors of the dataset and the already trained model, when this data is provided it fits the model with the data, then test this result and finally it returns the amount of days it was able to predict. Then a second function was created and following the back test methodology it was modeled to use a certain amount of the data to train and then predict the next year, retrain itself and predict the following year and so on. This is done to increase the prediction confidence. These two functions are presented above in figure 23.

```

def predict(train, test, predictors, model):
    model.fit(train[predictors], train["Target"])
    preds = model.predict(test[predictors])
    preds = pd.Series(preds, index=test.index, name="Predictions")
    combined = pd.concat([test["Target"], preds], axis=1)
    return combined
✓ 0.0s

def backtest(data, model, predictors, start=2500, step=250):
    all_predictions = []

    for i in range(start, data.shape[0], step):
        train = data.iloc[0:i].copy()
        test = data.iloc[i:(i+step)].copy()
        predictions = predict(train, test, predictors, model)
        all_predictions.append(predictions)

    return pd.concat(all_predictions)

```

**Fig 23:** Prediction and backtesting Functions.

After utilizing the backtesting function it was seen that we were able to predict 3062 days in which the price might increase, with a confidence level of 54.8% as it can be seen in figure 24, and it was able to successfully predict a 54.3% of the target days where the price is increasing.

```
predictions = backtest(spy, model, predictors)
✓ 2m 22.7s

predictions["Predictions"].value_counts()
✓ 0.0s
Predictions
1    3285
0    1769
Name: count, dtype: int64

precision_score(predictions["Target"], predictions["Predictions"])
✓ 0.0s
0.5452054794520548

#Prediction Percentages
predictions["Target"].value_counts() / predictions.shape[0]
✓ 0.0s
Target
1    0.543134
0    0.456866
Name: count, dtype: float64
```

**Fig 24:** Back Testing Results

### 3.2.8. Adding more data to improve accuracy.

To further improve the model, the addition of new features to the dataset can be done by finding relations between the time frames of the data, also called horizons. These horizons are the timeframes of 2, 5, 60, 250, and 1000 days. In the following figure 25, these horizons are used to acquire the average of each time frame and further use it to get the ratio between the closing values and these averages. Then another column is added with the appearing trends on this data.

```

horizons = [2,5,60,250,1000]
new_predictors = []

for horizon in horizons:
    rolling_averages = spy.rolling(horizon).mean()

    ratio_column = f"Close_Ratio_{horizon}"
    spy[ratio_column] = spy["Close"] / rolling_averages["Close"]

    trend_column = f"Trend_{horizon}"
    spy[trend_column] = spy.shift(1).rolling(horizon).sum()["Target"]

    new_predictors+= [ratio_column, trend_column]

)

spy = spy.dropna(subset=spy.columns[spy.columns != "Tomorrow"])

```

**Fig 25:** Addition of new features to dataset

Once again, the previous backtesting method is used to feed the new information into the model. This time only acquiring successful predictions for 327 days where the price increased.

```
predictions = backtest(spy, model, new_predictors)
✓ 1m 1.4s

predictions["Predictions"].value_counts()
✓ 0.0s

Predictions
0.0    3916
1.0     137
Name: count, dtype: int64

precision_score(predictions["Target"], predictions["Predictions"])
✓ 0.0s

0.5547445255474452

predictions["Target"].value_counts() / predictions.shape[0]
✓ 0.0s

Target
1    0.544535
0    0.455465
Name: count, dtype: float64
```

**Fig 26:** Backtesting after features addition

### 3.2.9. Predictions Result Comparison with “Buy and Hold” Strategy.

Figure 27 shows the function created to analyze the buy and hold strategy and its profitability results if stocks were bought thirty years ago. The initial investment provided was of one thousand (1000) euros, then they were converted into USD to be traded on the stock market and the actual value of the stocks acquired is presented. For reference and comparison purposes, the same strategy was evaluated with different investment budgets; one-hundred, ten-thousand, and one-hundred thousand euros respectively.

The results were as follows; thirty years ago, if a person invested one thousand euros on the SPY ETF, the person would have acquired 26.036106873011494 shares. If today that person would like

to sell those shares they are worth 13312.0013 euros, showing a profit of 12,312.001 euros. For the other initial investment options is the same value multiplied according to their corresponding factor of 10.

```
# Buy and Hold Strategy
#years30_ago_price = spy['Close'][0]
#actual_price = spy['Close'][-1]
euro_dollar = 0.8488
investment = [100,1000,10000,100000] # Initial investment

buy_and_hold = []
for i in investment:
    shares = (i/euro_dollar)/years30_ago_price
    value_added = actual_price*shares
    buy_and_hold.append(value_added)

buy_and_hold
```

✓ 0.0s

```
[1331.2001305578547,
 13312.001305578546,
 133120.01305578547,
 1331200.1305578544]
```

**Fig 27:** Buy and Hold Function

Now, the machine learning model implemented in the project acts as a tool to support an investor with daily trading strategy. This means that according to the result of the prediction, whether the next day price was going to rise or decrease, the investor would make the decision to sell and buy again the next day or vice versa.

Table 6 shows the confidence score of the model at different stages of the project. As it is seen, the confidence score was the highest in the initial model which did not include backtesting or additional data. Backtesting supports the predictions because it allows the model to better perform in real-world scenarios because it is like a reinforced version of the initial model.

**Table 6:** Confidence scores table for the different models trained.

	Confidence Score
Initial Model	0.5641
Including Backtesting	0.5452
Data Addition with backtesting	0.5547

Table 7 shows four columns; the ‘date’ column from month, day and year the stocks information is from, the ‘Target’ column shows the real price in the market, the ‘predicted’ column shows the predicted price by the ML model, and the ‘Signal’ column shows either 1 or 0. A value of 1 means that the model predicts that the share’s price will increase the next day, advising the investor to buy the same day and sell tomorrow to make some profit. A value of 0 is the contrary, the model predicts that the price of the share will drop the next day, advising the investor to sell the same day to not lose. Simulating these trades for the last thirty years with the same initial investment of one thousand euros, we obtained a profit of 12,149.015. A profit comparison between both strategies can be seen in table 8, above it the formula used to calculate the percentages if provided.

**Table 7:** Daily Trading with ML profitability

	Target	Predicted	Signal
Date			
2008-03-28	3418.280297	3435.985056	1
2008-03-31	3478.684255	3556.792576	1
2008-04-01	3568.248526	3559.135730	0
2008-04-02	3539.869265	3567.987911	1
2008-04-03	3570.070847	3564.082654	0
...	...	...	...
2024-04-26	13280.757660	13279.976608	0
2024-04-29	13240.922448	13069.605214	0
2024-04-30	13053.983391	13027.166233	0
2024-05-01	13126.103121	13149.015022	1



$$\text{Percentage gain} = \left( \frac{V_F - V_i}{V_i} \right) * 100$$

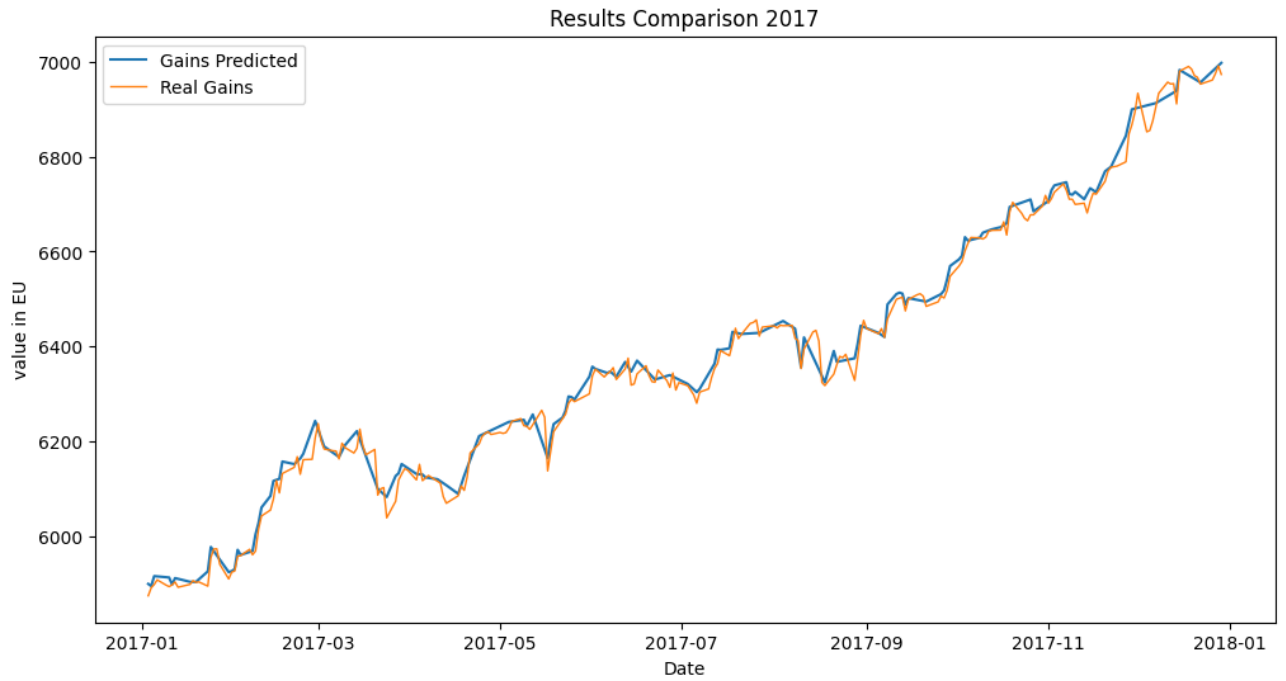
Where  $V_F$  is final value, and  $V_i$  is initial value invested.

**Table 8:** Strategies Comparison

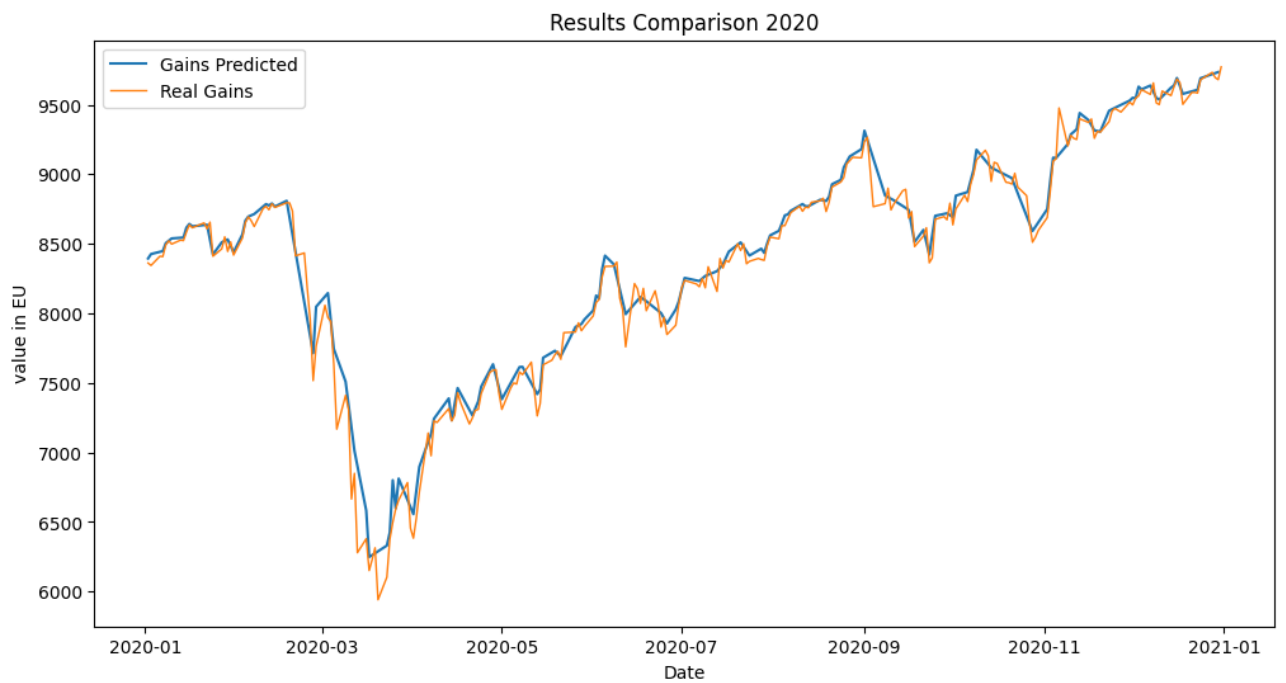
Trading strategy	Initial Investment, eur	Profit after 30 yrs, eur	Profit after 30 yrs, %
Buy and Hold	1000	12,312.001	1231,2%
Daily Trading	1000	12,149.015	1214,9%

After comparing both strategies, it can be seen that from this study, the buy and hold strategy generates more profit, the difference is small, buy and hold strategy is 16.3% more effective than the daily trading strategy. This can be because of different factors which influence this, there are many other datasets and extra tools that can be added to the model to increase its confidence score like overall sentiment. An example of sentiment influence in data is shown on figures 28 and 29. Figure 28 plots the data retrieved from the model, the predicted values vs the real market data. Here it is seen that even though the trained model made good predictions and follows the trend, its important to question the external factors that cause this trends.

In 2017 the price per share started out low, however throughout the year it kept rising at a constant rate. A major event that happened during 2017 which was of high interest to all the companies inside the SPY and S&P 500 was the election of the United States' president Donald Trump. It is no secret that Trump's personal interest involved pro-business policies, so it was expected by companies and investors to have a positive impact on this. Once Trump started his presidential period, he started reforming and adding tax cuts, the expected beneficial policies, and others including good's import tariffs and technologies<sup>[17,18]</sup>. The results of this actions can be seen on the presented upward trend in figure 28 Where the blue line represents gains predicted in the value of the shares of the SPY and the Orange line represents the real values that took place. It is worth mentioning that the model is unable to predict this trends for future dates, the reason why it is somehow getting close values is due to the fact that this is historical data that was used for testing. To show this in a better way, figure 29 shows the graph with a significant drop where the model is not that accurate making the prediction for, after investigation it was found that the major event that took place during this time was Covid-19<sup>[23]</sup>.



**Fig 28:** Real gain vs predicted gains in 2017.



**Fig 29:** Real gain vs predicted gains in 2020

## **Conclusions**

1. A general overview was provided about what the stock market is, its different areas, and trading assets. The scope of this project focused on two Exchange Trust Funds (ETFs) which were the S&P 500 and SPY. After understanding the stock market, ML & AI, and what instances of ML applied to the stock market already exist, it is evident that despite of the significant advance in the development of this technologies, the stock market is a field for which ML models can still be further tailored. Also, not all the ML models are fit to predict all the assets on the market, and analysis as the one performed in this investigation should be performed for other assets to find better suited predictions on specific assets.
2. There are a great number of machine learning algorithms that help predict the market's share price with common ones being reinforcement learning or supervised learning. Within these are the algorithms of linear regression, support vector machines, convolutional neural networks, decision trees, and the selected algorithm for this project: random forests.
3. All historical data from S&P 500 and SPY ETF's were downloaded from Yahoo finance, data from the last thirty years was extracted from it for the prediction of rise or drop in share prices. Data was separated according to training and testing data. Specific parameters were selected and backtesting was conducted for better model performance in real-world scenarios. The implementation of the ML model was done using python programming language and Scikit learn libraries.
4. Once the random forest model was trained and tested, it gave a confidence score of 56%. Using this model, two different trading strategies were compared for profit; the buy and hold strategy and the daily trading strategy. The buy and hold strategy proved to be more effective and generating 16.3% more profit than daily trading.

## **Observations and Possible improvements:**

In order to improve the model's predictions and confidence scores it is advised to include more data into the model like investing sentiment data sets, also using other machine learning models to compare their effectiveness for the application.

## List of references

1. BOUCHEFRY, K. EL - SOUZA, R.S. DE Learning in Big Data: Introduction to Machine Learning. In *Knowledge Discovery in Big Data from Astronomy and Earth Observation: Astrogeoinformatics*. 2020. p. 225–249. [žiūrėta 2024-05-03]. . .
2. EDGAR, T.W. - MANZ, D.O. Machine Learning. In *Research Methods for Cyber Security* [interaktyvus]. 2017. p. 153–173. [žiūrėta 2024-05-03]. . Prieiga per internetą: <<https://linkinghub.elsevier.com/retrieve/pii/B9780128053492000066>>.
3. PEDREGOSA FABIANPEDREGOSA, F. ir kt. Scikit-learn: Machine Learning in Python. In *Journal of Machine Learning Research* [interaktyvus]. 2011. Vol. 12, no. 85, p. 2825–2830. [žiūrėta 2024-05-03]. . Prieiga per internetą: <<http://jmlr.org/papers/v12/pedregosa11a.html>>.
4. (2) Yahoo Finance: Overview | LinkedIn. In [interaktyvus]. [žiūrėta 2024-05-04]. Prieiga per internetą: <<https://www.linkedin.com/company/yahoo-finance/>>.
5. 4 Stages of the Economic Cycle | Britannica Money. In [interaktyvus]. [žiūrėta 2024-05-03]. Prieiga per internetą: <<https://www.britannica.com/money/stages-of-economic-cycle>>.
6. (25) Machine Learning for Trading — Can It Predict the Trend? | LinkedIn. In [interaktyvus]. [žiūrėta 2024-05-04]. Prieiga per internetą: <<https://www.linkedin.com/pulse/machine-learning-trading-can-predict-trend-datatobiz/>>.
7. About Us – QuantumstreetAI. In [interaktyvus]. [žiūrėta 2024-05-04]. Prieiga per internetą: <<https://quantumstreetai.com/about-us/>>.
8. Artificial intelligence (AI) | Definition, Examples, Types, Applications, Companies, & Facts | Britannica. In [interaktyvus]. [žiūrėta 2024-05-02]. Prieiga per internetą: <<https://www.britannica.com/technology/artificial-intelligence>>.
9. ASSET | English meaning - Cambridge Dictionary. In [interaktyvus]. [žiūrėta 2024-05-04]. Prieiga per internetą: <<https://dictionary.cambridge.org/dictionary/english/asset>>.
10. Backtesting - MATLAB & Simulink. In [interaktyvus]. [žiūrėta 2024-05-04]. Prieiga per internetą: <<https://se.mathworks.com/discovery/backtesting.html>>.
11. Best Stock Brokers for Europeans in 2024 | BrokerChooser. In [interaktyvus]. [žiūrėta 2024-05-03]. Prieiga per internetą: <<https://brokerchooser.com/best-brokers/best-stock-brokers-for-europeans>>.
12. Composer – Investing. Built Better. In [interaktyvus]. [žiūrėta 2024-05-04]. Prieiga per internetą: <<https://www.composer.trade/>>.
13. Convert your savings into investments | Luminor. In [interaktyvus]. [žiūrėta 2024-05-03]. Prieiga per internetą: <<https://www.luminor.lt/en/investments>>.

14. Decision Tree Algorithm in Machine Learning - Javatpoint. In [interaktyvus]. [žiūrėta 2024-05-03]. Prieiga per internetą: <<https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm>>.
15. ETFs - Swedbank. In [interaktyvus]. [žiūrėta 2024-05-03]. Prieiga per internetą: <<https://www.swedbank.lt/private/investor/stock/lyxor?language=ENG>>.
16. Forecasting: What It Is, How It's Used in Business and Investing. In [interaktyvus]. [žiūrėta 2024-05-02]. Prieiga per internetą: <<https://www.investopedia.com/terms/f/forecasting.asp>>.
17. Here's How the Stock Market Performed Under President Donald Trump. In [interaktyvus]. [žiūrėta 2024-05-12]. Prieiga per internetą: <<https://markets.businessinsider.com/news/stocks/stock-market-performance-under-president-donald-trump-dow-jones-sp500-2021-1-1029987163>>.
18. If Trump wins, here's how the S&P 500 could react - CBS News. In [interaktyvus]. [žiūrėta 2024-05-12]. Prieiga per internetą: <<https://www.cbsnews.com/news/if-trump-wins-heres-how-the-s-p-500-could-react/>>.
19. Inflation vs. Deflation: What's the Difference? In [interaktyvus]. [žiūrėta 2024-05-02]. Prieiga per internetą: <<https://www.investopedia.com/ask/answers/111414/what-difference-between-inflation-and-deflation.asp>>.
20. Machine Learning Random Forest Algorithm - Javatpoint. In [interaktyvus]. [žiūrėta 2024-05-03]. Prieiga per internetą: <<https://www.javatpoint.com/machine-learning-random-forest-algorithm>>.
21. project-walkthroughs/sp\_500 at master · dataquestio/project-walkthroughs. In [interaktyvus]. [žiūrėta 2024-05-04]. Prieiga per internetą: <[https://github.com/dataquestio/project-walkthroughs/tree/master/sp\\_500](https://github.com/dataquestio/project-walkthroughs/tree/master/sp_500)>.
22. SHARE | English meaning - Cambridge Dictionary. In [interaktyvus]. [žiūrėta 2024-05-04]. Prieiga per internetą: <<https://dictionary.cambridge.org/dictionary/english/share?q=shares>>.
23. S&P 500: impact of COVID-19 vs previous major crashes 2020 | Statista. In [interaktyvus]. [žiūrėta 2024-05-12]. Prieiga per internetą: <<https://www.statista.com/statistics/1175227/s-and-p-500-major-crashes-change/>>.
24. SPY Interactive Stock Chart | SPDR S&P 500 ETF Trust Stock - Yahoo Finance. In [interaktyvus]. [žiūrėta 2024-05-04]. Prieiga per internetą: <<https://finance.yahoo.com/chart/SPY>>.
25. Technical Analysis Basics | What is Technical Analysis? | IG US. In [interaktyvus]. [žiūrėta 2024-05-02]. Prieiga per internetą: <<https://www.ig.com/us/trading-strategies/beginners-guide-to-technical-analysis-190430>>.
26. TICKER | English meaning - Cambridge Dictionary. In [interaktyvus]. [žiūrėta 2024-05-04]. Prieiga per internetą: <<https://dictionary.cambridge.org/dictionary/english/ticker?q=tickers>>.

27. TRADE | English meaning - Cambridge Dictionary. In [interaktyvus]. [žiūrėta 2024-05-04]. Prieiga per internetą: <<https://dictionary.cambridge.org/dictionary/english/trade>>.
28. Trade Ideas: AI-Driven Stock Scanning & Charting Platform. In [interaktyvus]. [žiūrėta 2024-05-04]. Prieiga per internetą: <<https://www.trade-ideas.com/>>.
29. Trading Strategy - Overview, Components, How To Develop. In [interaktyvus]. [žiūrėta 2024-05-02]. Prieiga per internetą: <<https://corporatefinanceinstitute.com/resources/career-map/sell-side/capital-markets/trading-strategy/>>.
30. What are Machine Learning Models? In [interaktyvus]. [žiūrėta 2024-05-02]. Prieiga per internetą: <<https://www.databricks.com/glossary/machine-learning-models>>.
31. What Is a Broker? Definition, Examples and How to Find One - NerdWallet. In [interaktyvus]. [žiūrėta 2024-05-02]. Prieiga per internetą: <<https://www.nerdwallet.com/article/investing/what-is-a-broker>>.
32. What is a Candlestick? - 2022 - Robinhood. In [interaktyvus]. [žiūrėta 2024-05-02]. Prieiga per internetą: <<https://learn.robinhood.com/articles/3YzdYQ8bI4XqfnYUNj3dac/what-is-a-candlestick/>>.
33. What Is a Neural Network? - MATLAB & Simulink. In [interaktyvus]. [žiūrėta 2024-05-03]. Prieiga per internetą: <<https://www.mathworks.com/discovery/neural-network.html>>.
34. What is Artificial Intelligence (AI)? Everything You Need to Know. In [interaktyvus]. [žiūrėta 2024-05-02]. Prieiga per internetą: <<https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence>>.
35. What is Stock? | Types & Examples - Lesson | Study.com. In [interaktyvus]. [žiūrėta 2024-05-02]. Prieiga per internetą: <<https://study.com/academy/lesson/what-is-a-stock-definition-types-examples.html>>.
36. What Is the Stock Market and How Does it Work? In [interaktyvus]. [žiūrėta 2024-05-02]. Prieiga per internetą: <<https://www.investopedia.com/terms/s/stockmarket.asp>>.
37. What's in a Name? Meta Platforms Stock Down 61% Since Its Infamous Name Change | The Motley Fool. In [interaktyvus]. [žiūrėta 2024-05-04]. Prieiga per internetą: <<https://www.fool.com/investing/2023/01/10/what-name-meta-platforms-stock-down-since-change/>>.

## Appendices

### Appendix 1. Jupyter Notebook

On the following page the notebook on which the practical part of this investigation was developed is being presented in pdf format.



JNotebook.pdf