Bear Market Prediction Using Logistic Regression, Random Forest, and XGBoost

Pedro Jorge Bastos

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor: Prof. Rui Fuentecilla Maia Ferreira Neves

Examination Committee

Chairperson: Prof. Teresa Maria Sá Ferreira Vazão Vasques
Supervisor: Prof. Rui Fuentecilla Maia Ferreira Neves
Member of the Committee: Prof. Rui António Dos Santos Cruz

November 2019

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

Firstly, I would like to thank to the Instituto Superior Técnico and respective workers with whom I crossed paths, for receiving me and provide me the right environment to complete my academic formation. I would like to thank Professor Rui Neves as well, for the orientation regarding my Master's dissertation and for teaching me notions of a different area of study. Also, thanks to my thesis' colleagues with whom I had some meetings and other good moments with good company.

Still about college, I would like to give a special thanks to my group of friends from Casa Rosinha: Basile Belime, João Cruz, Diogo Teixeira, Cristian Sava, Rafael Figueiredo, Fernando Nunes and Isabela Correia, who stayed by my side along almost the entire academic journey, and with whom I spent not only good moments with great joy, but also, not that good times, in which supporting each other we ended up doing better and more successfully.

Another special thanks to my group ”of and for” always, ”Porcalhotos”, for providing me the best moments in life and for proving me that some things will never change, regardless time and context. Thanks to my bestfriend and girlfriend Beatriz Costa, the person who has seen and knows my worst and best sides and still remains by my side as one of my biggest supports, and for that I feel really grateful. I grew up a lot with you and a special thank for the ”Co-orientation”.

Finally, special thanks to all my family, and a special words to my parents, who have always supported me and I know that always will, and for guiding me in my disoriented moments, to my older brother Miguel, who is my biggest example in life, and to my uncle Raquel, who is passing through rougher times.

Resumo

A bolsa de valores é considerado um dos sistemas mais complexos do mundo, composto por muitos segmentos sem um padrão claro definido para as suas variações de preço. Vários fatores influenciam os seus movimentos, e por isso fazer previsões na bolsa de valores, não é considerada uma tarefa fácil.

Este trabalho, como uma aplicação na área financeira, tenta prever com uma antecipação de seis e doze meses, três diferentes quedas de preço no indíce americano ”Standard & Poor’s 500”: :(i) -20% (”Bear Markets”), (ii) -17,5% e (iii) -15%. Para isso, foram produzidos e treinados quatro diferentes modelos computacionais baseados em diferentes algoritmos de ”Machine Learning” - ”Logistic Regression”, ”Random Forest”, ”XGBoost” e o conjunto de todos os modelos - e tendo como base os dados históricos de diversas variáveis económicas.

Realizaram-se testes desde 1970 até 2019 e foi possível detectar a maioria das grandes quedas no S&P 500, em particular ao detectar os eventos com 12 meses de antecipação. O modelo do ”Logistic Regression” superou os restantes, obtendo melhores resultados e detectando as quedas de mercado com mais antecedência. A abordagem de que combina os algoritmos, foi considerada o método mais equilibrado, pois combinava os melhores resultados dos diferentes algoritmos. Com a implementação de um algoritmo genético, também foi possível otimizar os resultados do modelo ”XGBoost” para diferentes casos de teste.

Palavras-chave: ”Standard & Poor’s 500”, ”Bear Markets”, ”Logistic Regression”, ”Random Forest”, ”XGBoost”, Algoritmo genético.

Abstract

The stock market is considered one of the most complex systems in the world, consisting in many segments whose prices move up and down, without generating a clear pattern. Several factors influence its movements so predicting the stock market with traditional time series analysis can not be considered an easy task.

Hence, in this work an application of machine learning algorithms in the financial area is developed to predict several market downfalls, with 6 and 12 months of advance, in the Standard & Poor's 500 index: (i) -20% (Bear Markets), (ii) -17.5% and (iii) -15%. For that, four different computational models based in different Machine Learning algorithms were produced and trained - Logistic Regression, Random Forest, XGBoost and an Ensemble of the algorithms used.

Doing out-of-sample tests from 1970 to 2019, it was possible to detect most of the big downfalls in S&P 500, in particular when detecting the events with 12 months of anticipation. The Logistic Regression model outperformed the other models having greater results and detecting the market downfalls with more antecedence. The Ensemble approach (joining all algorithms), was considered the most balanced method since it combining the best results of the different algorithms. Through the implementation of a Genetic Algorithm it was also possible to optimize the results of the XGBoost model for different test cases.

Keywords: Standard & Poor's 500, Bear Markets, Logistic Regression, Random Forest, XG-Boost, Genetic algorithm.

Contents

List of Tables

List of Figures

List of Acronyms

AUC Area Under the roc Curve

AUTP Area Under True Positives

ANN Artificial Neural Networks

BOFAML Bank of America Merryl Lynch

CBOE Chicago Board Options Exchange

EPS Earnings Per Share

FED Federal Reserve System

GA Genetic Algorithm

GDP Gross Domestic Product

GS Goldman Sachs Inc.

IMF Internacional Monetary Fund

MC Market Capitalization

NAFTA North American Free Trade Agreement

PER Price Earning Ratio

ROC Receiver Operating Characteristic

S&P 500 Standard and Poor's 500

SMA Simple Moving Average

XGBoost eXtreme Gradient Boosting

Chapter 1

Introduction

1.1 Motivation

The stock market is considered one of the most complex systems in the world, which consists of many segments whose prices move up and down, without generating a clear pattern. The securities prices' movements are not merely random, nor are simple theories adequate to explain market operation [1]. Each business has an external environment constituted by elements of diverse types: Spatial and location factors, the markets, knowledge flows and networks, public policy, and the society and natural environment [2]. These elements of the external environment will influence the businesses and consequently will shape the movements in markets, which will feedback and influence things back in the real world, as well.

Financial theory, supported by advanced mathematic models, using selected key variables, try to read, interpret and, anticipate variations, although the economy as a result of human activity, can never be deterministic. The analyses have always been statistics, and the number of variables to collect and correlate to find patterns, tends to infinity.

So, what can be new here? In an era where there is the access to an enormous quantity of historical data, linked with the recent and continuous advances in computational algorithms based on Machine Learning, is where lays the base of this thesis.

Investors would not be the only ones to benefit from a reliable prediction of stock's changes, more specifically, in Bear Markets - which is the subject treated in this thesis -, considering that some variations can generate events that may lead to dramatic changes in the society, as a whole (e.g., people, companies, governments, etc.). In a way, it is fair to say that the prediction of stock's changes can be so important as weather forecast, not only to role every day man's decisions but also to mitigate catastrophes.

With this work, it is intended to add a humble contribute to demonstrate the potential of computational algorithms as a support tool for financial data analytic science.

1.2 Topic Overview

The subject treated in this work is the prediction of Bear Markets and other big downfall events in Stocks, through the use of Machine Learning algorithms.

Although a consensus has not still be reached by the academic literature, on what bear and bull markets are by definition [3], the media nowadays delineate a “classic” or “traditional” Bear Market as a 20% decline in stock prices [4] and is a critical element of determining stock market investments.

Machine Learning is the scientific research of statistical models and algorithms that computer systems use to enhance their performance on a specific task, progressively. Machine Learning algorithms create mathematical models of sample data in order to make predictions or decisions regarding the proposed situation.

Several Machine Learning algorithms will be used, and through time-series analysis, resorting to economic variables, it is intended the construction of models that will somehow be in accordance with the specifications proposed for this work - predict stock market big price decreases, with time advance.

1.3 Objectives

The main goal of this thesis is to detect Bear Markets, and other significant market price decreases, with six-twelve months in advance, in the Standard and Poor’s (S&P) 500 index. Therefore, the whole work can then be divided into several objectives:

- Calculate and mark the several downfall events dates, through the analysis of S&P 500 index, to establish a target for the algorithm’s model;
- Gather several United States of America’ (US) economic variables and determine which of them allow to predict stock market downfalls and which of them have more relevance in this task;
- Produce several models applying intelligent computing techniques of Machine Learning to discover the market downfall event with advance;
- Arrange innovative computational techniques to optimize the results of the produced models.

1.4 Contributions

This work is intended to add to the academic community contributions in stock price decreases’ predictions, in terms of:

- Utilization of Machine Learning models to anticipate big downfall events in the stock market;
- An ensemble approach joining different Machine Learning models;
- Creation and implementation of a Genetic Algorithm for hyperparameters’ tuning of Machine Learning algorithms.

1.5 Thesis Outline

This thesis is composed by five main chapters, each one divided by several sections. Followed, is the chapters' order with the respective brief explanation:

- Chapter 1 - Introduction
  Presents an overview of the work, providing its motivation, topic overview, objectives, and contributions.
- Chapter 2 - Background
  Sets the theoretical basis of the concepts used in this thesis, not only on the economic and financial level but also on the computational Machine Learning models used to retrieve results. Includes also the state of art regarding this subject.
- Chapter 3 - Proposed System Architecture
  Based on the previous chapter, it reveals the proposed system architecture and implementation of several methodologies and approaches in order to achieve and accomplish the work's objectives proposed.
- Chapter 4 - Results
  The chapter where will figure the results obtained in the different case studies prepared for this thesis. Will commence with a brief explanation of the metrics used as criteria evaluation, and the methodology used summarized. Include also a discussion section, where the results obtained are compared with others already spoken in the state of art.
- Chapter 5 - Conclusion
  The final section where will be summarized all the work done, the several results obtained and the conclusions drawn from these. It will also feature the advises for further developments based in this work in the Future Work section.

Chapter 2

Background

In this chapter, it will be made firstly a passage and a more deep explanation of the important subjects to know before analyzing the practical work of this thesis. Economy and Machine Learning are the main themes and are going to be split into several subjects: From the stock market and its analysis to the Machine Learning models used and other methods of implementation and optimization.

The State of art is the final section of this chapter and will get into the related works where different Machine Learning methods and techniques, and economic variables are being used, supporting the choices made for this work in these matters.

2.1 Stock Market

The stock market is a public, controlled, secure, and managed environment for securities trading (stocks, stock options, real estate funds, etc.). Transactions may occur through stock exchanges or over-the-counter markets.

A stock exchange is a giant global network that tends to organize the market volume, where large daily amounts of money are moved back and forth. In short, almost 70 trillion dollars were traded in 2018 [5] and is more than the value of all goods and services in the whole world (Global Gross Domestic Product) economy [6].

Unlike a ”normal” market where its goods can be touched and taken home, on the stock exchange, only virtual goods are available. What is traded on this marketplace is predominantly securities. Securities are rights to assets, mostly in the form of shares that stand for a share in a company. The value of the company and the supply-demand rule will dictate the value inherent to its share. If a company hypothetically, starts increasing its profits gradually with a new business model, the value of its shares tends to increase, and from that, the investors can benefit and take profit from.

Stock exchanges from all countries in the world have their indexes, and all of these markets together create a global network marketplace. Dow Jones Industrial Average and S&P 500 are two of the most well known US indexes. Since the stock market brings together hundreds of thousands of market participants, who wish to buy and sell shares, it ensures (ideally) fair pricing practices and transparency in transactions.

2.1.1 Stock Market's Variations

The stock market movement is defined as non-stationary because the distribution of financial time-series is changing over time and is deterministically chaotic which means that financial time-series are short-term random but long-term deterministic. Many factors and unexpected events or incidents may cause the change of a financial time-series such as the stock market index and exchange rates, difficulting the prediction of financial market's movements [7].

However several works has been made throughout time trying to predict these movements and firstly as stated in the work by Chen [3], in order to predict stock market recessions, it is necessary to characterize its fluctuations. There are the increase and decrease trends, nominated as bull and bear trends, respectively. Despite all the works, a consensus has not still be reached by the academic literature, on what bear and bull markets are by definition [3]. However, the media nowadays delineate a "classic" or "traditional" Bear Market as a 20% decline in stock prices [4]. The same happens in the bull market, with the change of stock prices in the opposite direction (increase of 20% in stock price). A price decrease of 10% in generally nominated as a market correction. It is usual to consider all the business variations as a cycle, because it alternates between these two phases. It is also likely to add in consideration the points in between, where the market is "sideways" (figure 2.1).

Figure 2.1: Example of a financial time-series with a bull (green), sideways (blue) and bear (red) trends represented.

Several works tried to find patterns in the different stock market phases - Coakley and Fuertes [8], Perez-Quiros and Timmermann [9]. Also in this area, Edwards et al. [10], resorted to the use of non-parametric approaches to detect the different phases of a stock market in several countries. In works such the one from de Almeida et al. [11], there is the use of Machine Learning models in order to classify the different types of markets (Bull, bear or sideways).

2.1.2 Analyzing and Predicting in Stock Market

Loungani et al. [12] stated that, opinions diverge on, if stock market movements can predict business cycles. Fischer and Merton [13], in works regarding the link between movements in the S&P 500 and the economy alleged that *"stock price changes are the best single variable predictor of the business cycle."* and Barro [14] reports his positive opinion regarding this subject, extolling the power of the stock market's predictions, despite how difficult it is to make accurate macroeconomic forecasts. However, there were economists with opposing opinions, as well. For instance, according to Moore and Cullity [15], Paul Samuelson (1994) stated deprecatingly that: "The stock market is a terrific forecaster. It has predicted nine of the last five recessions.".

However, this work is focused not on the power of stock market predicting the US business cycles but in the predictions of the stock market itself (more specifically, the S&P 500 index), exploring the time-series analysis and prediction power of different economic variables.

Two different views about the predictability of the stock market are extolled in several works. Stock market prediction is not an easy task, due to the fact that is an environment essentially dynamic, nonlinear, complicated, nonparametric, and chaotic in nature [16]. The idea in which this statement is based on, led to the hypothesis where it is supposed that the prediction of future stock prices based on its past prices is not possible - The Random-Walk Hypothesis. On the other hand, there is the community believing that stock market's predictions based on past data is achievable and profit can be obtained through the exploitation of several technical and fundamental analysis, as well as momentum strategies (buy when the market is bullish, sell when the market is bearish) [17].

In his work, Chen described two of the reasons why the prediction of stock market is useful and appealing: [3]:

- Market participants may benefit from such predictions because the predictability would help them to form market-timing strategies. This question was already justified in several works, demonstrating that an investor could have more positive results following a market-timing strategy rather than a buy-and-hold strategy;
- Predicting recessions in the stock market would help policymakers.

Nowadays most of the community in the financial area, believes that is possible to make forecasts regarding stock market's movements. However, under the circumstances of nowadays society's evolution and with all the advances in Machine Learning and computational algorithms, the question became, "for how long a model based on the stock analysis, can forecast future movements of itself?". A model will never be completely effective for all sorts of situations, so innovation in these matters will always be plausible.

2.1.2.1 Fundamental Analysis

Fundamental analysis is the analysis of the financial, economic, and market situation of a company, a sector or economic data, a commodity or a currency, and its expectations and projections for the future.

From financial statements, it is retrieved the information about firms' fundamental values. Stock price at times deviate from these fundamental values, but it has a higher probability of ending up moving towards them. Therefore, the analysis of published financial statements can discover values that are not reflected in stock prices. The fundamental analysis chooses the "intrinsic values" discovered from financial statements as a reference to compare with stock prices, and this way identify overpriced and underpriced stocks [18].

The fundamental analysis is based on several factors which can be summarized in three main strands: Economic Analysis, Industry Analysis, and Company Analysis [19].

The macroeconomic analysis makes assumptions about a specific market based on macroeconomic indicators. These variables describe different segments of the economy, and different examples of it are stated in the topics below [19].

- Labor Matters:
  - Unemployment rate;
  - Average Hourly Earnings;
  - Employment Cost Index;
- Consumer Spending and Confidence:
  - Personal Income and Spending;
  - E-Commerce Retail Sales;
  - Consumer Confidence Index.
- National Output and Inventories:
  - Gross Domestic Product (GDP);
  - Index of Leading Economic Indicators;
- Money availability, Prices and Productivity:
  - Consumer Price Index;
  - Producer Price Index;
  - Yield Curves.

In Company analysis, many useful indicators are calculated in order to reach some conclusions about the company's profitability, price, liquidity, leverage, and efficiency [19]. To achieve a substantial study about a corporation, we should take into account more subjective parameters, i.e., business plan and management personnel. Some examples of these indicators are the Earnings Per Share (EPS), Price Earning Ratio (PER), and Market Capitalization (MC).

Table 2.1: Example of company analysis variables (Fundamental Analysis).

Indicator	Measure	Description
Earnings Per Share (EPS)	$EPS = \frac{Net\ income\ -\ Dividends\ on\ preferred\ stock}{Average\ outstanding\ shares}$	Profit allocated to each share, determining a share's price calculation of other indicators
Price Earning Ratio (PER)	$PER = \frac{Market\ Value\ Per\ Share}{EPS}$	Valuation of share price compared to EPS, time needed to recover investment, earnings growth in the future. Low PER: Buy (High PER can suggests higher earnings expectations)
Market Capitalization (MC)	$EPS = Company's\ Share\ *\ Market\ Price$	Company's dimension determined by the Total market

If an economic forecast shows that the economy will grow, some industrial groups will most likely benefit more when compared to others. That is why the industry analysis is also essential from the investor's point of view. To evaluate the potential of an industry sector, it is essential to consider different factors, like growth rate, regulation, market size, consumers, or the importance of that sector to the overall economy [19].

2.1.2.2 Technical Analysis

Technical Analysis is a tool that resorts to the use of technical indicators or technical indexes to analyze the future prices of any financial asset and different trends of the market. The analysis of returns and pullbacks of a given stock in order to produce indicators is an example of technical analysis. Technical indicators are mathematical formulas that are applied to the price or volume data of security for modeling some aspects of the movement of those prices [20]. One of the best known technical indicators is the simple moving average (SMA).

The SMA is the average of a set of past values of a given stock. The more used for this proposed are the 20, 40, and 60 moving average. The SMA smooth the prices in order to allow a better understanding on which way and how much the prices are moving. The difference from the simple moving average to the exponential moving average is that the last one adds significance to the most recent data.

$$SMA = \frac{1}{t}\sum_{i=t-n}^{t} x(i) \tag{2.1}$$

The simple moving average also makes part of another technical analysis tool called Bollinger Bands (BB). This tool is composed of three different bands: SMA (middle band), $SMA + Standard\ deviation$ (upper band), and $SMA - Standard\ deviation$ (lower band). The inclusion of the standard deviation will be useful in analyzing the volatility of an asset since the values will be higher/lower than the price values itself when there is a bigger/smaller volatility of the asset analyzed.

There are some disadvantages to the use of technical indicators. In his work Gorgulho et al. [21] enumerate some, stating that it is appropriate to gather several indicators due to the fact that one individual indicator can give false alarms and also, addresses the need to choose a time interval, which is a task that can be complicated.

2.2 Time-Series Analysis

A time-series is a set of historical measurements y_t, having into account the chronological sequence of an observable variable y at equal time intervals. Time-series studies, can be made for several purposes, such as the future's forecasting based on past knowledge, the understanding of the phenomenon underlying the measures, or merely a succinct description of the salient features of the series [22].

In time-series analysis, there is a concern with decomposing the variation in the series in four different components [23, 24]:

- Cyclical variations - Medium-term changes in time-series which end up to repeat in time. Most of economic/financial time-series demonstrate a sort of cyclical variation that can be described by four different phases: Prosperity, decline, depression and recovery (figure 2.2)

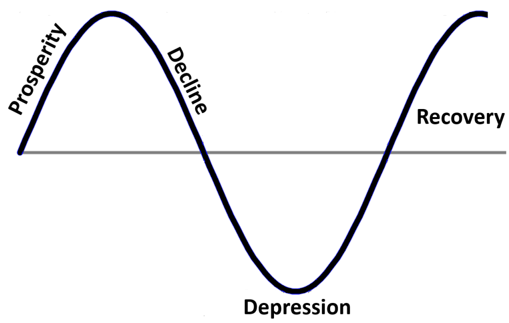


Figure 2.2: Different phases of a business cycle.

- Seasonality - Variations within a certain time span during a season (figure 2.3);
- Trend - Long term movement in a time-series or the long-term change in the mean level (figure 2.3);
- Other irregular variations - After removing trend and cyclic variations, it is still left the residuals, which can be or not ”random”.

A time-series $\{x(t), t = 0, 1, 2, ...\}$ generally follows a model probability that describes the joint probability of a random variable x_t, and the mathematical expression which characterizes this probability is named a stochastic process. A stationary process is a type of stochastic process in which the statistical properties, such as mean and variance, do not depend upon time [23]. Most of the probability theory of time-series is concerned with stationary time-series and a specific feature of Machine Learning methods, is that most of these can work with only stationary data. For this reason, most of the time-series analysis often requires to turn a nonstationary time-series into a stationary one [24, 26].

Time-series forecasting methods can be broadly divided into three different groups: subjective, univariate, and multivariate. In this work, it is supposed to obtain a multivariate procedure with Machine Learning models, trying to predict the S&P 500 index price drops, through the use of several economic variables. However, firstly, it will be needed to investigate whether each variable used as a feature, is stationary or not (univariate procedures). If we include a nonstationary time-series, it might mean that

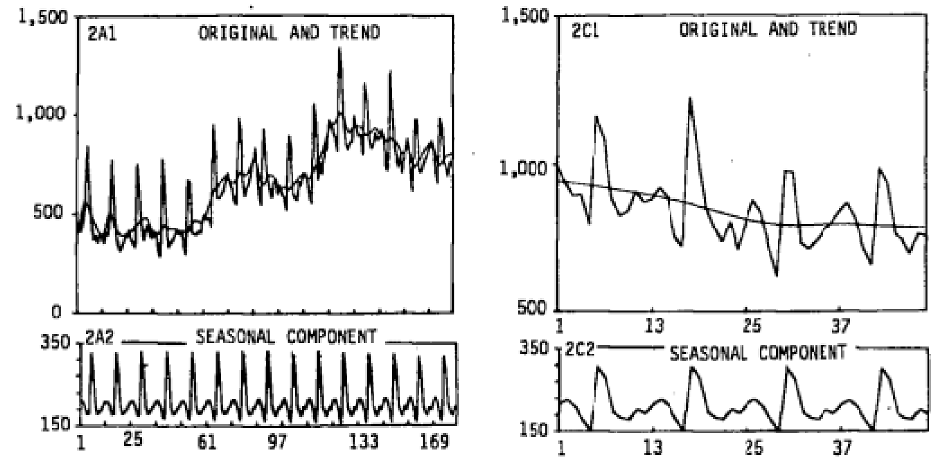


Figure 2.3: Example of time-series with the corresponding original signal, trend and seasonal component [25].

estimators are not consistent, and the standard statistical tests are not valid. Therefore, the models might infer a causal relationship where could not exist, between a set of variables [27]. Some mathematical tests such as Dickey and Fuller are generally used to detect stationarity in time-series data.

There are several methods used in order to turn a nonstationary univariate time-series into a stationary one. A nonstationary series which is integrated of order d can be made stationary by taking its differences d times. This method will remove an existent trend [27]. Also, to remove excess in terms of variance of a time-series, it can be applied a moving average in order to smooth the variable throughout time.

2.3 Machine Learning Models

In this section, follows a brief explanation of the algorithm models used in this work, as well as of the concepts which led to the use of these algorithms. To transform an intricacy related to the prediction of Bear Markets it will be made a revision on the Machine Learning approach taken.

2.3.1 Machine Learning approach

Prediction and prognosis problems, when interpreted from a Machine Learning point of view, are mainly classified into two different categories: Unsupervised and supervised Learning (figure 2.4).

In unsupervised learning, the algorithm attempts to group a set of different objects without any prior knowledge of these classes or any labeled output [29]. This method allows to have a more significant understanding of the dataset analyzed and also to estimate the number of classes.

In supervised learning, having as a benchmark a set of examples and their class labels (training set), the process consists in associating the different unlabelled objects to a specific output. Supervised learning can be subdivided into Classification problems, where, based on the training set, the algorithm will predict the label class of the new object, or in Regression problems where the output will be a real number instead of a label classes [29].

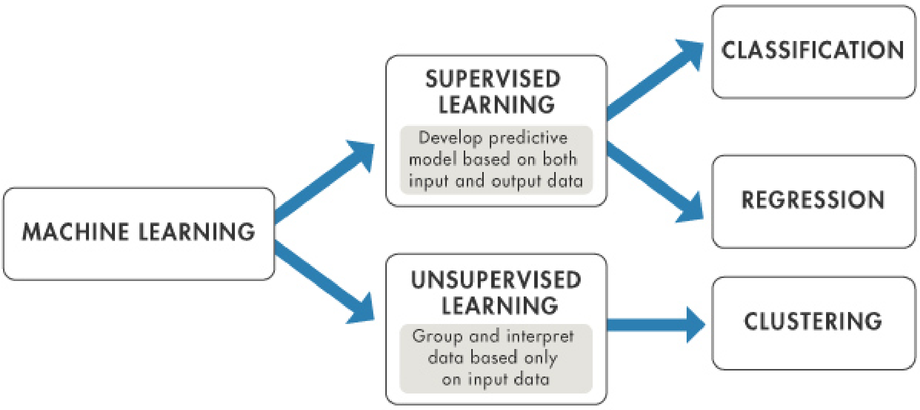


Figure 2.4: Types of Machine Learning problems and approaches [28].

The problem discussed in this work is of binary classification nature since ”Is it a Bear Market?” is a question with only two outcomes, and having two different possible outputs (class labels) fits this way into the Classification problem type (Supervised Learning).

To deal with this kind of problems, there are specific linear and non-linear algorithms more suitable for the production of models that will try to predict a final output of binary nature. Below in the present work, there is a brief explanation of three examples of these types of algorithms: Logistic Regression and the decision trees algorithms based, Random Forest and eXtreme Gradient Boosting (XGBoost).

2.3.2 Logistic Regression

Regression analysis is used to determine the magnitude of relationships between variables and to model relationships between them and for predictions based on the models.

The outcome of logistic regression is a function that describes how the probability of an event of binary classification nature, varies with the predictors (features) and respective parameters [30] (expression 2.2). From the dataset, the method used to estimate these parameters is called maximum likelihood. The Likelihood function is a function of the unknown parameters in one's model and, thus, can alternatively be denoted as $L(\Theta) = \beta_0; \beta_1; \beta_2; ...; \beta_p$. The standard approach for maximizing an expression like the likelihood function for the binomial example here is to use calculus by setting the derivative dL/dp equal to zero and solving for the unknown parameter or parameters [31].

Logistic regression will obtain a final result to solve the problem, applying the logit transformation to the dependent variable. In essence, the logistic model predicts the logit of Y from X. The logit function is the natural logarithm of odds of Y, and odds are ratios of probabilities (p_i) of Y happening [32]. This logistic regression model is expressed in 2.2.

$$Logit(Y) = \ln \frac{p_i}{1 - p_i} = X \tag{2.2}$$

The input X will correspond to the following expression 2.3, of the parameters and respective coefficients.

$$X = z_i = \beta_0 + \beta_1 * x_{i1} + \beta_2 * x_{i2} + \ldots + \beta_j * x_{ij} \tag{2.3}$$

Where z_i value corresponds to the odds ratio, x_{ij} is the j^{th} predictor for the i^{th} case, β is the j^{th} coefficient, and *p* is the number of predictors.

From the expression 2.2, it is possible to derive 2.4 that describes the relationship between z_i and the probability p_i of the event to occur

$$p_i = Probability(Y = outcome\,of\,interest \,|\, X = z_i) = \frac{e^{z_i}}{1+e^{z_i}} = \frac{1}{1+e^{-z_i}} \tag{2.4}$$

This relation has the following graphic representation presented in figure 2.5. This graphic representation demonstrates that the expression 2.4 will give outcome values that can only lay between zero and one. These result values, from the logistic regression model, will represent the probability of an event to occur.

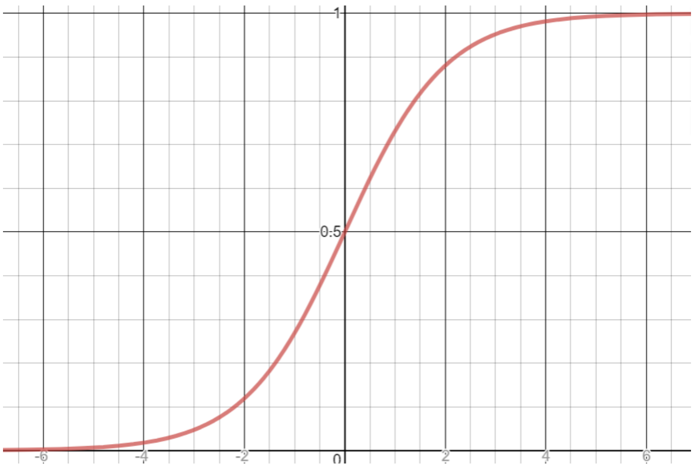


Figure 2.5: Plot of the function 2.4, which corresponds to Logistic Regression possible output values.

Since the probability of an event must lie between zero and one, it is unrealistic to model probabilities with linear regression techniques, because the linear regression model allows values to be greater than one or smaller than zero. The logistic regression model is a type of generalized linear model that extends the linear regression model by comprising the range of real numbers to the 0-1 range. This way, logistic regression is preferred in cases where the response variable can take only binary values [33].

2.3.3 Decision Trees

Non-linear techniques could be used to obtain a more accurate classification if the relationship between variables is not linear in parameters [33].

Decision trees (figure 2.6) are a popular tool used for classification and prediction purposes. Decision trees generally consist of the following three steps:

- Structuring the problem as a tree by forming end nodes of the branches, which are connected for a specific path or scenario along the tree;

- Assigning subject probabilities to each represented event on the tree and the respective payoffs for consequences;

- Identifying and selecting the appropriate course(s) of action based on analyses.

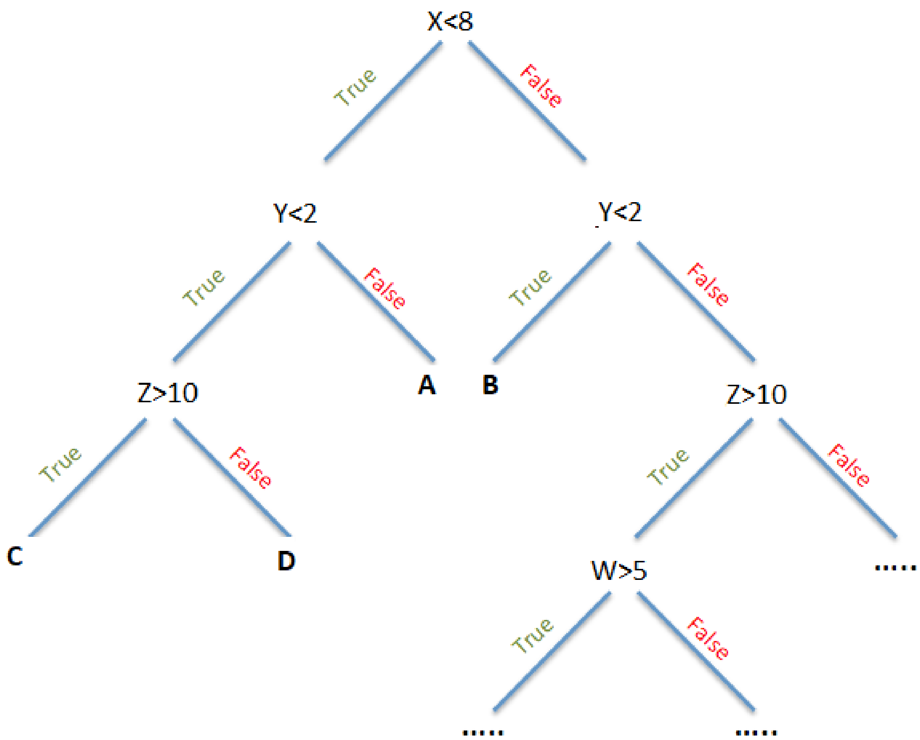


Figure 2.6: Example of a Decision Tree.

Recently there has been much interest in “ensemble learning” — methods that generate many classifiers and aggregate their results. Two well-known methods are boosting and bagging of classification trees [34].

The XGBoost and Random Forest algorithms are applications of boosting and bagging trees, respectively. Below there is a brief explanation of each algorithm.

2.3.3.1 Random Forest (Bagging Trees)

In bagging, there will not exist dependency between successive and previous trees since each tree is independently constructed using a bootstrap sample of the data set. In the end, a simple majority vote is taken for prediction [34]. The idea of bagging is based on creating several subsets of data from the training sample chosen randomly with replacement. Now, each collection of subset data is used to train their decision trees. As a result, we end up with an ensemble of different models. The average of all predictions from the different trees is used, which turns out to be more robust than a single decision tree.

The random forest algorithm (figure 2.7) is an application of bagging trees, adding an extra step: In addition to taking the random subset of data, it also takes the random selection of features rather than using all features to grow trees.

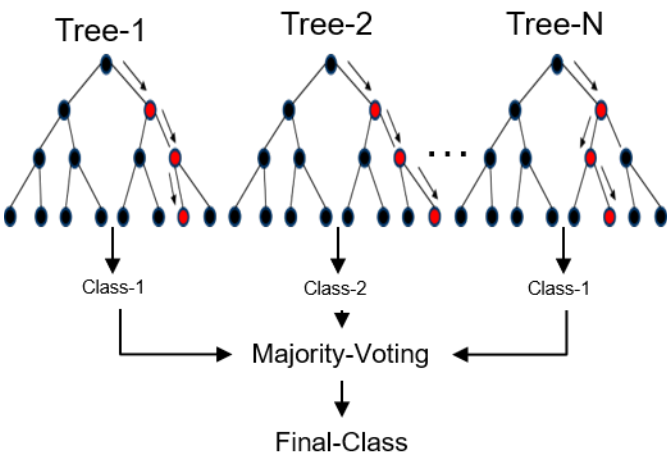


Figure 2.7: Example of Random Forest with N decision trees.

Noise in the data may cause the tree to grow in a completely unexpected manner, and trees that are grown deep to learn highly irregular patterns tend to overfit the training sets. Random Forests overcome this problem by training multiple decision trees on different subspaces of the feature space at the cost of slightly increased bias. This method allows that none of the trees in the forest sees the entire training data [35].

2.3.3.2 XGBoost (Boosting Trees)

The XGBoost is an algorithm widely used by the data science community and one of the most well recognized as well. It is a classification method based on decision trees concept, but with a learning component.

A boosting algorithm is a method that takes the predictors considered weak learners and converts it into strong learners [36]. The idea stands on training weak learners sequentially, each trying to correct its predecessor.

In boosting, successive trees give extra weight to points incorrectly predicted by earlier predictors. In the end, a weighted vote is taken for prediction [34]. Gradient Boosting algorithm is one of the variations of boosting, which represents the learning problem as gradient descent in order to minimize an arbitrary differentiable loss function L that measures the performance of the model on the training set [37].

The boosting algorithm will run for **M** boosting iterations, consisting in the following steps [37]:

- A model F_m is defined to predict the labeled variable y ($F(x) = \hat{y}$);
- This model will be associated with a residual $f_m(x) = y - F_m$ which is fit to the residuals from the previous step;
- Now, F_m and $f_m(x)$ are combined to give F_{m+1} ($F_{m+1} = F_m + f_m(x)$), the boosted version of F_m. The loss function from F_{m+1} will have a lower value than that from F_m;
- To improve the performance of F_{m+1}, we could model after the residuals of F_{m+1} and create a new model $F_{m+2} = F_{m+1} + f_{m+1}(x)$.

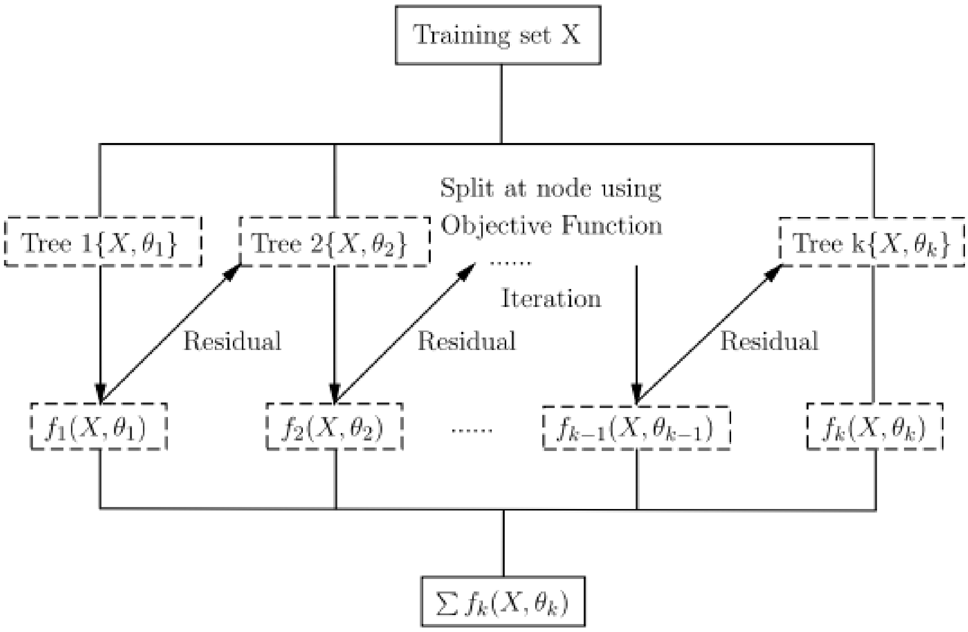


Figure 2.8: Flow chart of extreme gradient boosting (XGBoost) [38].

This can be executed for several iterations until residuals have been minimized as much as possible (Figure 2.8).

Gradient boosting is the base model of XGBoost. As an improvement, the XGBoost will add regularization to the loss function to establish the objective function measuring the model performance [37–39], which is given by the expression 2.5.

$$Obj(\Theta) = L(\Theta) + \Omega(\Theta) \tag{2.5}$$

This function consists of two parts: $L(\Theta)$ and $\Omega(\Theta)$, Θ refers to the various parameters in the formula. $L(\Theta) = l(y_i, \hat{y}_i)$ calculates the difference between the prediction and the true label for a training data set and can be any convex and differential function. Square loss function and Logistic loss are the most commonly used. The $\Omega(\Theta)$ describes the complexity of the estimator (tree) and is defined as:

$$\Omega(\Theta) = \gamma T + \frac{1}{2}\lambda w^2 \tag{2.6}$$

Where T is the number of leaves of the tree, and w is the leaf weights (i.e., the predicted values stored at the leaf nodes). The use of this function will make the process tend to create simpler trees, penalizing each tree leaf and extreme weight with the parameters γT and λw^2, respectively, and these coefficients can be set before the model training process (hyperparameters). This regularization function helps controlling model overfit. Cross-validation as overfitting control and algorithm parameter setting will be the next topics to be addressed in this thesis.

2.3.4 Hyperparameters and Genetic Algorithms

Gradient boosting, random forest, and neural networks are examples of Machine Learning algorithms that, for regression and classification processes, will need to set several hyperparameters before running them. These algorithms already own first-level model parameters, which are defined during the training process. Besides these, it is also possible to define the hyperparameters in such a way that it maximizes the results [40].

For an appropriate selection of hyperparameters, Machine Learning users can resort to the default values of hyperparameters that are specified in implementing software packages, or manually configure them based on recommendations from examples researched. Also, empirically with or without the help of other optimization algorithms, it is possible to obtain the best hyperparameters combinations [40].

Within these optimization algorithms, there are several options to use, such as the grid and random search, which do a simpler search, and others more complex like Genetic Algorithms (GA) and Bayesian optimization and in the State of Art chapter will be explained this choice.

GAs are adaptive methods which may be used to solve search and optimization problems. They are based on the genetic processes of biological organisms and premised on the principles of natural selection and "fittest survival", the GAs are able to evolve solutions to real world problems.

In nature the living beings always fought for the several resources and to counter nature adversities in order to survive. The most successful species to fit under these conditions were the ones that would produce more offsprings and whose "genes" would more likely be preserved for further generations. Contrariwise, the species that struggle more or that are least successful to fit, are unlikely to produce more offsprings and guarantee succession [41].

This analogy is plausible since the GAs will work over a population of individuals (or "chromosomes"), each one constituted by parameters (or "genes") (figure 2.9).

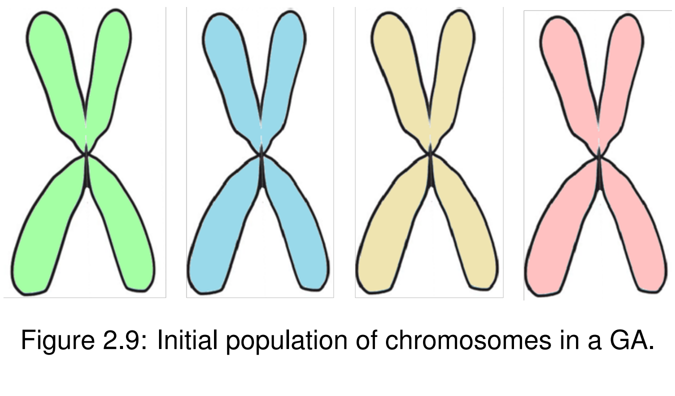


Figure 2.9: Initial population of chromosomes in a GA.

In a generation it will be assigned a value of "fitness" (in this work case, it can be some metric used to evaluate the Machine Learning model) to each individual and the ones with greater values will survive for the next generation while the others will be excluded (figure 2.10).

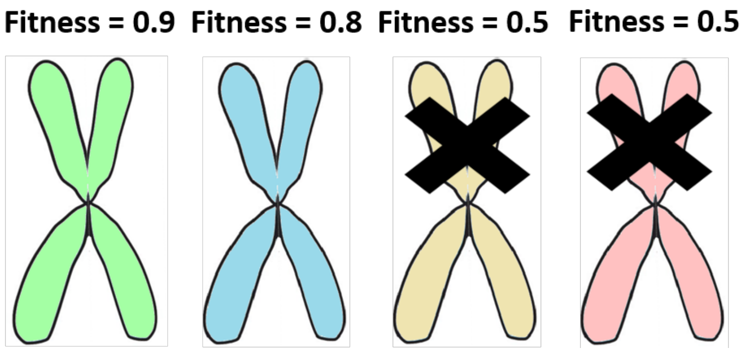


Figure 2.10: Selection phase between parents chromosomes, in a GA.

Among the selected individuals a crossover process will be carried out so that the new generation (children chromosomes) turns out to be a mixture of the surviving individuals (parents chromosomes). (figure 2.11)

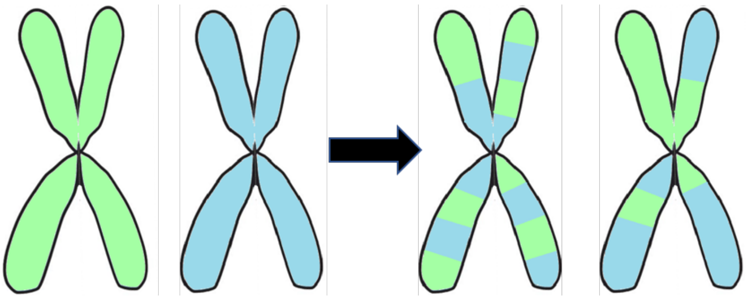


Figure 2.11: Crossover phase to generate children chromosomes, in a GA.

It will still be executed a mutation in each one of the children chromosomes, in order to make a vast research in the possible values for each parameter and not "focus" only on the initially generated values (figure 2.12).

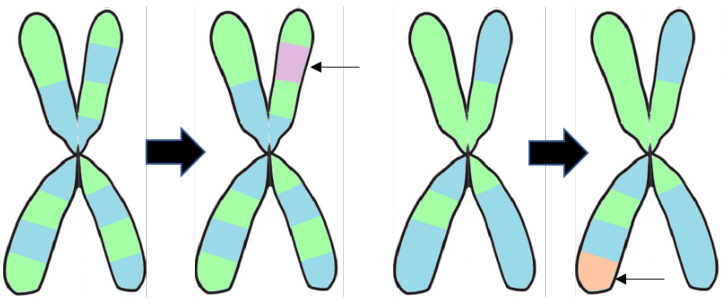


Figure 2.12: Mutation phase of the children chromosomes, in a GA.

The process of Train-Selection-Crossover-Mutation will be executed until it found (ideally) the converged value of "fitness" (figure 2.13).

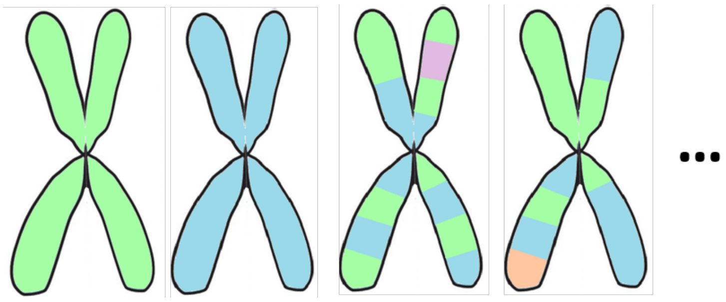


Figure 2.13: New generation ready to repeat the GA process.

In the approached cases of this thesis, the GA will have as optimization goal to find the combination of algorithm hyperparameters - representing the genes of each chromosome - that maximize the fitness function results. The function choice will be based on the best metric to evaluate analytically the problem in question in each case study.

2.3.5 Cross-Validation

Before proceeding to the classification process, it is necessary to divide the entire data set into training and test subsets.

One of the drawbacks of the nonlinear approaches, usually when analyzing a small data set, is the model overfit that occurs when a model learns the detail and noise in the training data to the point that negatively impacts the model's performance on new data. Cross-Validation is one of the techniques used to overcome or to estimate the effect of overfitting.

The "Hold-out" method will assign data points to two sets D0 and D1, normally called the training set and the test set, respectively (figure 2.14). Subsets' size choice is arbitrary, although typically, the test set is smaller than the training set. Then the model is trained on D0 and tested on D1. This is a simple manner of splitting the data set. However, it does not cross-validate, and therefore, the overfitting can not be detected.

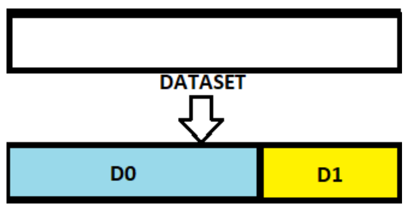


Figure 2.14: Hold-out data partition method.

The basic principle behind a cross-validation system is to break the data into several subsamples for further training and testing iterations in order to estimate the model's overfitting and predictive performance [42]. K-fold is an example of a common-used cross-validation method. In K-fold, the training subset is split into several other subsets and validated by different testing subsets, as well. This process occurs through several iterations, having this way a more reliable source about the model's predictive performance (figure 2.15).

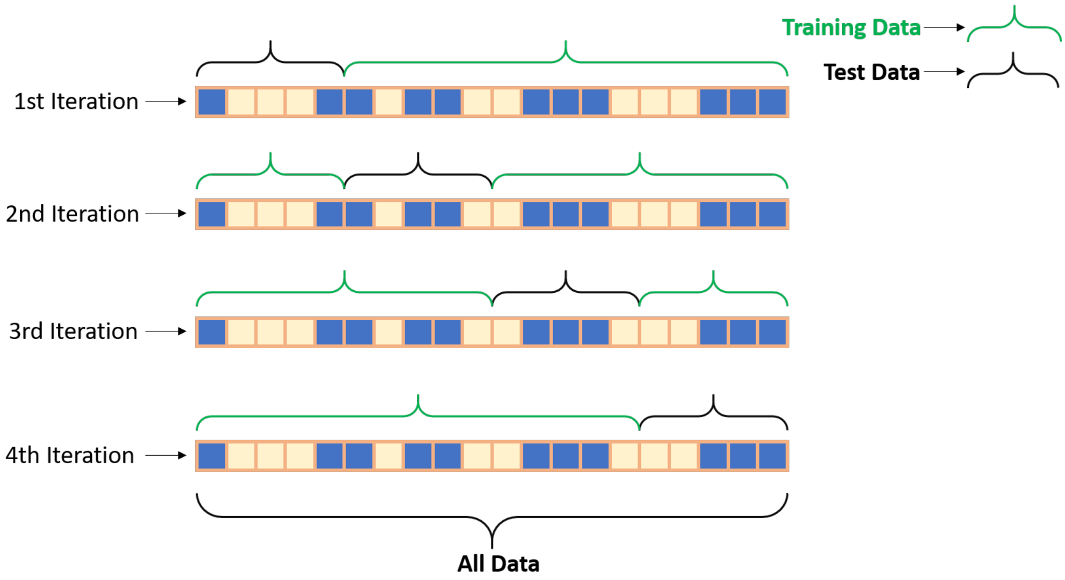


Figure 2.15: K-fold Cross-Validation method with K=4.

2.4 State of Art

Recently there have been significant developments in computational algorithms' study, which led to the attempts and the accomplishment of several works in diverse areas in order to take advantage of this evolution. Finance is one of the areas where much use of computational algorithms began being used in order to predict, for example, stock prices and respective variations. Most stock traders nowadays depend on Intelligent Trading Systems, which help them in predicting prices and making instantaneous decisions based on several situations and conditions [43].

After explaining more deeply in sections 2.1 and 2.3, the theoretical bases of this work, next section, will show examples of works, in which different models and data were used and how they can be appropriate for this work. Thus, it is possible to obtain a more reasoned perspective of the choices made regarding the approach adopted for this work.

2.4.1 Stocks and Indicators

After researches made to find which financial stock would be more appropriate to apply this work, the choice fell on the S&P 500 index. The scenario of analyzing all global economies is unrealistic due mainly to several structural differences between them and the delivery deadlines of this work. Therefore, the choice of the US economy seemed appropriate, having into account that it is considered one of the strongest at the global level, like is possible to verify in the 2018's Gross Domestic Product (GDP) Rank World Development Indicators from The World Bank Group [44]. Consequently, it can cause significant effects on all other economies worldwide. S&P 500 was the index chosen because it is the broadest measure of the US economy among the major indexes, due to the number of important entities it comprises.

After some research, the indicators used as Bear Market signs by the Bank of America Merryl Lynch

Table 2.2: Bear Market post signs by the Bank of America Merryl Lynch [46].

Indicator	Category	Current	Trigger	Triggered	Data since	Hit rate %**
Fed raising rates	Credit	100bp	>75bp	☑	1982	100%
Tightening credit conditions	Credit	-8.5%	>0%	☐	1990	100%
Trailing S&P 500 12m returns	Returns	19.7%	>11%	☑	1936	92%
Trailing S&P 500 24m returns	Returns	34.8%	>30%	☑	1936	92%
Low quality outperforms high quality (last 6m)	Returns	-0.5ppt	>0ppt	☐	1986	100%
Momentum outperforming (6m/12m)	Returns	+2ppt/+3ppt	>0ppt	☑	1986	100%
Growth outperforming (6m/12m)	Returns	+2ppt/+2ppt	>0ppt	☑	1986	100%
5% pullback over prior 12m	Returns	0	>0	☐	1928	92%
Low PE underperforms (6m/12m)	Returns	+2ppt/-4ppt	<0ppt	☐	1986	100%
Conf Board consumer confidence	Sentiment	130	>100	☑	1967	100%
Conf Board net % expecting stocks higher	Sentiment	27	>20	☑	1987	100%
Lack of reward to beats	Sentiment	+1.2ppt	<1ppt	☑	2000	100%
Sell side indicator	Sentiment	56.1%	>63%	☐	1987	100%
FMS cash levels	Sentiment	4.4%	<3.5%	☐	2001	100%
Inverted yield curve	Sentiment	57bp	<0bp	☐	1962	88%
Chg in long-term growth expectations	Sentiment	+1.9ppt	+0.6ppt	☑	1986	100%
Rule of 20	Valuation	22	>20	☑	1960	100%
VIX rises over prior 3m	Sentiment	13	>20	☐	1986*	100%
ERR rule	Growth	Yes***	See footnote	☑	1986	100%

*Based on VXO 1986-1989
**% of bear markets where signal was trigged
***ERR rule: Within six months' window 1) the 1m ERR is below 1.0 for two or more months, 2) the 1m ERR drops from 1.0+ to below 1.0 and, 3) the 3m ERR is under 1.10
Source: BofAML US Equity & Quant Strategy, BofAML Global Investment Strategy, FRB, S&P, Conference Board, BLS

(BOFAML) were found (that can be found in table 2.2). According to an article in [45], the BOFAML made the following declarations about this list of indicators:

"we compiled a list of Bear Market signposts that generally have occurred ahead of Bear Markets. No single indicator is perfect, and in this cycle, several will undoubtedly lag or not occur at all. However, while single indicators may not be useful for market timing, they can be viewed as conservative preconditions for a Bear Market."

The veracity of these economic variables can be reflected by the figure 2.16, where it is possible to see that almost all of the signals were triggered before previous Bear Markets.

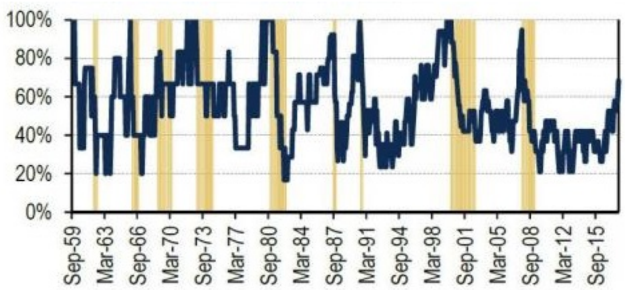
The article [45], was written by the end of 2017 and as it can be seen in table 2.2, by that time already 11 of 19 Bear Markets signals had been triggered, and S&P 500 price was one of the highest of all time,

a)

Start of bear market	Available signals	% triggered at the market peak or during a bear market	% triggered at tl market peak
Nov-68	5	100%	80%
Jan-73	6	100%	83%
Nov-80	6	100%	100%
Aug-87	13	92%	92%
Jul-90	17	100%	94%
Mar-00	17	100%	100%
Oct-07	19	100%	95%
Current	**19**	**68%**	**68%**

Source: BofAML US Equity & Quant Strategy

b)



Source: BofAML US Equity & Quant Strategy

Figure 2.16: a) Table with BOFAML's Indicators Triggered in previous Bear Markets and in 2018. b) time-series with percentage of Bear Market signals triggered with Bear Markets shaded [45].

Figure 2.17: S&P 500 with price marked in end of 2017 [47].

i.e., still far from being considered in downtrend (figure 2.17) [47].

Several articles came out throughout 2018, highlighting BOFAML's warnings about the increase of triggered indicators [45, 48]. In June of 2018, BOFAML confirmed that 14 of the 19 indicators had been triggered [48]. A few times later and after reaching an index price over 2900 in October of 2018, the S&P 500 went on a downtrend, and in December of 2018, registered values below 2400, a price decrease of almost 20%, thus, close to being considered a Bear Market. These facts valorize BOFAML's signals by verifying an existent correlation between these signals being triggered and the stock market's declines.

Other entities such as Goldman Sachs Group Inc. (GS), also designed indicators for Bear Market predictions. From a survey made, it was possible to find the different variables that are combined to form this GS's indicator (figure 2.3) [49].

Table 2.3: Bear Market's signals by Goldman Sachs Group Inc [49].

	Level	Percentile
Unemployment	3.7	94%
Shiller PE	30.9	93%
ISM	57.7	76%
Term Structure of Yield Curve	0.9	66%
Core Inflation	2.2	36%
GS Bear Market Risk Indicator		**73%**

** 100 percentile means these variables are at their highest level except for SYC and Unemployment where 100% means they are at their lowest.*

Source: Haver, Bloomberg, Goldman Sachs Global Investment Research

The respective indicator in mid-2018 reached a Bear Market risk of 75%, which is not only nearly 10% higher than a year ago, but well above where it was just before the last two market crashes, being in the "red zone" and with one of the highest values ever (figure 2.18). Thus, the situation by that time can be considered urgent based on this indicator [49].

Figure 2.18: Goldman Sachs' Indicator value throughout time [49].

Despite existing evidence of an inverse correlation between the market's returns and GS's indicator, as shown in figure 2.18, there are still some doubts on how it is supposed to interpret the GS Bear Market signal. In the article [49] is stated that *"Goldman's answer is two-fold, laying out two possible outcomes from here, either a sharp, "cathartic" Bear Market, or just a period of slower, grinding low returns, without a clear trend in the market for the foreseeable future"*.

Beyond the economic variables found of relevant financial entities, works were found from particulars, in which are used economic variables for prediction of Bear Markets or recessions. In his work, Chen [3], searched for the possibility of different macroeconomic variables correlating with Bear Market time spans and conclude that it is possible to obtain better results using certain variables, such as the term spreads (3years-10years) and inflation rates.

Figure 2.19 shows the variations of the term spread (3years-10years) in time, also known as the yield curve, and the recession's periods represented in shades throughout the graphic. Analyzing the graphic representation in 2.19, it is possible to denote evidence of a correlation between the market direction and this variable. Before each period of recession, it is observable the passage of the indicator to negative values for a certain period.

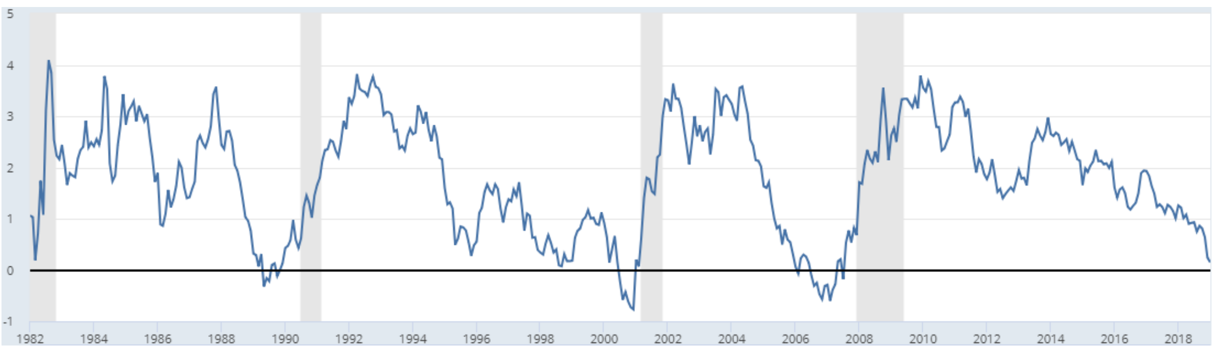


Figure 2.19: Term Spread (3years-10years) time-series data, with recession periods in shade [50].

In [42], there are also several macroeconomics indicators used to predict recessions in the economy. Most of the economy recessions were detected with probabilities above 50% with the models used.

Some of the indicators used in this thesis, such as the "VIX" and "Yield Curve", were already mentioned as BOFAML's Bear Market signs.

There are also some contents allocated on the website of Yardeni [51], which are more related to the prediction of Bear Markets. There, it is possible to find some economic variables, such as the Commodity Research Bureau (CRB) raw industrials spot price index and the misery index, that present a significant correlation with market downfall events.

In conclusion, indicators used by relevant finance entities (BOFAML, GS) demonstrated to correlate with Bear Markets as most of them were triggered at such times. Furthermore, other economic variables used as indicators in Chen [3] and Barbosa [42] also helped to produce good results to predict Bear Markets and US economic recessions, respectively. Therefore, the incorporation of these indicators in this thesis' practical work may be a promising approach.

One of the goals in this work will be to gather as many variables of these referred, as possible, and not only to verify the correlation between them and market trends but also, try to get a conclusion on which of these variables are more important/reliable for predictions.

2.4.2 Models and Results

In section 2.3, there was a brief theoretical explanation of different Machine Learning algorithms used to produce predicting models - logistic regression, random forest, and XGBoost. This section will reference the several works made in the financial area, using those and other models and whether or not achieved significant positive results.

The inherent non-linearity in data discourages the use of linear classifiers. However, in work by Li et al. [52], "*The best model turned out to be linear classifier: logistic regression. It gave 56.65% successful rate and 2000% cumulative return over 14 years*". Also, in other works in this area, logistic regression is used as one of the models for forecasting market trends - [42, 53, 54].

In his work, Creamer and Freund [53] used logistic regression, random forest regression techniques for predicting performance, and quantifying corporate governance risk in the case of Latin American markets.

Carmona et al. [54] achieved 94% accuracy using XGBoost to predict the bank failure in the USA, between 2001-2015, being the best algorithm in this work comparing with results obtained with other different models such as the random forest, and logistic regression and obtained 92% and 84,21% of accuracy, respectively.

Barbosa [42], in his work, anticipated US recessions with probabilities above 0.5 in some cases, using logistic regression, random forest, XGBoost, and the Ensemble of these three. The best results predicting recessions with no lag were using the logistic regression model (Area Under the roc Curve (AUC) = 0.90). However, the random forest model obtained the best results in the remaining tests with 6, 12, and 18 months of anticipation with results of 0.9 for the AUC metric. XGBoost was considered the one with a more consistent prediction maintaining the quality of its predictions no matter the changes.

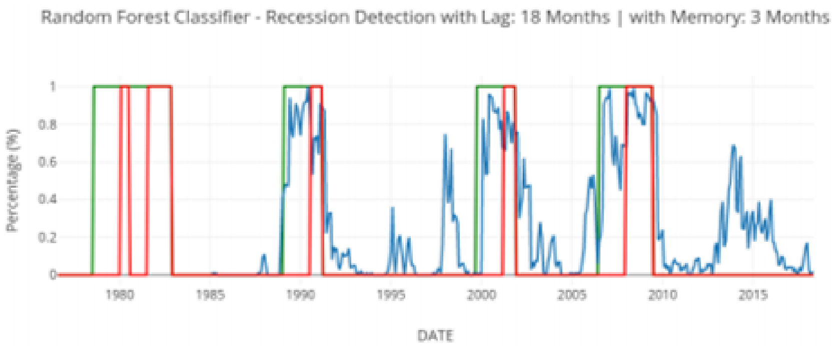


Figure 2.20: Graphic representation of probability results retrieved from models, trying to predict recessions throughout time [42].

Dey et al. [55], in his work, used XGBoost to predict financial securities trends and had 87% accuracy for the 60/90 days predictions. Yu [56], using XGBoost as well, achieved 85% accuracy in predicting the Chicago Board Options Exchange (CBOE) Volatility Index.

Barbosa [42] using logistic regression, random forest, and XGBoost and Liu and Moench [57] using probit models, both works with multivariate models obtained good metric values predicting economic recessions. Both these works, make similar approaches as the one adopted in this work, like the graphic representation of the time-series models' predictions, with probabilities of recession throughout time, and some of the metrics used as criteria evaluation (e.g., AUC and recall). Instead of trying to predict the downfalls in a single stock market, they went for the predictions of US economy recessions. In figures 2.20 and 2.21, graphic representations of their results are shown.

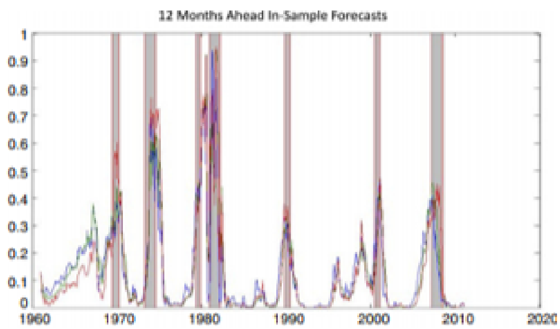


Figure 2.21: Graphic representation of probability results retrieved from models, trying to predict recessions throughout time [57].

Other examples of works in this area are using ANN and GA. The use of ANN was not well taken into account for this work, mainly because it is characterized by its "black box" nature - it provides little explanatory insight into the relative influence of the independent variables in the prediction process [58]. Lu [59], Zhang and Wu [60], Vanstone and Finnie [61] are examples of ANN works in this area. Neves and Horta produced several works using GA with relatively good metric results - [11, 21, 62]. However, for this binary classification problem, it was intended that for the prediction, other algorithms could constitute a better approach.

The works of Xia et al. [63], and Nobre and Neves [64] utilized techniques in order to optimize the hyperparameters of the XGBoost algorithm. In the work of Xia et al. [63] was observed that the

XGBoost solution with bayesian optimization outperformed other models, such as logistic regression and random forest, and techniques for optimization, such as grid-search and manual search. The work of Nobre and Neves [64] made a robust system combining Principal Component Analysis, Discrete Wavelet Transform, and XGBoost to trade in the financial markets. The XGBoost hyperparameters were optimized, resorting to the use of a GA. This system was able to obtain an accuracy higher than 50% in its predictions. Both works give a good indicator that optimizing the XGBoost algorithm is worth a try and allows for better results. However, based on other works found utilizing GAs in finance's purposes, the GAs were the approach chosen for this work. Also, Mori et al. [65] made several comparisons between both approaches, and the Bayesian Optimization ended up having worse results, which helped to support the choice made.

In the table 2.4 it is summarized all the referenced works, with the title of work, models used and results obtained.

This is the current state of art. Not all the works presented use these algorithms for the specific prediction of Bear Markets. However, it was found good results in forecasting other subjects also in the financial area with time-series analysis. Considering all the works presented, XGBoost and random forest had the best results, and appear to be a reasonable choice for the Machine Learning approach in the binary classification problem of this work. The use of logistic regression, is also often observed in several works, and will be considered for the approach in this work in order to have a linear model, differing from the other non-linear models chosen (random forest and XGBoost). In his work, Barbosa [42] used similar approaches like the ones chosen so far to apply in this work (XGBoost, random forest, and logistic regression), and as was already stated in the present section, good results were obtained in the prediction of recessions in the US economy, anticipating in some cases with probabilities above 0.5.

Table 2.4: Results of related works, using the models used in this work and other approaches.

Authors	Work Description	Models used	Best Results	References
Haoming Li, Zhijun Yang and Tianlun Li (2014)	Predict daily returns of U.S stocks based on their trading data and external financial indices	• Linear Regression • Logistic Regression • SVM	Logistic Regression with best results: Successfull rate of 56% with 2000% cumulative returns over 14 years.	[51]
Germán Creamer, Yoav Freund (2015)	Quantify the corporate governance risk in the case of Latin American markets	• Adaboost • Logistic Regression • Random Forest	All models with similar results.	[52]
Pedro Carmona, Francisco Climent, Alexandre Momparler (2017)	Predict bank failure in the U.S. banking sector	• Logistic Regression • Random Forest • XGBoost.	XGB with ~95% accuracy and ~98% AUC outperformed minimally Random forest. LR had the worst results.	[53]
Rodrigo Lopes do Ó Barbosa (2018)	Prediction of U.S. economic recessions	• Logistic Regression • Random Forest • XGBoost	LR the best predicting recessions with no lag (AUC=0.90). RF outperformed the remaining models for the remaining tests with AUC=0.90 in some test cases. XGB was the most consistent.	[41]
S Dey, Y Kumar, S Saha, S Basak (2016)	Predict the trend of stock market	• XGBoost	XGBoost obtained ~90% accuracy, and outperform results from other papers that used different models such as ANN, LR, RF and SVM	[54]
Michael Yu (2017)	Predict the future behaviour of the Chicago Board Options Exchange (CBOE) Volatility Index (VIX)	• XGBoost	XGBoost reached 75% in certain tests	[55]
Yufei Xiaa, Chuanzhe Liua, YuYing Li b, Nana Liu (2017)	A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring	• Focus on XGBoost optimized via Bayesian Optimization; • Other Machine learning well known used and also other optimization techniques.	XGBoost using Bayesian optimization technique outperformed the other models and techniques used for optimization	[62]
João Nobre, Rui Ferreira Neves (2019)	Combining Principal Component Analysis, Discrete Wavelet Transform and XGBoost to trade in the financial markets	• System Combining: • Principal Components Analysis; • Discrete Wavelet Transform; • XGBoost with GA optimization.	the analyzed systems were able to obtain an accuracy higher than 50% in its predictions	[63]

Chapter 3

Proposed System Architecture

The following chapter will explain the process to obtain results and the architecture implementation proposed for this work. To engage the problem stated, it is essential to make a reliable architecture. Decisions were made based on different conclusions taken from the concepts explained in Chapter 2. The process will be based in three different main steps - Data pre-processing, Classification process, Validation of results -, and the architecture will be constituted by several modules - Data Module, Transformation Module, Classification/Validation Module, and User Interface Module.

3.1 Process Executed by the System

The process executed by the system to obtain the final results, is based on three main steps: data pre-processing, classification process, and validation of the results.

In the data pre-processing step, firstly, there will be need to choose the economic variables to use in the classification process, that are figured in the Data Module. Then, a conversion will be made to all the raw data (S&P 500 index and economic variables) into *Python* dataframes and make transformations to this data according to algorithms' preferences. To get the anticipations in predictions, a lag will be added to the stock's dataframe, so that way, the models are able to try to correlate the different economic features used with time spans chosen before the actual market downfall events occur. Before it enters the classification section, the data will be divided into input and output dataframes. The conversion and transformation of data will be made by the transformation module.

The classification process will be made resorting to a time-series cross-validation method, with out-of-sample data tests, that will make a number of iterations decided by the user and split the data that number of times. In each iteration, the output will be updated, depending on which time instant the sliding window is, in the data. After this, it will be made the model's training. Three different Machine Learning algorithms will be used - logistic regression, random forest, XGBoost - plus the ensemble of these three, to produce models to predict the wanted events (market downfalls). The results acquired will be of probabilities of a market downfall event happens in medium-long term (depending on the lag added). In this process, the predictions made by the models and the features' importance values when training the model, will be saved in different dataframes for further utilization.

It will also be tested the possibility of optimizing algorithm's results through the optimization of the hyperparameters resorting to an implemented GA.

In the optimization process, instead of using the XGBoost method with default hyperparameters, it will firstly proceed to a hyperparameters' optimization through the use of a Genetic Algorithm (GA), expecting in the end, to optimize the results. As it was explained in the chapter background 2, in addition to the training and test subsamples of the data, in these sorts of cases will exist one more subset called validation subset, so it is possible to validate the best hyperparameters without testing with the same training subset of data.

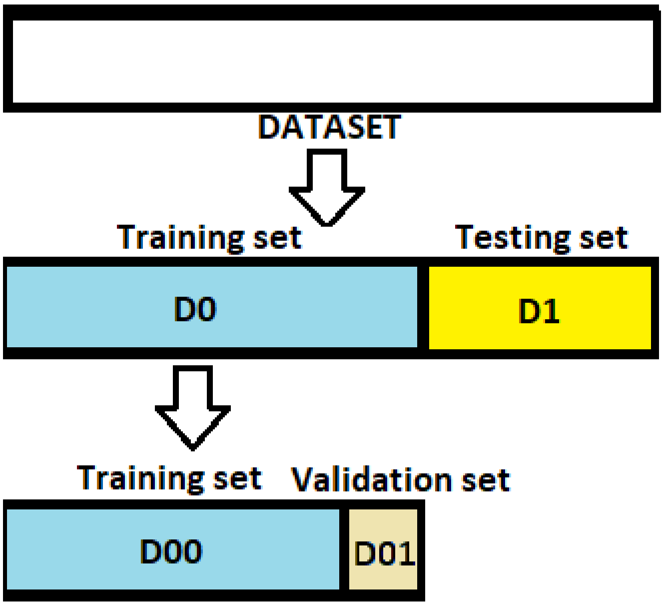


Figure 3.1: Example of data partition with hyperparameters tuning.

After this partition, the GA will initiate the process and choose the best values for the parameters that will be further used in the simulations with the testing subset of data. The GA will consist of 5 different steps that were already described in the background chapter 2 in the section 2.3.4: initialize population, train population, selection, crossover, and mutation.

In the end, passing to the results' validation step, by using the dataframes containing the models' results it will be plot a graphic representation of a time-series with the results and calculated the evaluation criteria (metrics) in order to have an understanding of the results retrieved from the models and to draw conclusions from them. Different metrics will be used, such as the Confusion Matrix, accuracy, recall, precision, AUC, and the integral of the results under the "True Positive" area (AUTP).

The classification process (including the hyperparameters tuning) and validation of results are steps executed by the Classification/Validation Module.

The diagram of the figure 3.2 presents all this process explained, that will be used for the several case.

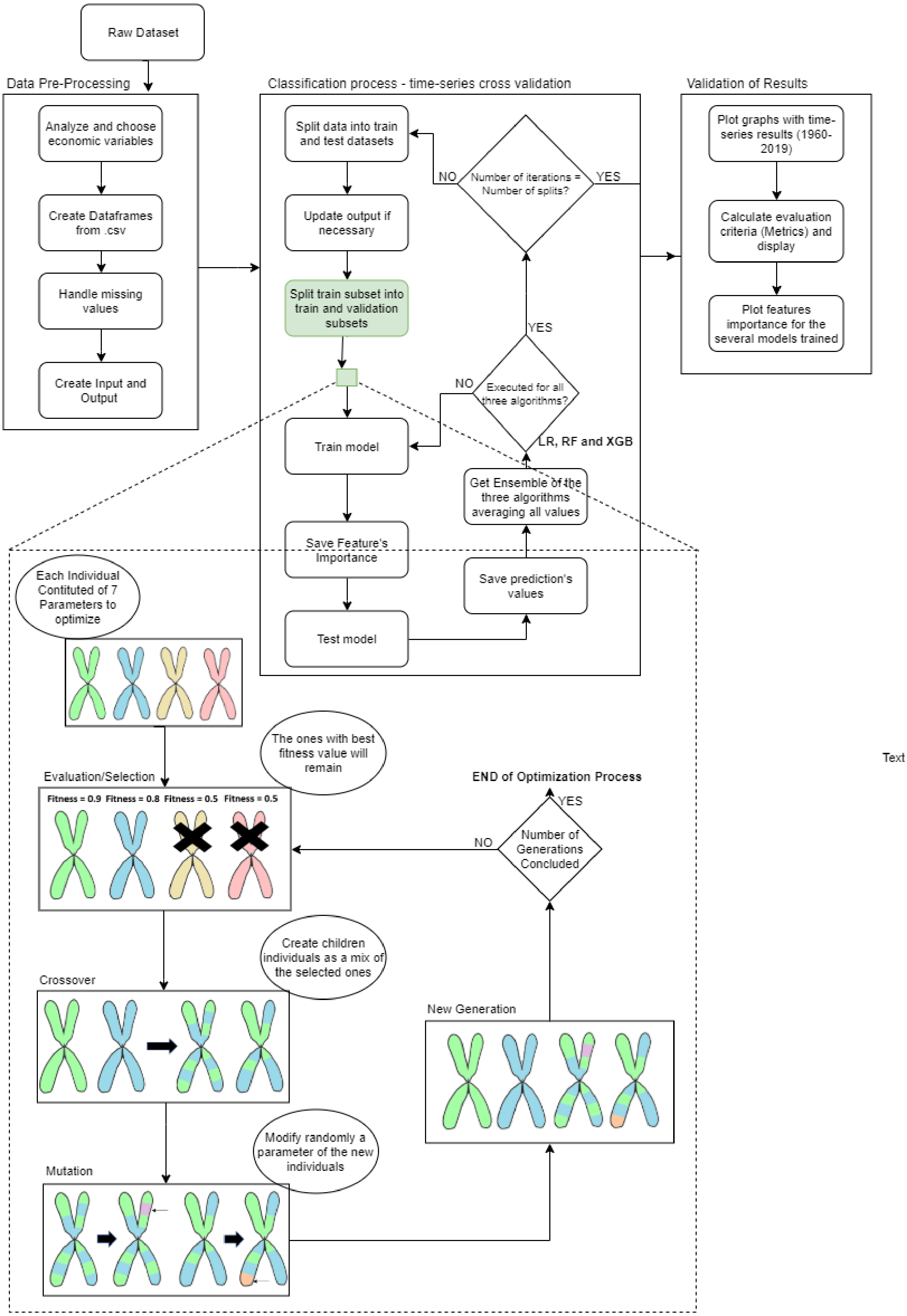


Figure 3.2: Diagram of the whole process made by the system implemented. In the Classification Process, it is marked in green colour, the addictions made for the implementation of the Genetic Algorithm.

3.2 System Architecture Implementation

An architecture containing modules will be used in order to execute the process explained and get the different sections connected and working independently. This architecture will lead to a more organized system, where throughout the development will be easier to detect errors and to prevent the propagation of them. If it is necessary to add or modify the different modules, this can be made separately without compromising the whole system functionality.

The modules that constitute the system are the following four: Data module, Transformation Module, Classification/Validation Module, and User Interface Module. The architecture proposed is represented by figure 3.3 and the description of each module is made in the following sections.

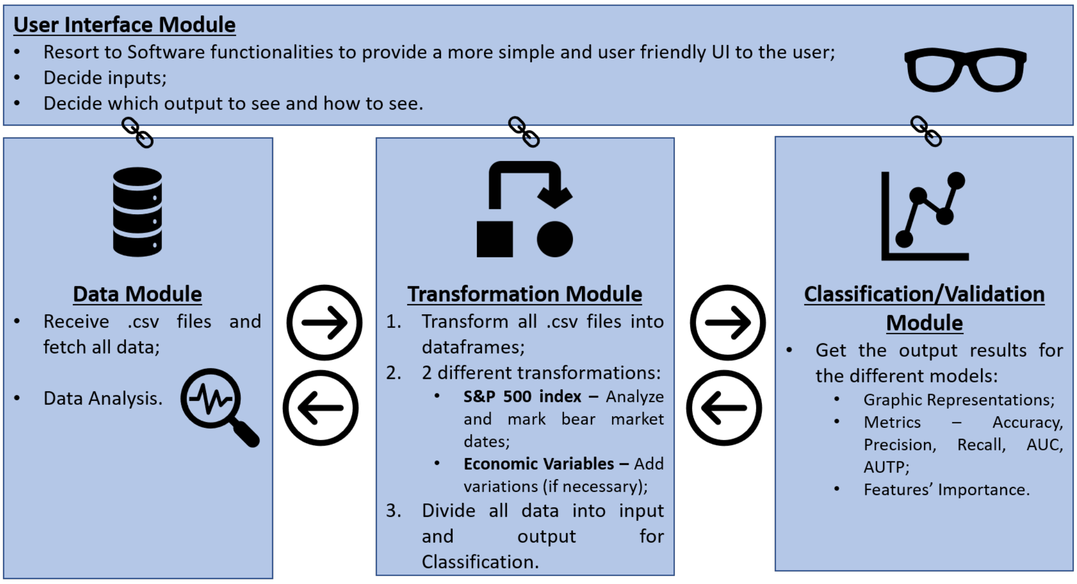


Figure 3.3: Module-based architecture representation, with the functionalities executed by each module.

3.2.1 Data Module implementation

This section will be figured the data obtained throughout several types of research. This data will include the different economic variables information as well as the S&P 500 index price, historical data mostly from 1960. This module will have an analysis part as well, regarding all these data obtained, and to help the user in making further choices for the classification module.

3.2.1.1 Stock and Features' acquirement

In the State of Art Section, it is mentioned several economic variables used not only in other research works made by particulars - [3, 42, 51] - but also by relevant financial institutions - BOFAML and GS (tables 2.2 and 2.3). The goal at first was to find and store as many variables of these, intending to use them as features and combine them for further classifications using Machine Learning models. In the following table 3.1, there is an example of some of the economic variables that will be used in this work, with the respective description.

Table 3.1: Economic variables arranged for this work.

Indicator name	Type	Description
CBOE Volatility Index	Sentiment	Measures market expectation of near term volatility conveyed by stock index option prices.
Conference Board Consumer confidence	Sentiment	Reflects prevailing business conditions and likely developments for the months ahead. This monthly report details consumer attitudes and buying intentions, with data available by age, income, and region.
Core Inflation	Consumer Prices	The change in prices of goods and services except those from the food and energy sectors.
CRB Raw Industrials Index	Returns	Measure of price movements of 22 sensitive basic commodities whose markets are presumed to be among the first to be influenced by changes in economic conditions.
Credit Tightening from banks for Commercial and Industrial loans	Credit	Net Percentage of Domestic Banks Tightening Standards for Commercial and Industrial Loans to Large and Middle-Market Firms.
Effective Federal Funds Rate	Credit	Interest rate at which depository institutions trade federal funds (balances held at Federal Reserve Banks) with each other overnight.
Growth outperforming S&P 500 equalweighted	Returns	Returns of 6 and 12 months were higher by S&P 500 Growth Index than S&P 500 equalweighte.
Institute for Supply Management (ISM) Purchasing Managers' Index	Production & Business Activity	Ccomposite index based on the diffusion indexes of five of the indexes with equal weights: New Orders (seasonally adjusted), Production (seasonally adjusted), Employment (seasonally adjusted), Supplier Deliveries (seasonally adjusted), and Inventories.
Misery Index	Economic Activity	Unemployment rate plus yearly percent change in CPI.
Momentum outperforming S&P 500 equalweighted	Returns	Returns of 6 and 12 months were higher by S&P 500 Momentum Index than S&P 500 equalweighted.
S&P 500 12 months returns	Returns	Percent of 12 months return in S&P 500.
S&P 500 24 months returns	Returns	Percent of 24 months return in S&P 500.
Shiller P/E S&P 500	Valuation	Price earnings ratio is based on average inflation-adjusted earnings from the previous 10 years.
Rule of 20	Valuation	Market P/E + annual inflation (Y%).
Unemployment rate	Labor	Number of unemployed people as a percentage of the labour force.
US Job Claims	Labor	U.S. Employment and Training Administration, Initial Claims.
Yield Curve (10Y-2Y)	Sentiment	Series is calculated as the spread between 10-Year Treasury Constant Maturity and 2-Year Treasury Constant Maturity.
Yield Curve (10Y-3M)	Sentiment	Series is calculated as the spread between 10-Year Treasury Constant Maturity and 3-Month Treasury Constant Maturity.
5% Pullback over prior 12 months	Returns	Number of stock decreases (over 5%) in the preceding 12 months of the bull market peak.

To obtain these several variables from table 3.1, it was made the access to websites and terminals with database access that contain several economic variables. Examples are the Federal Reserve Bank of St. Louis website (FRED) [66], Yahoo Finance website [67], and computer terminals with access to Bloombergś and Thomson Reuters Eikonś databases [68, 69]. The files are only composed of *.csv* files and can be retrieved directly.

Besides gathering all the economic variables that will be used as features, it was needed the stock price's historical data as well, where will be focused on the study of bear markets. As was already stated, the S&P 500 was the chosen market to analyze. The database of the S&P 500 index with all the price's variations throughout the years could be obtained from Bloomberg's databases (figure 3.4) [68].

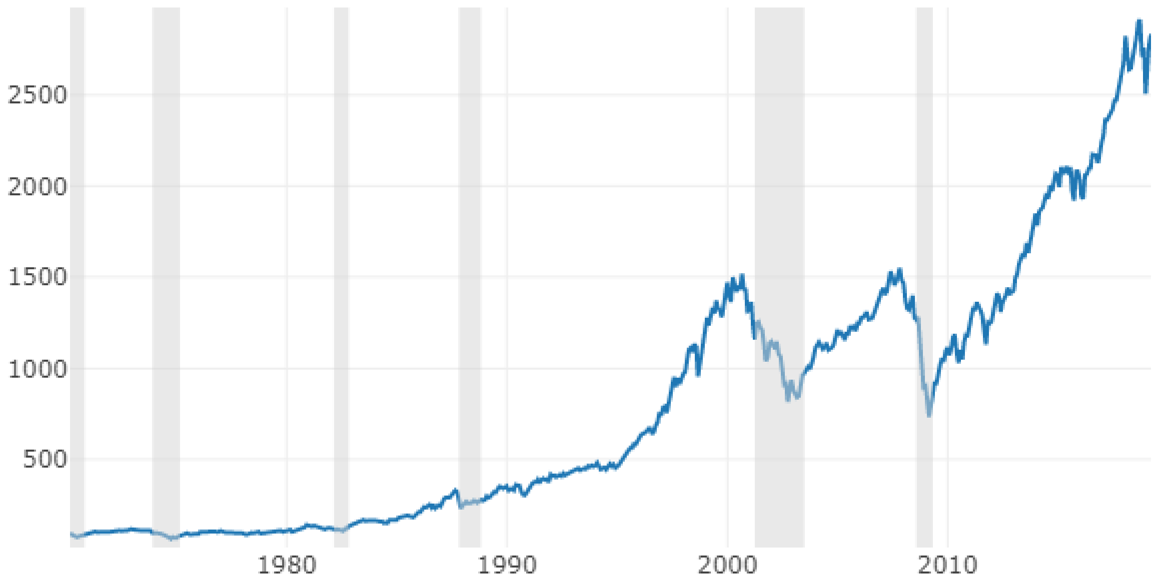


Figure 3.4: S&P 500 index price since 1970 with Bear Markets marked as shades.

3.2.1.2 Data Analysis

Regarding this module, it will also be possible to do further analysis of the economic variables' characteristics or on the index price values. It will be possible to analyze each variable independently or to analyze the relations between all variables (correlation).

Firstly it will be possible to represent the description of the signal in several values (e.g., max, min, mean) regarding itself (figure 3.5). In order to facilitate the signal's analysis, it is also possible to get a graphic representation of the indicator's historical data in a time-series format. Resorting to the use of libraries such as *Plotly* [70], it was made a function called *plot_with_bmpts()*, which plots the signal along with the market downfall events marked (market price decreases of 20%, 18%, and 15%). It is essential to obtain a representation of the indicator, having signalized the bear market periods over time, in order to be able to make prior assumptions about the correlation of each economic variable with the market falls, just by these analyses (figure 3.5). In figure 3.6, there is an example of this method output for CBOE Volatility Index: VIX.

```
Name: VIXm, dtype: float64
count    390.000000
mean      19.739274
std        6.126560
min       10.764000
25%       14.477000
50%       18.645500
75%       24.029750
max       42.166000
```

Figure 3.5: Output of *describe()* method for the economic variable CBOE Volatility Index: VIX.

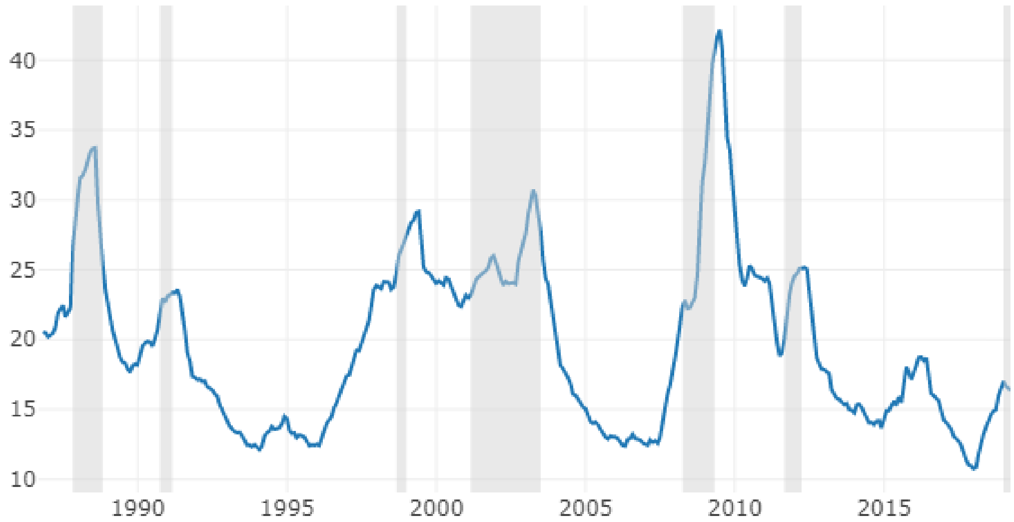


Figure 3.6: Output of method *plot_with_bmpts()* for CBOE Volatility Index: VIX. Plot with 18% market downfall events marked as shades.

It will also be possible to analyze the variables in a general manner. In order to help to find a correlation between signals and if possible, reduce unnecessary variables that can have similar behavior through time as others. With a method from *Pandas* library [71], *corr()*, it is possible to compute pairwise correlation of columns (features) that can be plot as a heat map resorting to *seaborn* library's methods (figure 3.7).

The user will be allowed to choose the indicators and the time intervals to use in further classification. For this choice, the prior use of these "data analyst" features of this Module, could help to achieve better outputs since this way a well-founded choice can be made. Removing from the choices variables that have a high correlation with others or that at first sight compared with the S&P 500, it does not have any correlation with the market downfall events.

After an explanation of the storing procedures and analysis and referred to the several variables retrieved to be used as classification features, in the next section, it will be approached the phase that consists in the transformation of data. Moreover, it will allow the user to handle the data and use it in several methods called in for the different modules of this work.

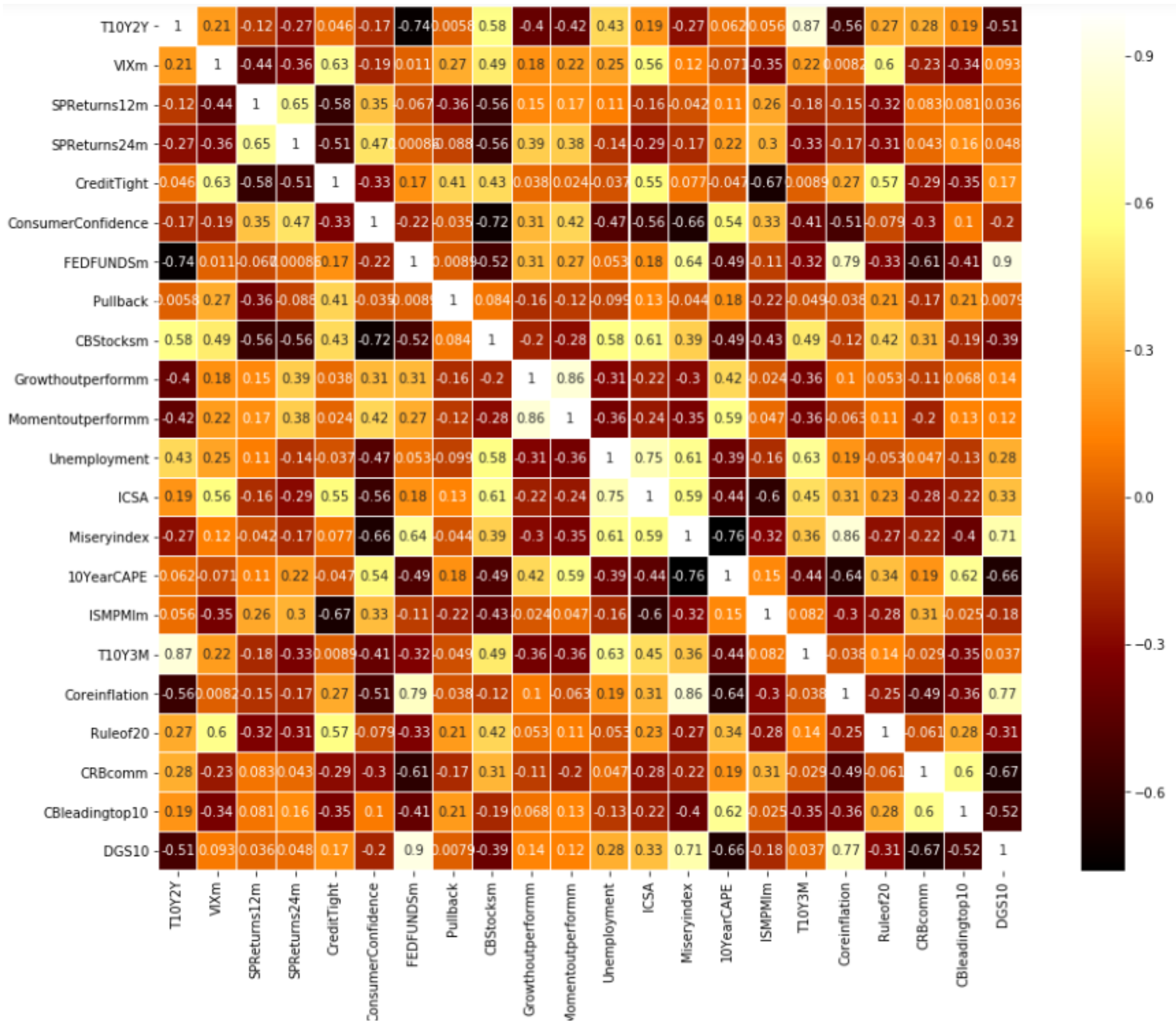


Figure 3.7: Correlation between the several economic variables used for this work.

.

3.2.2 Transformation Module Implementation

This section will firstly guarantee the transformation of all data into *Python* Dataframes. It will also permit the modification of the different variables in order to be in the right type for the classification (stationarity). Hereupon it will be merged the different data (indicators and S&P 500 index), and coherency will be obtained regarding the time interval and granularity in which is evaluated. Finally, it will guarantee the transformation of these into input and output before it enters in the classification section.

3.2.2.1 Convert .csv files

Firstly it will be converted all .csv files into *Python* Dataframes so that it can be used in the further features in the modules of this work. To convert the .csv files into *python* Dataframes, it will be made the resort to the method *read_csv()* from *Pandas* library [71]. A Dataframe is a Two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns) [72] (figure 3.8). This data type will facilitate some operations to modify the data, for example, to merge or

	DATE	VIXvar	ConsumerConfidencevar	ICSAvar	Pullback	Miseryindexvar	10YearCAPEvar	Ruleof20
0	1960-01-29	1.007067	0.011430	0.609195	0.0	-0.007931	6.505567	22.839755
...	...	...	...	...	...	...	...	...
681	2016-10-31	-15.687547	0.399404	-4.232192	4.0	0.377293	4.493159	25.464598
682	2016-11-30	-20.296368	0.520155	-5.140025	4.0	0.384690	5.718228	25.564121
683	2016-12-30	-18.176075	0.520155	-6.220977	2.0	0.339416	6.752261	25.626302

Figure 3.8: Example of a Dataframe time-series type, with historical values for several economic indicators.

split two different Dataframes. It is also a data type accepted by the several classification methods used in this work.

3.2.2.2 Stock price decrease dates and lag addiction

As it was already stated, in the present work, the detection of significant market downfalls will be focused on the S&P 500 index. Thus, it will be necessary to analyze this index and save in Dataframes the dates of significant decreases in the stock price and then add a lag to these downfalls. This process will allow that in the classification layer, the models used can evaluate the weights of each indicator for different periods before the event of downfall happens. It will be considered a bear market, from the moment where there was a 20% (or 18% or 15%, depending on the test case) drop in stock's price, considering that this bear trend has ended when there is a 25% price increase from the lowest point during the bear market period (figure 3.9). This calculation will be made through the created method *bmDates()* and will return a new dataframe with the labels of these market downfall events marked.

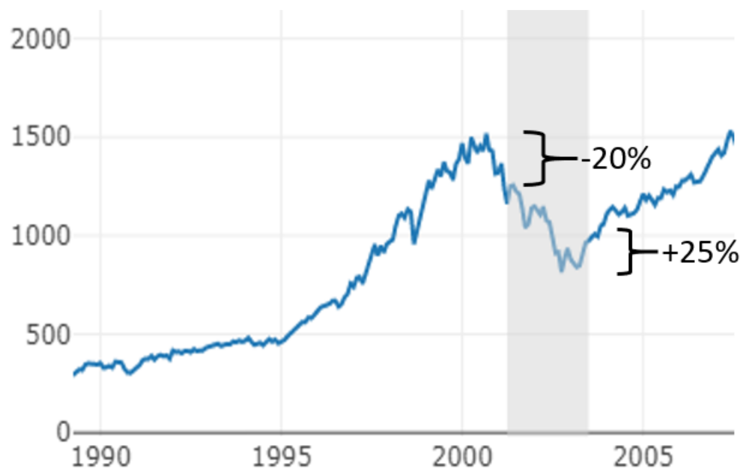


Figure 3.9: Graphic representation of the bear market timeframe considered for this work.

The user will have the possibility to choose the lag size to apply in the index downfalls before the classification process begins. In figure 3.10 is displayed an example of what is intended to obtain after lagging the signal, with 6 and 12 months of lad added.

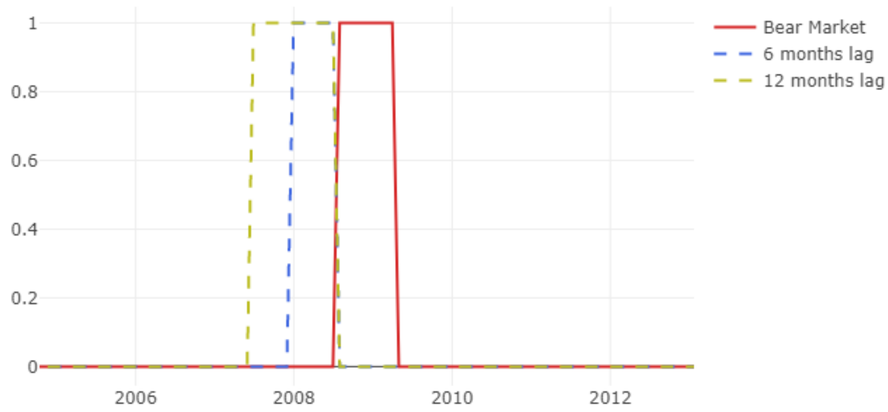


Figure 3.10: Example of the lag applied to 2008's Bear Market.

3.2.2.3 Features' transformation

As it was already explained in the background section 2.2, in order to make a multivariate forecast in the S&P 500 index, firstly, it will be needed to verify if each variable used is stationary. If not, it will be necessary to proceed to a transformation. The method *Dickey_Fouller()* from the *statsmodels* library [73] will be used to check if a variable is stationary or not.

To turn the variable from non-stationary into stationary, two different methods will be used: From *Pandas* library [71], the *rolling()* method will allow to obtain a smoother time-series with the mean value of the previous x chosen instants (figure 3.11); In order to remove a time-series trend, a created method *timeseries_variation()* will make a new time-series with the values of x instant variations. Instead of getting the absolute value of the time-series, it will end up as the value of variations within the chosen time span (figure 3.12).

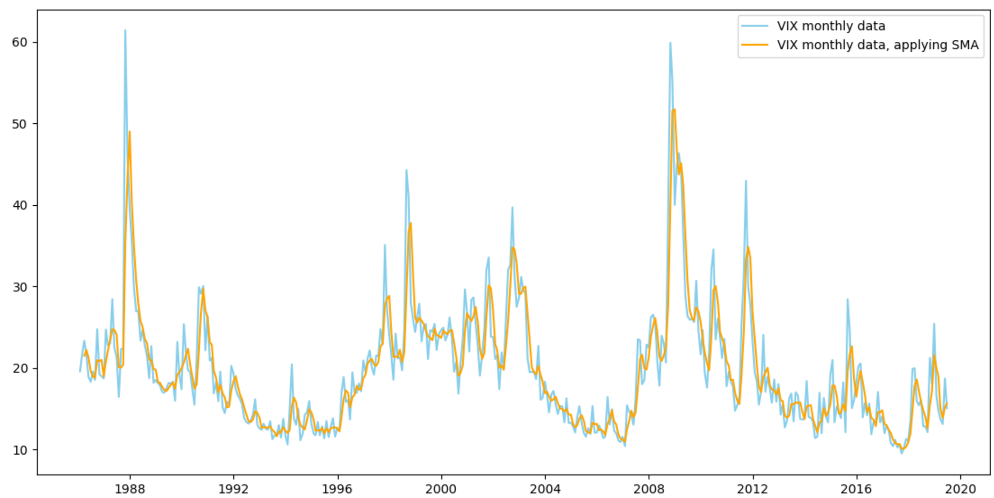


Figure 3.11: Example of SMA applied into a variable.

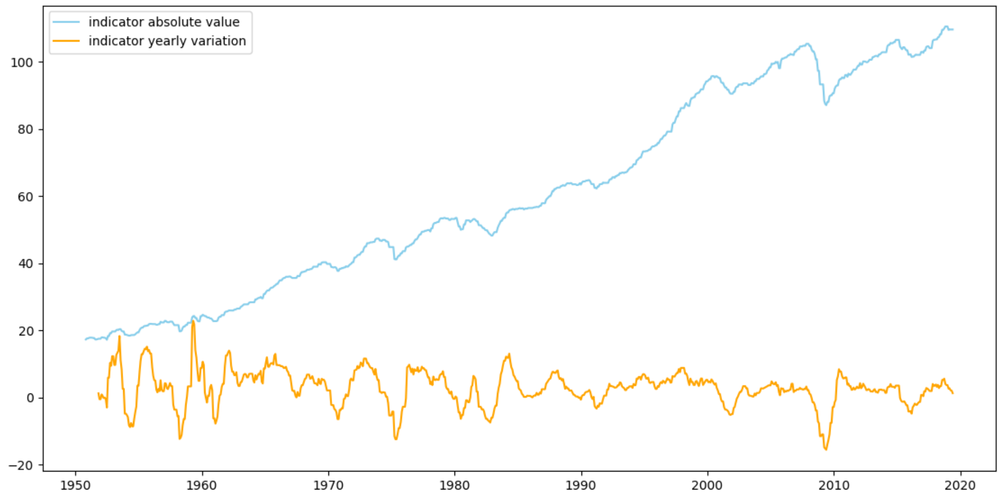


Figure 3.12: Example of a time-series with the indicator (blue and nonstationary) and its variation (orange and stationary).

3.2.2.4 Handle missing values in Dataframe

After choosing the desired features from the data module and adding the lag to the stock index signal, it will be made a merge to get a dataframe that contains the whole data (method *merge_dates_indicators*).

In this work, it will be used three Machine Learning different algorithms: logistic regression, random forest, and xgboost. There is no equal history available for each economic data arranged for this work. Consequently, it will be necessary to find a way to handle the existent missing values in data, order to use it in the algorithms. The xgboost is the only algorithm that accepts missing values when training models. In the remaining methods, the criterion to fill the time fields with missing values is primarily, to choose the closest previous value of the same variable and if there is none, the type of fill will differ for each algorithm:

- For the random forest, the none values will be filled with the average of all the other values of the variable in question;
- For the logistic regression, will substitute the none values with zero.

These choices were decided empirically, having into account the experiences with better results. This process will be made resorting to a created method *merge_dates_indicators()* which will use the method *fill()* from the *Pandas* library.

3.2.2.5 Create Input and Output for the Classification

After having the whole data prepared in one dataframe, it is necessary to divide this into input and output dataframes to use in the following classification process.

The input will consist of a time-series dataframe with the historical data from 1960 to 2019 of each economic feature chosen. The output labels will be classified as a positive class inside pre-downfall and downfall zones, and negative class in the remaining timeframes of the dataframe. However, the output dataframe will enter the classification process with only negative outputs and then will be updated throughout the process. This updating process will be better explained in the Classification/Validation Module section.

The next section will describe the final module that will show the results for what is pretended in this work - Classification/Validation Module.

3.2.3 Classification/Validation Module Implementation

This module will be responsible for receiving the datasets regarding the different variables chosen as features and transformed into an input dataframe. Also, handle S&P 500 index already marked with the price decrease events in which the classification labels will be based on (output), so in the end, the models are tested producing forecasts anticipating significant downfall events in the S&P 500 index, from 1970 to 2019.

The response gave by this module will be achieved using several models produced by computation algorithms, already described in the Chapter 2. The user will be getting the response of classification for the three different models (logistic regression, random forest, and xgboost), plus the average of these (Ensemble approach).

The different labels that will correspond to the model's output will be in binary form. As was already stated in the previous section 3.2.2, the criterion chosen to differ the positive and negative label, will stand on:

- Positive Labels - If an instant is inside a zone of price downfall event (can be for -20%, -18% or -15%) or inside a time span before this downfall, that corresponds to the lag added. The lag's lenght is a parameter decided by the user. If the classification is targeted to catch 20% downfalls with six months of lag, then the labels will be positive six months before the 20% fall was a reach, until the label that marks a 25% increase from the lowest point during this downtrend phase (end of bear period);
- Negative Labels - Remaining periods which are not inside price downfall or pre-downfall event zones.

In the figure 3.15 is an example of an updated output with time spans turning into positive labels.

It will make sense to use a cross-validation method, to ensure that the models produced are timeless validated, i.e., the models could make reasonable forecasts for different moments in both older and recent times. It also helps the user to detect deficits in the system, such as overfitting or undesirable features.

3.2.3.1 Time-series Cross-Validation

An explanation of cross-validation method was given in section 2.3. However, considering that the prediction strategy of the present work involves the analysis of time-series data, traditional cross-validation (like k-fold) should not be used. Time dependency is one drawback, i.e., with time-series data, the split of training subset should be done using a different approach, since the predictor must retain all the information chronologically about an event after it occurs, in order to fit the model. When the choice of a

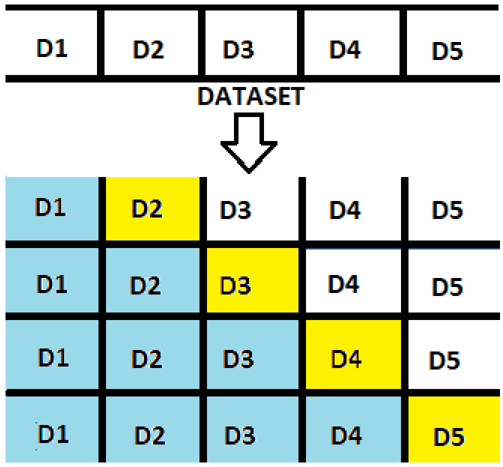


Figure 3.13: Illustration of time-series cross validation method. Training dataset (blue) and test dataset (yellow).

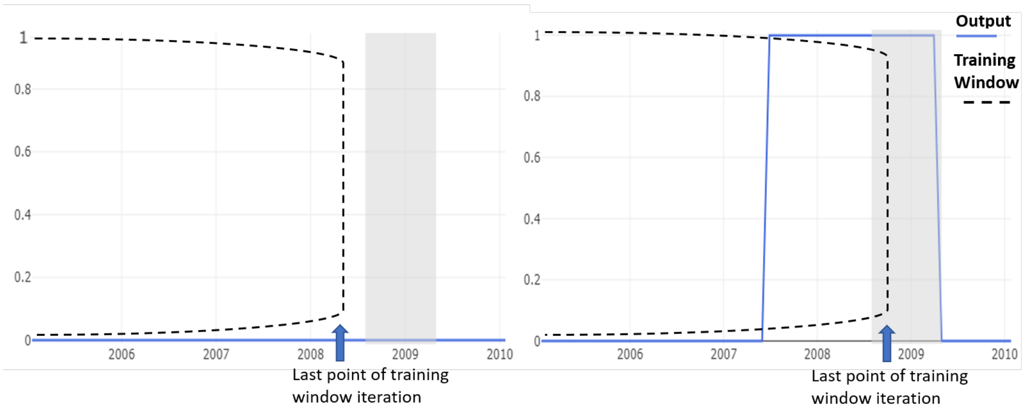


Figure 3.14: Illustration of the sliding window method and the effect in the output's labels.

test set is arbitrary, it may mean that the test set error is a poor estimation of an error on an autonomous test set.

In the time-series cross-validation method of this work, the training subsample of data will be split into time units (e.g., days, months). Each time unit is considered as a test subset, and the previous time units will be the training subset. The training subset starts with a minimum number of observations and uses the following timeframes of data to test the model. This will be made throughout the whole data set and ensures that the time-series aspect of the data is considered for prediction.

From python library *Sklearn* [74] it is possible to simulate this kind of algorithm for all models, through the *TimeSeriesSplit()* method which allow this division of the dataset.

The best way to get this cross-validation simulation closer to reality is to start the classification process with only negative labels and update it throughout the process. The window of training is going to enlarge in each new iteration, and it is intended that the user only knows that it is inside a positive class label instant, after the window gets inside the initial point of a significant downfall period, in the training process. Before this, it is not possible to know whether it is inside a lagged zone of a further downfall event (figure 3.14).

In the figure 3.15 is an example of the time-series cross-validation method with K=3 (divided the dataset into three parts) adapted to this work, with the respective output given by the algorithm models and the big downfall events and respective lagged zones marked.

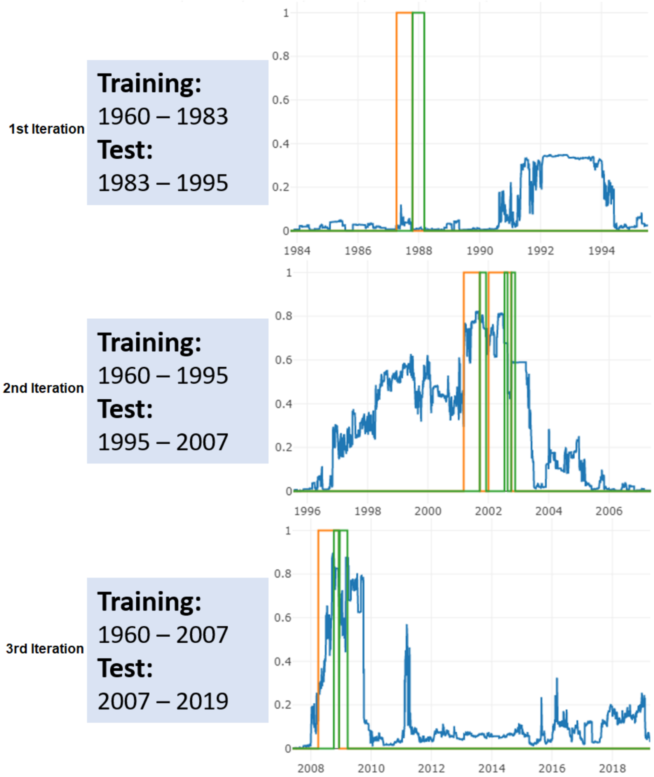


Figure 3.15: Illustration of the time-series cross validation approach used in the present work, with K=3.

3.2.3.2 Classification process

After obtaining input (features) and output (class labels) dataframes, the classification process will resort to several *Python* methods to help in this matter. In order to initiate any classification process, the dataframes will be divided (as was already explained in the time-series cross-validation section) in as many parts as the user decide using *TimeSeriesSplit()* method. Besides the normal classification process where the algorithms are used with defaults hyperparameters (created method *classification_simple()*), there will be the option to make an alternative classification for the xgboost algorithm with optimized hyperparameters (created method *classification_tuning()*). In the simple classification, it will be made the number of iterations equal to the number of divided parts chosen by the user. The classification with tuning will take a little longer due to the number of generations created in the GA. Approximately it will have a complexity of $number\ of\ iterations \times number\ of\ generations$.

In each iteration, the input and output subsets will be divided into train and test subsets (and validation subset if there will be any hyperparameter tuning). Then it will verify if the first point of the new training part as already passed any bear market date, through the method *newBMdates()*, and if so, it will update the output training labels.

The *Python* methods used to produce the different models will differ with the algorithms, using the *LogisticRegression()* for the linear approach and *RandomForestClassifier()* and *XGBClassifier()* for the nonlinear approach. All of these methods can be imported from the *Sklearn* library [74]. After training these models, it will be obtained a value of probability for each point of the testing dataset using the method *predict_proba()*, available for the several algorithms. These values will be stored in dataframes for each algorithm, and the average results of these three (ensemble approach). It will also be saved in four different dataframes the features' importance values, calculated by the method *feature_importance()* in the non-linear algorithms (random forest and XG) and *coef_()* in the linear algorithm (logistic regression).

For the case with tuning classification, the classification process will be slightly different, with the addiction of several methods created for the GA. The GA will consist in 5 different steps that were already described in the background chapter 2 in the section 2.3.4, that will have the respective *Python* methods: initialize genes population (method *initilialize_population()*), train population (method *train_population()*), new parents selection (method *new_parents_selection()*), crossover (method *crossover uniform()*) and mutation (method *mutation()*). The created method *OptimizeXGBoost()* will be responsible to gather all these methods and guarantee the correct interaction and behavior of the different GA steps.

At the end of all iterations, the final four dataframes (logistic regression, random forest, xgboost, and Ensemble approach) will detain the results of all iterations through a merging process as well as four python dictionaries with the features' importance for all the models produced in each iteration.

3.2.3.3 Validation of the results

The four dataframes created containing the results of classification can be displayed using methods from the *Plotly* library [70], which make the graphic representation of the results for all the models throughout time, from 1970 until 2019, with the market downfalls and respective lags marked. These graphic representations will help to make an intuitive analysis of the results, comparing the probabilities of each model with the market downfalls' dates.

The four dataframes created containing the results of classification will also be used to calculate the metrics resorting to methods from the *Sklearn* library. These results will be constituted by the results for different metrics used, such as the confusion matrix, accuracy, precision, recall, AUC, and AUTP, for an analytic examination of the results (figure 3.16).

Through the use of *feature Importance* methods of *Sklearn* [74] it will also be possible to verify which are the features that had greater importance in the construction of a model by the different algorithms, examining the features' importance for the four models in each iteration when a new model is trained. This tool will only be possible to execute after the classification process where will be saved as a *Python* dictionary, the feature importance for different models produced in different periods of the classification process. A created method made called *printFeaturesFromDict()* will receive a chosen dictionary and plot it (figure 3.17) through the use of methods from *Plotly* library [70].

```
ROC-AUC:
LR: 0.892712059171023
RF: 0.8259180733811321
XG: 0.8155113535209672
avg: 0.8800394905077119

AUTP:
LR: 0.6438209051640817
RF: 0.3776774193548387
XG: 0.36811821296953806
avg: 0.4632055124961529

Confusion matrix with 0.5 threshold:

ConfusionLR:
 [360  91]
 [25  112] accuracy: 0.8027210884353742 recall: 0.6129032258064516 precision: 0.5517241379310345

ConfusionRF:
 [392  59]
 [55   82] accuracy: 0.8061224489795918 recall: 0.27419354838709675 precision: 0.5815602836879432

ConfusionXG:
 [398  53]
 [54   83] accuracy: 0.8180272108843537 recall: 0.3870967741935484 precision: 0.6102941176470589

Confusionavg:
 [389  62]
 [33  104] accuracy: 0.8384353741496599 recall: 0.5161290322580645 precision: 0.6265060240963856

Confusion matrix with 0.35 threshold:
```

...

Figure 3.16: Example of the metrics results of all models, displayed after classification process.

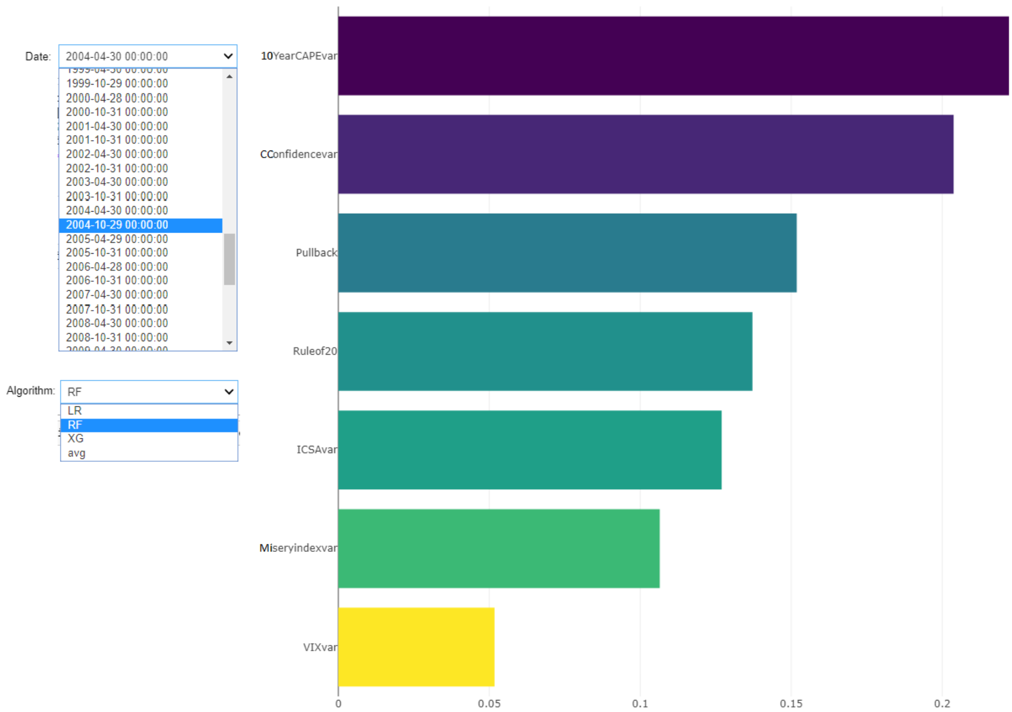


Figure 3.17: Example of input and output of method *printFeaturesFromDict()*.

This tool will be valuable in a way that can compare the feature importance for different dates, helping not only to analyze and interpret the results by the algorithms but also to see which economic variables are more important in the detection of the several market downfall events.

3.2.4 User Interface (UI) Module Implementation

This section is connected to all the different layers explained above. The goal is to take advantage of the *Jupyter Notebook* functionalities and create a simple and user-friendly UI that allows the user to handle the different parts of the process that were already mentioned above.

It was taken advantage of the different functionalities of the *Jupyter Notebook* Platform. Dividing the code into differently named sections and several widgets such as the Select widget, or dropdown widget are two examples of it. These widgets can be made by methods from the library *ipywidgets* [75]. The figure 3.19 displays examples of how the user can choose the features for classification, resorting to the select widget without changing the code.

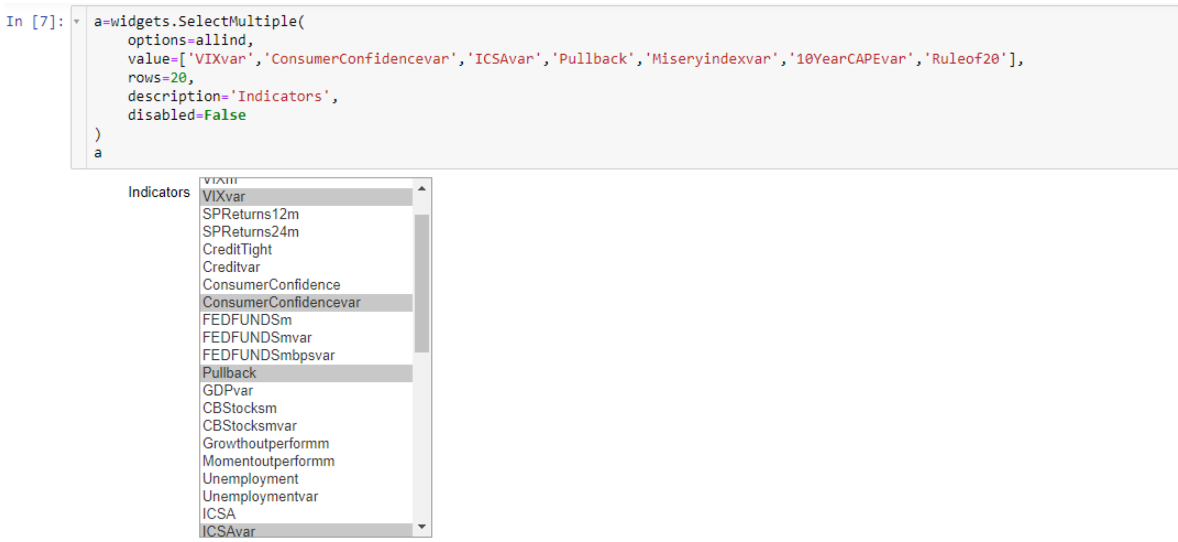


Figure 3.18: Select widget used for features' selection.

The following figure displays an example of how the dropdown widget is used for the selection of an economic variable to represent graphically.

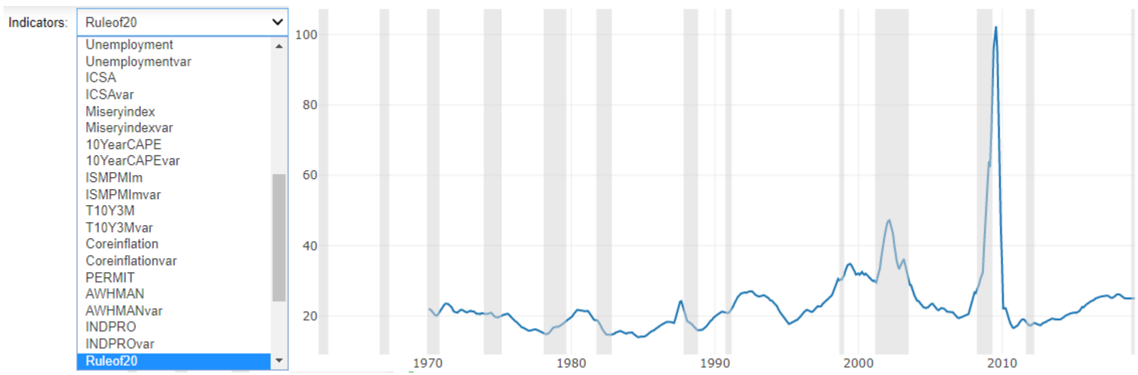


Figure 3.19: Dropdown widget used for economic variable selection.

Also, the *Plotly* library [70] gives the possibility to have graphic representations, with different functionalities such as zoom in/out and just by hovering above any time instant with the computer mouse, the graph returns the values corresponding to that instant (figure 3.20). This great-looking and fullyinteractive forms of graphic representation will facilitate the user job analyzing the different results.

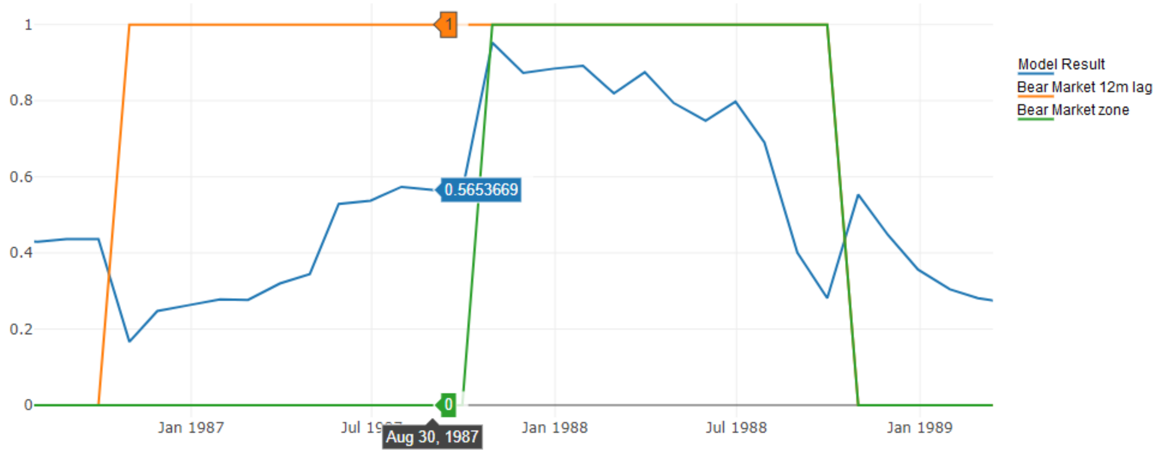


Figure 3.20: Example of interactive graphic representation with time-series model results.

Chapter 4

Results

The main goal of the present work will be to anticipate bear markets (-20%) and other significant price decreases (-18% and -15%) in S&P 500 index. There are several steps needed to make in order to accomplish the final goal of this work. The methodology used for the different case studies will be summarized in this chapter, although all the functionalities included, were already explained in the Architecture Implementation section 3. Also, it will be explained the different metrics used as evaluation criteria for this work. Finally, it will be displayed the results obtained for the different case studies as well as the conclusions and explanations drawn from these results.

4.1 Metrics

In order to have an analytic analysis to compare the results in the different models used and in the several case studies made, different metrics will be used to obtain a valid evaluation criterion. The metrics chosen were: Confusion Matrix, Accuracy, Precision, Recall, Area Under the ROC curve (AUC) and Area Under True Positives (AUTP).

4.1.1 Confusion Matrix

A confusion matrix comprises the results of four important classification concepts - True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) - into a matrix (figure 4.1).

TP result is when the result of the prediction is positive, being the training label also positive. **FP** result is when the result of the prediction is false, being the training label positive. **TN** result is when the result of the prediction is negative, being the training label also negative, and finally, the **FN** result is when the result of the prediction is positive, being the training label negative.

Adapting these concepts to the present work, the positive results correspond to zones of market downfalls or pre-downfalls, and the negative results correspond to the remaining timeframes where these events do not happen.

This metric allows the observation of results in a more user-friendly and intuitive way. The metrics that will be explained below have the formulas constituted by these confusion matrix concepts.

	Predicted: Negative(0)	Predicted: Positive(1)
Actual: Negative(0)	TN	FP
Actual: Positive(1)	FN	TP

Figure 4.1: Constitution of a confusion matrix.

Accuracy is defined as the sum of true results divided by the number of false results, both positive and negative (expression 4.1).

$$Accuracy = \frac{TP + TN}{FP + FN} \tag{4.1}$$

It gives the rate of results well predicted, taking into account a certain threshold that will separate positive and negative results.

Precision is given by the number of TP results divided by the sum of TP and FP results (expression 4.2).

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

This metric gives a rate on which it is possible to verify how precise/accurate was the model in terms of positive results, i.e., how many positive labeled results are actually positive.

Recall is as the TP results divided by the number of TP and FN results (expression 4.3).

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

This metric gives a rate on which it is possible to verify how many positive labels the model captured. The recall is also known as the true positive rate.

4.1.2 Area under the ROC Curve (AUC)

The ROC curve is a graph that plots the true positive rate as a function of false positive rate ($= \frac{FP}{FP+TN}$). The false positive rate in opposite to the recall gives a result on how many negative labels the models captured.

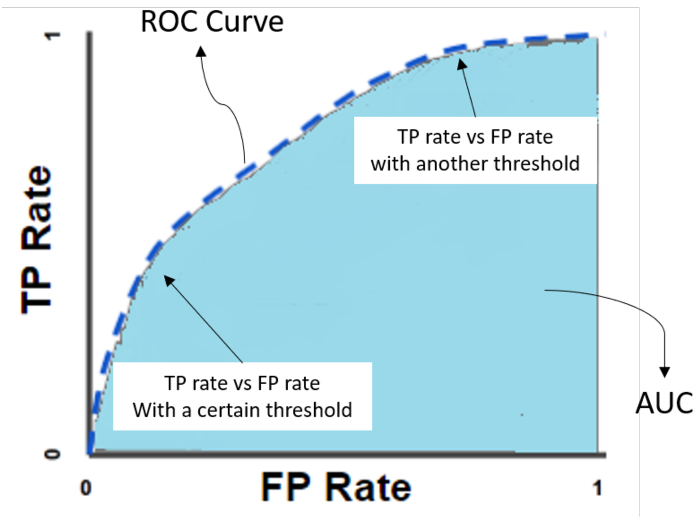


Figure 4.2: ROC (discontinuous line) and AUC (blue filled area) illustration with concepts explained.

The curve will be set, having into account different thresholds, and it is used as a performance measurement for classification problems. Decreasing the threshold, it is expected that the number of TP and FP also decreases, and vice versa, when increasing the threshold. The figure 4.2 displays an example where the ROC curve is drawn.

The AUC is defined as the area (integral) inside the ROC curve. It will provide a probability that the model ranks a random positive example more highly than a random negative example. The AUC is classification-threshold-invariant, i.e., measures the quality of the model's predictions irrespective of what classification threshold is chosen. In figure 4.2, it is possible to see an example where AUC is the light-blue filled area under the ROC curve.

4.1.3 Area Under TP (AUTP)

The AUTP is defined by the sum (integral) of probability results inside the TP labeled zones divided by the total area inside the TP (expression 4.4).

$$AUTP = \frac{Area\ inside\ algorithm\ results\ inside\ TP}{Total\ area\ inside\ TP} \tag{4.4}$$

The AUTP is similar to the recall metric in a way that gives an overview of the TP results. However, it is focused more on the ”intensity” of results, on the points classified in positive class labels. Ideally, the whole area with positive labels had to be covered by probability = 1 to have the maximum value of AUTP. Adapting this concept to the present work, having a higher probability score (close to 1) from a year before the bear market will give a higher AUTP. In figure 4.3, there is an example wherefrom the probability results gave by a model, is possible to retrieve the important area for the AUTP calculation. All the probabilities in the green area will be divided by the sum of probabilities from the green and red areas. This metric will only consider positive labels the time spans of market pre-downfalls.

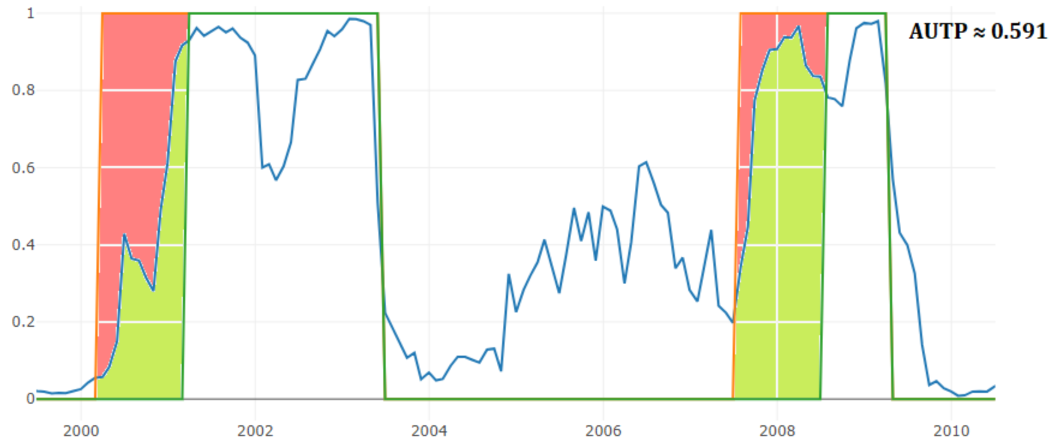


Figure 4.3: Display of algorithm's results and important areas for the AUTP result.

4.2 Case Studies and Respective Inputs

The tests made in the present work are based in three main case studies:

- **Case Study A** - Resort to Machine Learning algorithms - Logistic Regression, Random Forest, and XGBoost - in order to produce and train computational models that will be used to predict price decreases of at least 20% (Bear Markets) with six and twelve months of anticipation in the S&P 500 index.

- **Case Study B** - Resort to Machine Learning algorithms - Logistic Regression, Random Forest, and XGBoost - in order to produce and train computational models that will be used to predict, with six and twelve months of anticipation, price decreases in S&P 500 of at least: i) 17.5% in the first test case; ii) 15% in the second test case.

- **Case Study C** - Study focused on the XGBoost algorithm, where again, it will be produced and train computational models to predict market decreases of 20% and 17.5% with the difference on Hyperparameters' tuning for the XGBoost Algorithm.

Both case studies will make a similar process (explained in section 3.1), in order to obtain the results. However, choices of input had to be make by the user in the steps of data pre-processing, classification process, and validation of the results.

Empirically, having into account the best results in classifications and with the help of the metrics used and graphic representations, it was observed that the reduction in the number of economic variables led to better results in all algorithms. In the following case studies, the variables chosen as features are represented in table 4.1.

As it was already explained in the classification process, it will be implemented the sliding window method (cross-validation) in order to evaluate effectiveness and validity for the different models over time. All data since the year 1960 will be divided into 120 parts, making a new forecast every six months, approximately. It does not seem right to widen too much this window because this way, part of the testing set predictions would be based on training data quite distanced temporally. On the other hand, if the

Table 4.1: Final economic variables chosen to use in the several case studies of the present work.

Indicator name	Type	Description
CBOE Volatility Index	Monthly, Absolute value, Moving average of previous 10 months	Measures market expectation of near term volatility conveyed by stock index option prices
Conference Board Consumer confidence	Monthly, Variation value (Semester),	Reflects prevailing business conditions and likely developments for the months ahead. This monthly report details consumer attitudes and buying intentions, with data available by age, income, and region.
5% Pullback over prior 12 months	Monthly, Absolute value	Number of stock decreases (over 5%) in the preceding 12 months of the bull market peak
Job Claims	Monthly, Variation value (Semester)	U.S. Employment and Training Administration, Initial Claims
Misery Index	Monthly, Variation value (Quarter)	Unemployment rate + yearly percent change in Inflation.
Shiller P/E S&P 500	Monthly, Variation value (year)	Price earnings (P/E) ratio is based on average inflation-adjusted earnings from the previous 10 years.
Rule of 20	Monthly, Absolute value, Moving average of previous 5 months	Market P/E + yearly percent change in Inflation.

window is too small, it is more likely to find parts of the data with the wrong labels - negative outputs that only become positive as soon as the window goes through the exact moment of the downfall. In the table 4.2 is summarized the time space tested in the several case studies.

Table 4.2: Parameters set for the classification process, regarding time and data divisions

Parameters	Values
Study Period	29/01/1960 to 2019-03-29
Train Period	Begins at 29/01/1960
Test Period	Begins at 29/01/1970
Nº of Splits	120

It will also be tested the possibility of optimizing the algorithm's results through the optimization of the hyperparameters resorting to an implemented genetic algorithm (Case study C). As it was explained in the chapter background 2, in addition to the training and test subsamples of the data, in these sorts of cases will exist one more subset called validation subset. In order to obtain the validation subset, the training subset will be divided into two parts - 75% for training and 25% for validation in this work (figure 3.1).

This GA will have as genes 7 different possible hyperparameters for XGBoost: maximum tree depth for base learner (max_depth), Boosting learning rate (learningRate), Number of trees to fit (nEstimators), Minimum sum of instance weight(hessian) needed in a child (minChildWeight), Minimum loss reduction required to make a further partition on a leaf node of the tree (gammaValue), Subsample ratio of the training instance (subSample) and Subsample ratio of columns when constructing each tree (colSampleByTree) [76].

Initializing the population, the number of initial chromosomes was defined as 50, considering still a great value because having more different initial chromosomes will lead to major diversity, so the probability of converging to a not so good value is lower. In the training section, after testing all chromosomes with the fitness function, approximately 80% with the best results will be selected (40 chromosomes in this case). The not chosen chromosomes will be excluded from the population. To refill the population, new chromosomes will be created by crossing the previously chosen parents. This crossover will be made as a uniform process, combining an equal number of genes from each parent. Each of the newly created chromosomes, beyond the crossover, will add a new mutation for one of the parameters (randomly chosen). The maximum mutation step size will correspond to approximately 0.1**order of the magnitude of the maximum value* for this parameter. E.g. For a parameter where the maximum allowed value is 10, the step size will be approximately one. If the step size was too small, it would conduce to a smaller variance of the parameter in question, and the mutation could lead to less impact. The new population will be constituted by 75% of offspring and 25% of the parents (35 offspring chromosomes and 15 parents chromosomes, adapting to this work). The selection of the best parents will be made on the ones with greater fitness value (survival of the fittest). The population was restored, and the new generation is ready to repeat the whole process.

This process of training-selection-crossover-mutation is repeated the number of generations initially defined. The number of generations was set to 30 for the present work. This number is important since if it is small, it is unlikely to find the set of parameters that maximize the optimization. On the other hand, if the number of generations is too big, it can lead to unnecessary iterations and overfit for a small set of training datasets.

Table 4.3: Choices made on GA methods and parameters and for the XGBoost parameters values' intervals.

GA Part	Method and Values		XGB parameters	Values
Initial Population	50		**Learning Rate**	[0.01,1.00]
Generations	30		**Nº Estimators**	[10,1000]
Crossover	Unirform, 80% of the parents		**Depth**	[1,15]
Mutation	Random parameter, Step size ≈ 0.1*order of magnitude of maximum parameter value		**Child Weight**	[0,10]
Survivor Selection	75% Offspring, 25% Parents		**Gamma**	[0.1,10]
			Max subsample	[0.1,1]
			Max colSamplebyTree	[0.1,1]

The choices made for the GA methods and parameters and for the XGBoost parameter values' intervals (table 4.3) were based on recommendations from examples researched, literature read [76, 77], and empirically having in account the several experiences made with different values.

4.2.1 Case Study A: Predicting Bear Markets through the use of Machine Learning Algorithms

This case study will focus on trying to anticipate bear markets (market price decreases of at least 20%) in the S&P 500 index. It was obtained results trying to forecast market downfall events with six and twelve months of anticipation with three different algorithms - logistic regression, random forest, XGBoost -, plus for the average (ensemble approach) of all together. In total, eight different results were collected and will be analyzed separately hereinafter. These models will be produced and trained according to the values of the economic variables already shown in table 4.1. In the methodology section, it was already explained the whole process of classification.

The output will detain the positive and negative labels. The labels considered positive for the different case studies will be constituted not only by the whole downfall events zones but also by the periods before the event happens, with the size of the lag different depending on the respective test case (six or twelve months). However, the most important scenario is to have the time spans before the bear market period, "filled" with high probabilities, which will indicate that the downfall event is being anticipated. That said, although all metrics' results remain important for future model comparisons, there are results fetched from some metrics more important and enlightening for the present work.

There will be two different results for the recall metric, one in which the downfall zones do not count as a positive class label and other where it counts. This decision was made to take into account two types of TP results: success only in the pre-downfall zones and success in both pre-downfall zones plus the downfall itself period until it is over. The AUTP metric only will count the pre-downfalls zones as positive. Being the most important result, to have the time spans before the bear market "filled" with high probabilities, and the metrics recall only counting pre-downfall zones, plus the AUTP, will enable to have a more insightful analysis on these "pre-downfall" zones' results. The remaining metrics (accuracy, precision, and AUC) count both pre-downfalls and downfalls zones as positive class labels for its calculation.

Two confusion matrix with different thresholds (0.5 and 0.35) were made to consider results with probabilities that, although not reaching 0.5, remain relatively high and still make sense to open a new perspective, considering TPs. The values for the different metrics obtained in this case study are presented in table 4.4.

The accuracy and precision will decrease with a lower threshold because there is a greater increase in FP than in TP. The increase of FP is not always seen as a bad scenario because it often coincides with a significant stock market price decrease that simply just did not drop enough to be classified as a positive class label in this specific case study. It also might correlate with economic instability, a subject that will be treated in the final review of this case study. However, although the accuracy and precision metrics consider the FP for its results, they are still important to ascertain greater volatility in the results. Although sometimes not differing much from the other algorithms, the Ensemble approach obtained most of the best results in both accuracy and precision metrics for tests with six months anticipation training, and XGBoost for tests with twelve months of anticipation training.

Table 4.4: Metrics' values for case study A.

Metric:	Accuracy		Precision		Recall (w/ falls)	Recall (w/o falls)		AUTP	AUC
Model **Threshold:**	0.5	0.35	0.5	0.35	0.5	0.5	0.35		
6 months									
Logistic Regression	0.8695	0.8542	0.6012	0.5673	**0.8716**	**0.7188**	**0.7813**	**0.6167**	0.8813
Random Forest	0.8616	0.8105	0.6138	0.4929	0.6817	0.2875	0.5219	0.3914	0.8781
XGBoost	0.8644	0.8542	0.6198	**0.5839**	0.6881	0.3750	0.4688	0.3703	0.8802
Ensemble (LR, RF, XGB)	**0.8836**	**0.8546**	**0.6333**	0.5628	0.8046	0.5438	0.6427	0.4595	**0.8969**
12 months									
Logistic Regression	0.8068	0.7102	0.5635	0.4416	**0.7986**	**0.6129**	**0.7581**	**0.6377**	**0.8721**
Random Forest	0.8076	0.7304	0.5972	0.4548	0.5705	0.2806	0.4758	0.3458	0.8000
XGBoost	**0.8372**	**0.8164**	**0.6667**	**0.6026**	0.6187	0.3065	0.4194	0.3300	0.8372
Ensemble (LR, RF, XGB)	0.8336	0.8074	0.6333	0.5628	0.6978	0.4185	0.6427	0.4379	0.8680

The AUC metric, although also considers the FP for its calculation, has in favor that measures the quality of predictions for positive and negative results, independent of the threshold used. The logistic regression and the Ensemble models also dispute the best value in the AUC metric.

The success in anticipating the event will be more reflected in the recall (TP rate) and the AUTP because these metrics do not have into account the FP results for its calculations. In this case study, it is possible to observe that the logistic regression obtained the best results in both recall and AUTP metrics.

The more balanced results were the ones from the Ensemble approach, obtaining most of the second-best results in all metrics.

For a more intuitive analyze it was chosen to plot the results in a graphic representation with the bear market's time spans marked, the lag imposed on the respective test marked (six or twelve months in pre-downfall), and the probabilities gave by the algorithm for each instant tested.

The results for the logistic regression algorithm (figures 4.4 and 4.5) are apparently good, presenting high probabilities of bear markets, not only inside the bear market periods, but also in the time spans before. This explains the relatively high results obtained in the recall and AUTP due to the high TP classifications. Although almost in all negative labeled zones, the algorithm shows low probabilities, there are still some peaks classified as FP. However, these peaks, as already stated, are not always undesirable or false alarms since some of them coincide with other significant price decreases in the S&P 500 or with time spans right before the exact six/twelve months of lag applied. It is also visible that the test for twelve months of anticipation could still detect the bear market of 1987 with some anticipation while that can not be said for the six months test. That scenario can be explained by the greater amount of data used in training the twelve months anticipation test case.

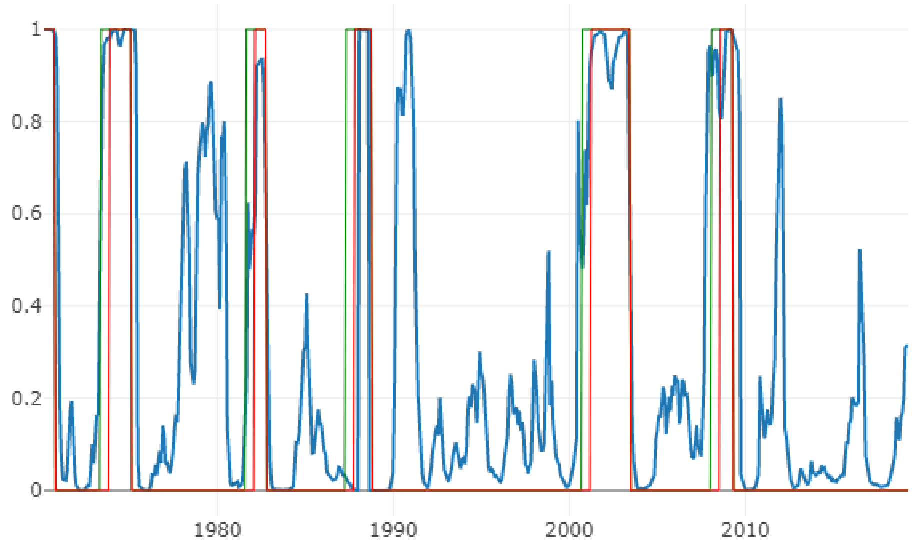


Figure 4.4: Time-series out-of-sample tests to anticipate bear markets (-20%) in S&P 500 with 6 months of advance, from 1970 to 2019. Model: Logistic Regression.

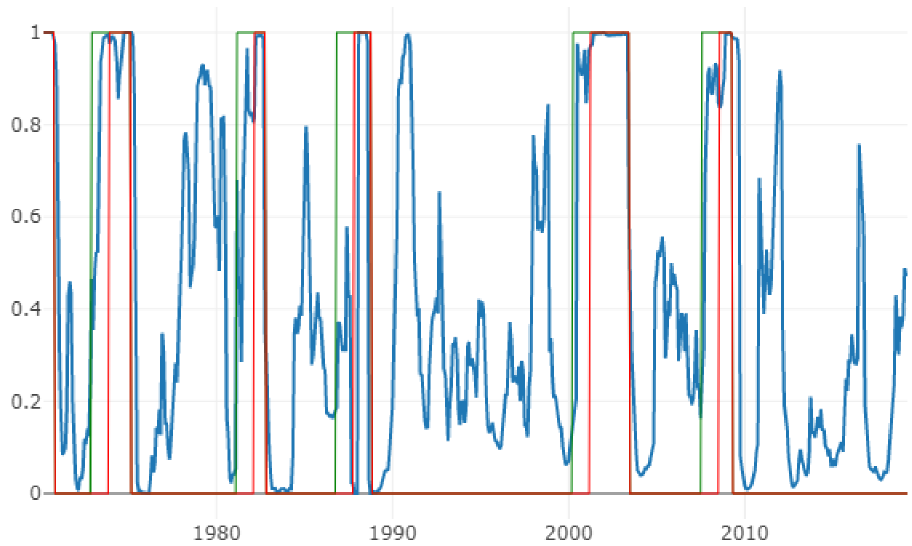


Figure 4.5: Time-series out-of-sample tests to anticipate bear markets (-20%) in S&P 500 with 12 months of advance, from 1970 to 2019. Model: Logistic Regression.

In the XGBoost results (figures 4.6 and 4.7), it is possible to observe less and smaller variations of the results compared with the logistic regression model, especially in negative labeled zones. Although it gives some worse results in the positive labeled areas comparing to the logistic regression model, these smaller variations in the negative labeled areas explain the higher results in the accuracy and precision metrics. Almost all downfall events were detected with some anticipation, except the one in 2001 in both tests and in 1987 for the six months of anticipation test.

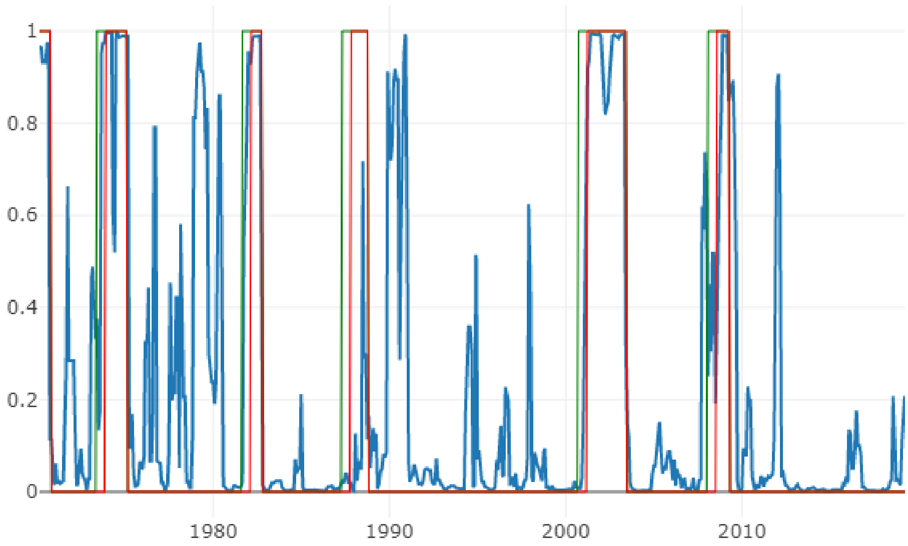


Figure 4.6: Time-series out-of-sample tests to anticipate bear markets (-20%) in S&P 500 with 6 months of advance, from 1970 to 2019. Model: XGBoost .

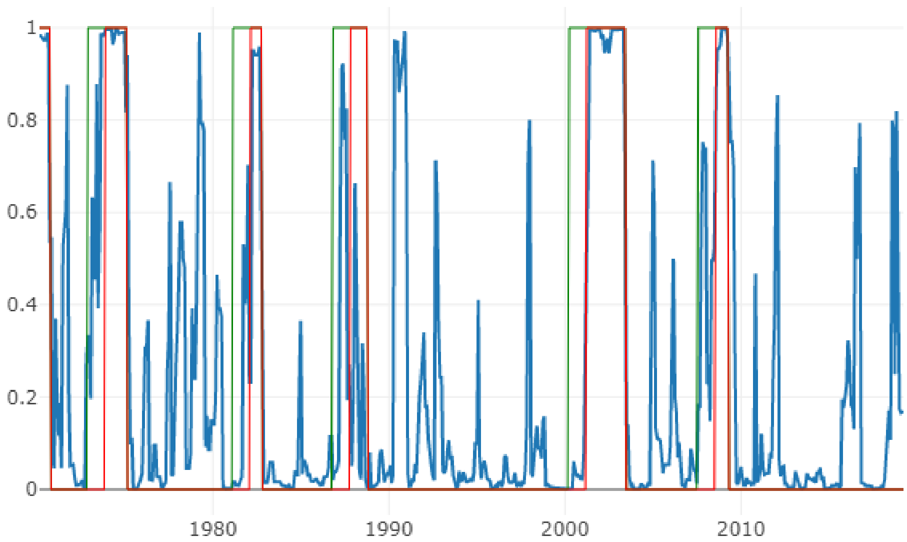


Figure 4.7: Time-series out-of-sample tests to anticipate bear markets (-20%) in S&P 500 with 12 months of advance, from 1970 to 2019. Model: XGBoost.

Observing the results for the Ensemble of the three algorithms (figures 4.8 and 4.9), it is possible to verify the failure by all algorithms to detect the bear market of 1987 with six months of anticipation in training. Nevertheless, all the other downfalls were caught with values of probability above 0,4 anticipating the bear market zones, and due to the XGBoost lack of FP, it "cleans" the negative labeled zones, having less FP than the logistic regression model. This was demonstrated by the metrics where this approach had the second-best results in almost all metric fields, combining this way the best results of each model.

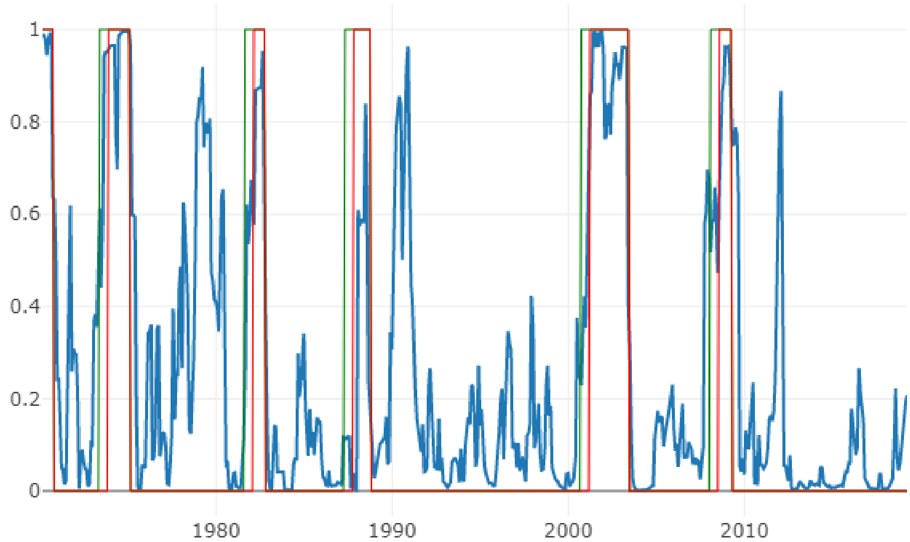


Figure 4.8: Time-series out-of-sample tests to anticipate bear markets (-20%) in S&P 500 with 6 months of advance, from 1970 to 2019. Model: Ensemble Approach.

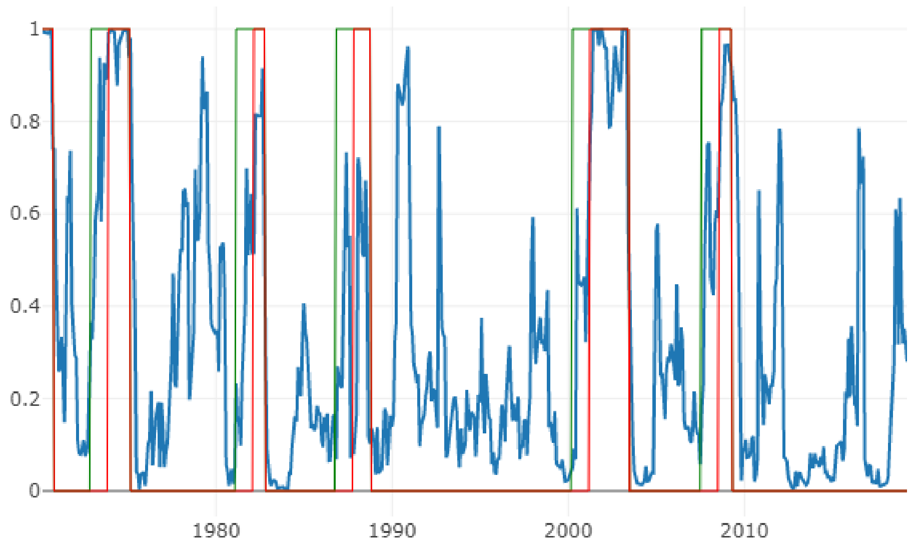


Figure 4.9: Time-series out-of-sample tests to anticipate bear markets (-20%) in S&P 500 with 12 months of advance, from 1970 to 2019. Model: Ensemble Approach.

It is also curious to observe in the bear market of 2001-2003, the decay of the probability results (around the middle of 2002) in some of the cases tested, with a subsequent increase soon after (figure 4.8 shows an example of this). The market quick variations can explain the result volatility. Since the year 2002, where there were variations of the index price that could indicate the end of the bear market era, with some increases around 20% (figure 4.10). However, these climbs were all followed by even greater falls, demonstrating that the market was still on a bearish trend. This section had a duration of almost two years and a half, since the initial mark with the 20% fall.

Observing the features' importance for different years, it was visible that four variables stood out from the others. The Conference Board Consumer Confidence, the number of 5% pullback before the bull peak, the Rule of 20 and Shiller P/E were the variables most taken in consideration when training the models. The Consumer Confidence variable is an indicator that reflects prevailing business conditions

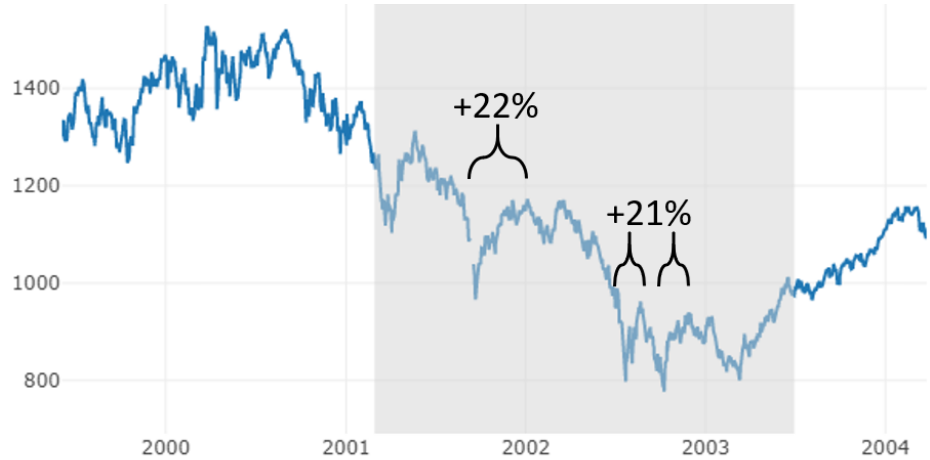


Figure 4.10: S&P 500 several variations in the bear market time-span of 2001-2003.

and likely developments for the months ahead [78], and that may impact the investors' overview and sentiment of the actual economic conditions and consequently impact the market's movements. The Shiller P/E and Rule of 20 are variables of stock valuation category, and their significance in model training might mean that the over/sub valuation of stocks influences the market's variations as well. Nonetheless, empirically was observed that taking out the remaining features of the tests led to a negative impact in the results. In figure 4.11 is presented the feature importance for the models trained right after two different bear markets: 1987 and 2008.

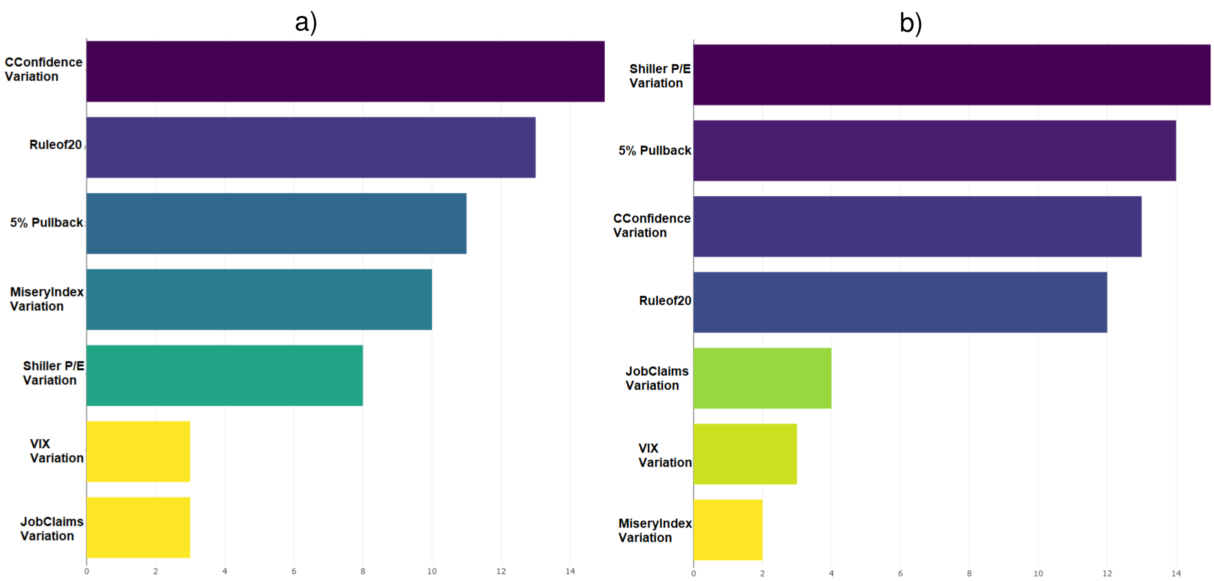


Figure 4.11: a) Average Feature Importance of the three models trained (logistic regression, random forest and XGBoost) to predict bear markets with twelve months of anticipation. In the time-series instant: 29-01-1988 (after 1987 bear market). b) Average Feature Importance of the three models trained (logistic regression, random forest and XGBoost) to predict bear markets with twelve months of anticipation. In the time-series instant: 30-04-2009 (after 2008 bear market).

Giving a general analysis of all tests of this case study, it is still possible to verify FP results in the following dates (Considering as Threshold=0.4): ≈ 1978, ≈ 1984, ≈ 1990, ≈ 1994, ≈ 1998, ≈ 2004, ≈

2011, $\approx$ 2016 and $\approx$ 2018. Training the data having the positive labels twelve months before the bear market zone, it is possible to observe by the graphic representation of the results that beyond the fact that the event's anticipation detection is reached earlier, it also get better results in the detection of 1987 and 2001 bear markets, comparing to the tests for six months of anticipation. The use of the Ensemble approach can distinguish better between the positive labeled zones and the FP comparing to the logistic regression model. However, the logistic regression presented the best results in the positive labeled zones, with greater values and more antecedence when anticipating.

4.2.2 Case Study B: Predicting Downfalls of 17.5% and 15% through the use of Machine Learning Algorithms

Resorting to models produced by Machine Learning algorithms as in the Case Study A, this case study will focus on two different sub-test cases:

1. Trying to anticipate market price decreases of at least 17,5% in the S&P 500 index with six and twelve months in advance;

2. Trying to anticipate market price decreases of at least 15% in the S&P 500 index with six and twelve months in advance.

The only difference within the whole process to the Case Study A will be on the output of the data set used. In addiction to the time spans marked as at least price -20% decreases, the data set will add as well the price decreases of at least 17,5% as positive labels in the first test case, and in the second test case it will be added the price decreases of at least 15% as well. Adding more positive labels to the output can impact the training of the models by the algorithms.

Some of the output new positive labels added is this case study match with bear markets from the previous case study, with the difference that is most often observed some months before. Consequently, in these cases the anticipation will try to be verified a few times earlier relatively to the tests in the previous case study. In the figure 4.12 is displayed an example for the data labels' differences between both case studies.

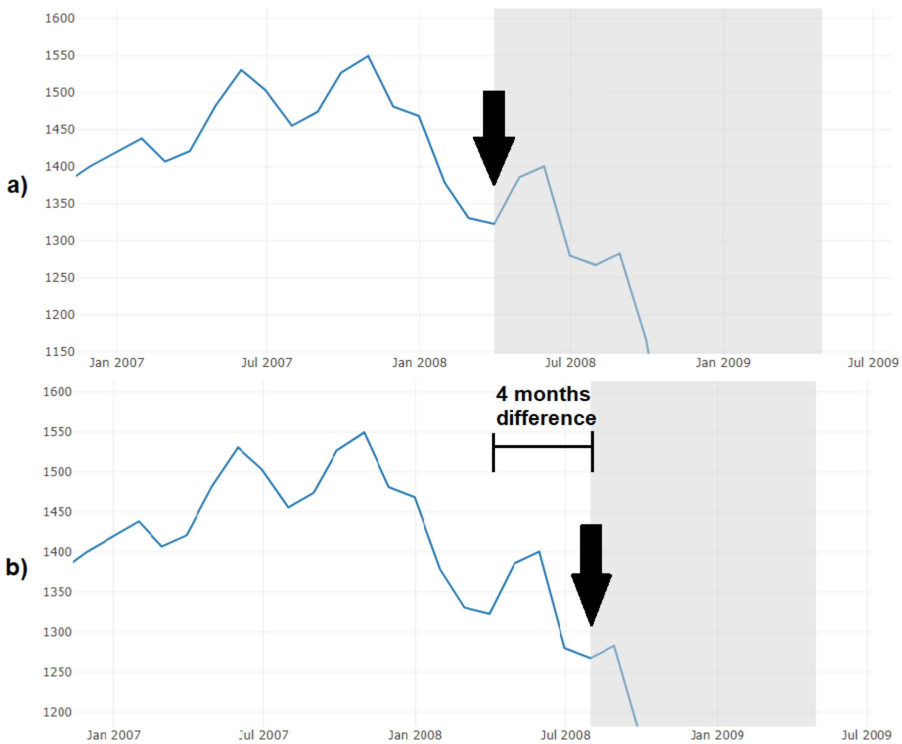


Figure 4.12: This figure exemplies the dataset labels' differences in the labels for the different case studies. a) Time span of S&P 500 price, with beginning of 20% price decrease marked. b) Time span of S&P 500 price, with beginning of 17.5% price decrease marked.

As proceeded in the previous case study, the same metrics were used as analytic evaluation criteria for all the four models produced, and the results are displayed in tables 4.5 and 4.6.

Table 4.5: Metrics results for market downfall events of at least 18%.

Metric:	Accuracy		Precision		Recall (w/ falls)	Recall (w/o falls)		AUTP	AUC
Model **Threshold:**	0.5	0.35	0.5	0.35	0.5	0.5	0.35		
6 months									
Logistic Regression	0.8407	0.8034	0.7337	0.6461	**0.7807**	**0.4516**	**0.5806**	**0.5189**	0.8835
Random Forest	0.8277	0.7886	0.7708	0.6391	0.6508	0.2734	0.4565	0.3514	0.8570
XGBoost	0.8305	**0.8186**	0.7702	**0.7151**	0.6631	0.3000	0.3548	0.3411	0.8639
Ensemble (LR, RF, XGB)	**0.8564**	0.8175	**0.7957**	0.6829	0.7358	0.3758	0.4637	0.4038	**0.8896**
12 months									
Logistic Regression	**0.7492**	**0.6983**	0.6749	0.6030	**0.7733**	**0.5691**	**0.6667**	**0.5453**	**0.8313**
Random Forest	0.7157	0.6453	**0.7030**	0.5637	0.5583	0.2329	0.4175	0.3363	0.7354
XGBoost	0.7169	0.6820	0.6961	**0.6082**	0.5749	0.2439	0.3171	0.2853	0.7317
Ensemble (LR, RF, XGB)	0.7204	0.6722	0.6876	0.5870	0.6093	0.3000	0.4837	0.3889	0.7883

Table 4.6: Metrics results for market downfall events of at least 15%.

Metric:	Accuracy		Precision		Recall (w/ falls)	Recall (w/o falls)		AUTP	AUC
Model **Threshold:**	0.5	0.35	0.5	0.35	0.5	0.5	0.35		
6 months									
Logistic Regression	0.8220	0.7814	**0.7658**	0.6572	**0.7658**	**0.4865**	**0.6351**	**0.5028**	0.8582
Random Forest	0.7817	0.7551	0.7511	0.6413	0.6287	0.2534	0.4878	0.3745	0.8431
XGBoost	0.7840	0.7814	0.7594	**0.7061**	0.6368	0.2838	0.3784	0.3292	0.8429
Ensemble (LR, RF, XGB)	**0.8231**	**0.7864**	0.7599	0.6943	0.7117	0.3899	0.5392	0.4029	**0.8711**
12 months									
Logistic Regression	**0.6949**	**0.6661**	0.6677	**0.6195**	**0.7333**	**0.5109**	**0.6350**	**0.5545**	**0.8022**
Random Forest	0.6611	0.6187	**0.6887**	0.5893	0.5344	0.2307	0.4672	0.3540	0.7091
XGBoost	0.6429	0.6241	0.6610	0.6264	0.5474	0.2336	0.2847	0.2787	0.7167
Ensemble (LR, RF, XGB)	0.6783	0.6474	0.6842	0.6161	0.6161	0.3361	0.5419	0.3837	0.7605

At first impressions, it is possible to observe that all results on accuracy had decrease, which was expected due to the increase of positive labels. Not only the number of market downfall events is higher but also because for the same downfall event, the anticipation in some cases will have to be made earlier comparing to the previous case study (as observed in figure 4.12), to have as much TP results as possible.

Although there is a similarity between the values of the different algorithms for accuracy and precision, in the tests for 17.5% and 15% downfall events with six months anticipation, the tendency remains for XGBoost and the Ensemble approach to have better results in these fields. However for these metrics in the 17.5% price decrease test case with twelve months of advance, the best results are spread for the several algorithms and in the 15% test case with twelve months of anticipation the logistic regression takes almost all the best results in all metrics, differing this way to the results obtained in the previous case study.

On the other hand, logistic regression continues to have the best results in metrics which the true positive rate has a bigger weight (recall and AUTP). The AUC metric, as in the previous case study, has the best results divided between the logistic regression and the Ensemble approach.

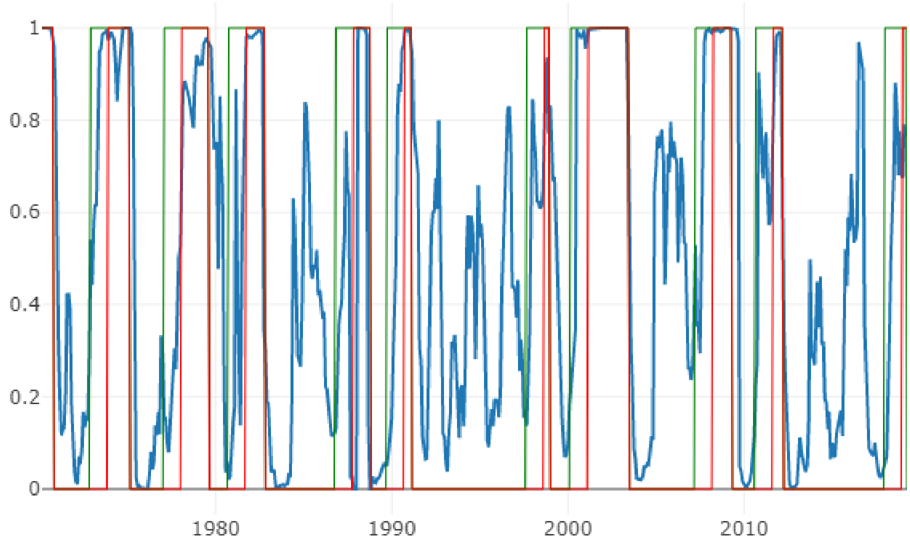


Figure 4.13: Time-series out-of-sample tests to anticipate -17.5% decreases in S&P 500 with 12 months of advance, from 1970 to 2019. Model: Logistic Regression.

Based on this graphic representation displaying logistic regression's results (figure 4.13), it is possible to observe that most of the positive results (TP and FP) reach up to 70%. That explains not only the higher values in metrics such as recall and AUTP but also the lower results in accuracy and precision for both confusion matrix (Thresholds at 50% and 35%), comparing to the case study A. However, all the downfall events were caught by this model, although some events having small probabilities and anticipation detecting. The downfall of 1978 was only caught with 0.35 of probability, and already really close to the downfall event.

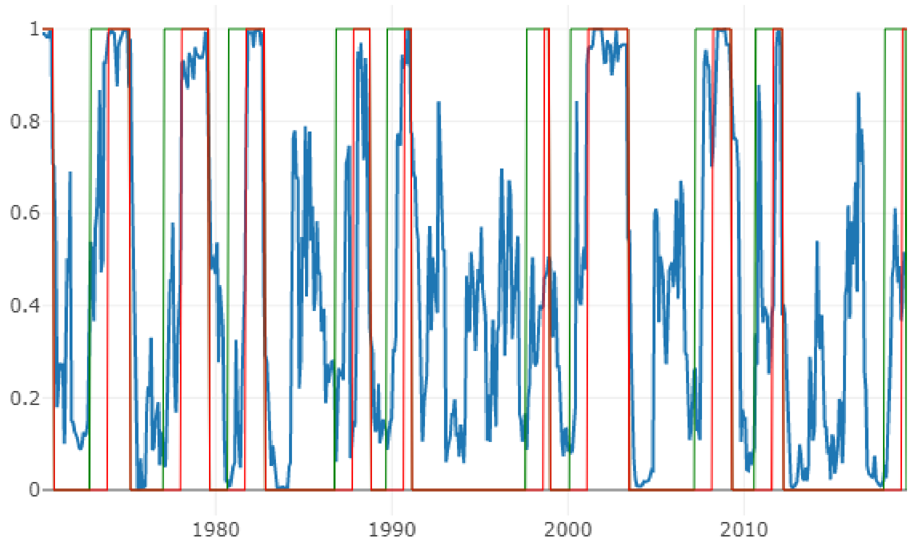


Figure 4.14: Time-series out-of-sample tests to anticipate -17.5% decreases in S&P 500 with 12 months of advance, from 1970 to 2019. Model: Ensemble Approach.

As observed in the figure 4.14 for the tests trying to anticipate 17.5% market's downfalls, it is possible to verify that with the exception of the 1982 market downfall event, almost all the others were caught by the Ensemble approach, although with some lower probabilities and less anticipation then in the logistic regression model. That demonstrates the metrics results where logistic regression algorithm had the best results (recall and AUTP). However, the amplification of the results in negative labeled zones is also verified for the logistic regression model and that volatility in the results explains the higher values in precision and accuracy by the Ensemble approach for twelve months training.

Regarding the feature importance for the -17.5% market price decrease, it was denoted that another economic variable was also taken into more consideration in models' training, comparing to the previous case study: Misery Index. This variable takes the unemployment rate for its calculation which could be more correlated with the country economy state. The more importance of this feature might be related to the aggregation of more types of downfall events in these test cases, and some of these downfalls often coincide with periods of economy recession - 1990 bear market for example. The figure 4.15 shows the average feature importance of the three different models in instants after the 2001 big market decrease.

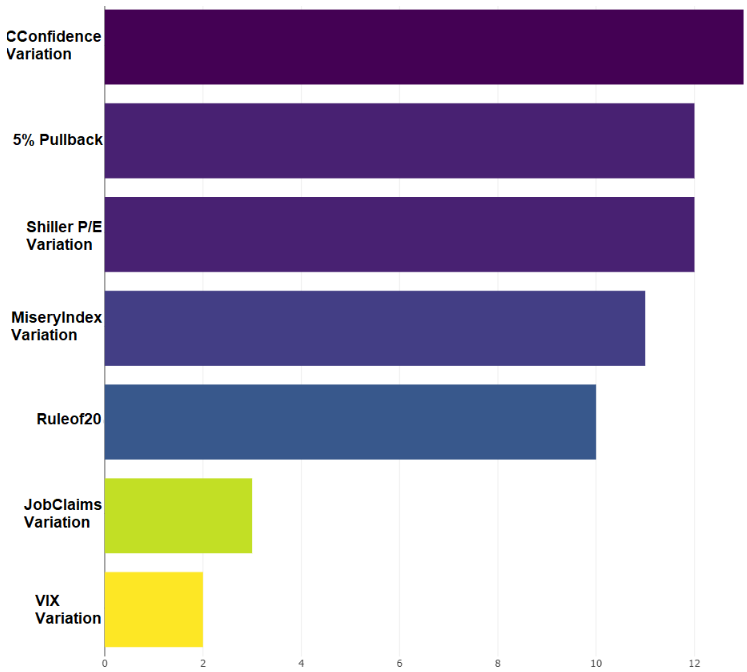


Figure 4.15: Average Feature Importance of the three models trained (logistic regression, random forest and XGBoost) to predict -18% market decreases with 12 months of anticipation. In the time-series instant: 30-02-2002 (after 1987 bear market).

Some of the FP results in the previous case study are now correspondent to >17.5% price decreases. However, it is still possible to verify some FP results in the following dates (Considering as Threshold=0.4): ≈1984, Through the middle of 90s (≈1994-1996), ≈2004 and ≈2016.

Analyzing the figure 4.16 with the results for the Ensemble approach in tests for the -15% market price decreases, is possible to verify that does not differentiate from the test case of 17.5% decreases (figure 4.14). Both are very similar, with an increase in the absolute values of all results in the test case of 15% price decrease. That can be positive in a sense that the TP results are almost certain in some cases (Probability of an event happening = 1), however, a negative point about this is that the FPs also have higher absolute values, and there is almost no distinction between positive and negative zones. It

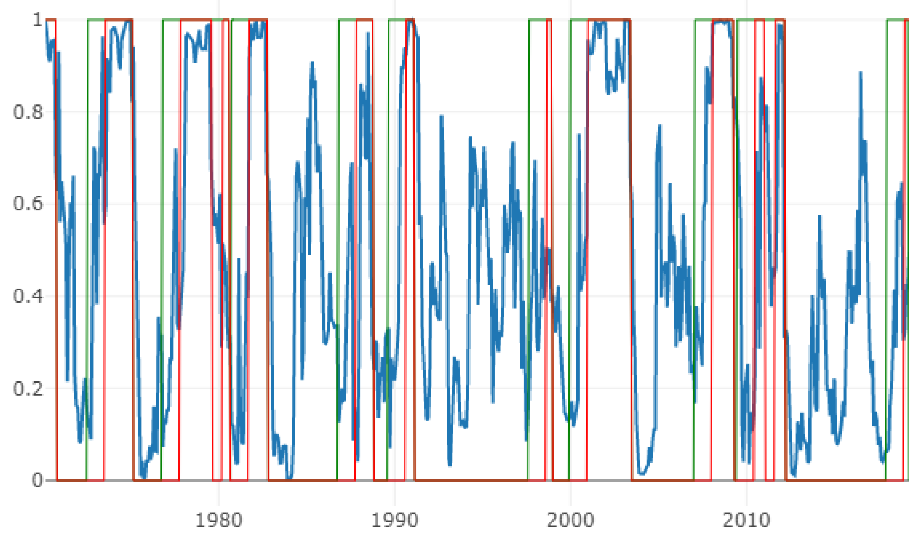


Figure 4.16: Time-series out-of-sample tests to anticipate -15% decreases in S&P 500 with 12 months of advance, from 1970 to 2019. Model: Ensemble Approach.

is possible to acknowledge the increasing of the volatility for all algorithms, comparing to the results from the case study A. Regarding FP results, some of them in the previous test cases (-20% and -17.5% price decreases) are now correspondent to >15% falls. However, it is still possible to verify some FP results in the following dates (Considering as Threshold=0.4): ≈1984, Through the middle of 90s (≈1994 and ≈1996), ≈2004 and ≈2016.

4.2.3 Final Review of Case studies A and B

Distinguish Bear Markets (Case study A) with other stock market's times, did not reveal to be an easy task and one of the reasons why is because the time spans where Bear Markets happen, are not the only ones where are unusual feature variations in the market. For instance, there were the recessions of 1981 and 1990 corresponding to market price falls of $\approx$15% and 19.9%, i.e., not considered bear markets. When training the data in chronological sequence, in these periods of recession there may be similar variations of instability as in bear market labeled zones (positive labels), however the model will label these instants as a negative output and consequently will make it difficult to anticipate further falls with similar variations.

In this chapter it will be reviewed the right and wrong results obtained by the algorithms (TP, TN, FP and FN) in the previous case studies.

Starting with the FPs results, in the case study A it is possible to observe that these type of result often occur by the years of 1979, 1984, 1990, 1994, 1998, 2004, 2010, 2011, 2016 and 2018.

The FPs of 1979, 1990, 1998, 2010, 2011 and 2018 are then seen as positive labels in the case study B, because they are all classified as price decreases above 15%. Owing to this fact, it is natural that the models find similarities in the variations of these time frames where are also big market downfall events, with those of bear market. These FPs peaks are not necessarily wrong or undesirable since they still coincide with significant declines in the S&P 500 index. So it was proven the ability of the models to catch smoother, but still significant downfall events in the market. However there are still the FP peaks of 1984, 1994, 2004 and 2016 that are not positive labels in the case study B but still coincide with events which led to market's instability, as will be explained hereupon.

In the year of 1994 the US economy was finally starting to recover from a big downturn (recession of 1990) and naturally with the economy booming and interest rates pinned at low levels, the stock market was on a high trend. Nonetheless, there were concerns regarding the inflation increase caused by the booming economy and this would lead to Federal Reserve System (FED) raising interest rates throughout the year of 1994. The stocks, meanwhile, had a volatile year [79]. In the case study B is also possible to observe FP results by the year of 1996, which in part coincide with some economic instability related with the ones from 1994 now explained. In 1994 Mexico reached agreements in negotiations for a foreign trade agreement with the United States later known as the North American Free Trade Agreement (NAFTA). Still in 1994 the FED raised its policy rate (b) in figure 4.17). The result was a strong decline in portfolio investment and in 1995, Mexico underwent the worst recession, destabilizing financial markets. Fearing that this crisis could impact the economy itself and others worldwide, the US joined with Internacional Monetary Fund (IMF) and had an intervention with a large scale standby loan for Mexico. This kind of international bailout, sparked much criticism saying that created moral hazard and is as well one of the reasons for the economy instability by that time [80].

"When will the FED start tightening monetary policy?" is a question that has been a concern for investors throughout the years. Due to correlation with markets, theories have come up saying that this may have some impact on stock variation [81]. Besides the increase in Federal Funds Effective Rate

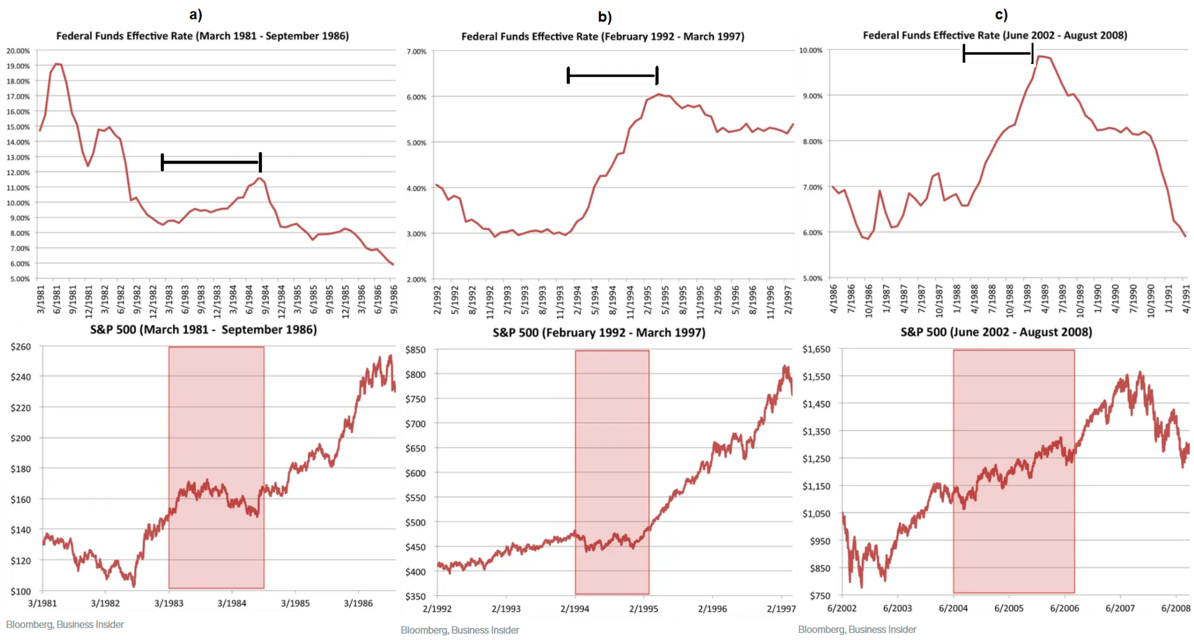


Figure 4.17: 3 examples where the Fed Funds Effective Rate presented correlation with market trends' stops. a) 1984-1986; b) 1994-1996; c) 2004-2006 [81].

which contributed 1994-1996 instability already stated, there were also other times where FED's rate had its influence. In 1983-1984 there was a market correction which coincide with the rate's increase [81] (a) in figure 4.17). By the year of 2004 was denoted an increase in the FED's rate until July 2006, as concerns over a brewing housing bubble mounted. Following the June 2004 announcement of this increase, stocks sold off – but only until August, when they began a rally toward new all-time highs (c) in figure 4.17) [81]. 2004 was also a volatile year regarding the Conference Board Consumer Confidence index, which as was demonstrated in the results, was one of the indicators with bigger weight when training models throughout the time-series.

In 2016, a set of significant financial and political events elevated the instability [82, 83]:

- Political events such as the BREXIT referendum;
- The election of the 45th president of the US, Donald J. Trump;
- The ending of the Iran sanctions created ripples in the world's economies;
- and the chinese market crash and slowdown of the its economy in the beginning of the year.

Now giving an overview of the FN results, in the case study A in test cases with six months of lag added, there are FN results mostly in the 1987 bear market also known as Black Monday.

On October 19, 1987 the S&P 500 Price Index dropped to 224.84, down -33.24% from its prior high on August 25, 1987. That day the S&P 500 dropped -20.47%, which corresponds to the largest stock market decline ever in one day [84]. Faster downfalls are logically expected to difficult the respective predictions. However there are a lot of facts pointed as reasons for this price decrease in the market which are not easy to find out with economic variables: In the week before there were escalations of the confrontations between the US and Iran so the global events were tense [84]; the causal relationships among national stock markets around the October 1987 stock market crash [85]; the failure of stock markets and derivatives markets (options and futures) to operate in sync [86]; and also many analysts agreed that stock prices were overvalued in September, 1987 [87]. In the figure 4.18 is possible to observe two of the economic variables that are calculated using stock's PE, and is clear to observe an increase of both variables before 1987.

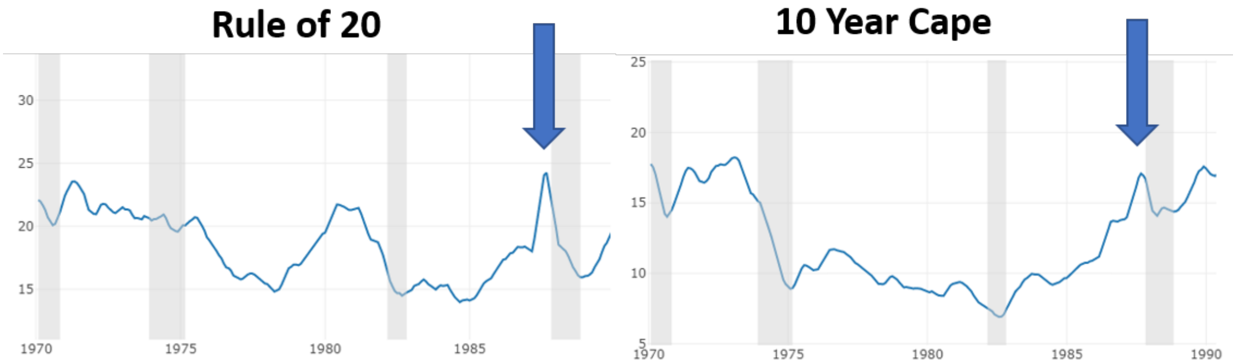


Figure 4.18: Graphic representation of Rule of 20 and 10 year Cape: Two economic variables that use P/E ratios for its calculations. Bear markets time spans in shade.

Even with data from these variables, the downfall event was not anticipated by all tests, and it might also be due to the lack of data before this date, to train the models on.

The major differences between case studies A (Bear Markets) and B (18% and 15% downfalls) were denoted on a bigger volatility and amplification in the results of case study B. The increase in the positive labels of downfall events can be one explanation for these facts. Having more downfall scenarios (almost the double as in case study A), which can be different in several aspects, can make it easier to find a new economic conjuncture in the time-series which can relate to previous trained events by the algorithms. This had also a positive effect when predicting some of the downfall events (2001 and 2008 for example), reaching higher probabilities of downfall sooner than in the case study A. The disadvantage of this increase in absolute values in case study will be in decreasing the differences between TP and FP, making hard to distinguish both types of results.

4.2.3.1 Discussion

Comparing to the state of art, the works of Barbosa [42], and Liu and Moench [57] as was already stated used similar methodologies for evaluation criteria having both results for out-of-sample predictions with six and twelve months of advance. Instead of predicting bear markets and other big stock price decreases, these works are focused on the prediction of US economy recessions.

In the work by Liu and Moench [57] the best results for the probit models with out-of-sample tests, in AUC metric are around 0.9. In the present work, the Ensemble approach had around 0.9 in out-of-sample data tests for -20% and -18% downfalls. However, the AUC metric has a component that depends on the false results (FP and FN), so having a data set mainly labeled as negative, the AUC metric does not gives the best insight regarding the TP results. Nonetheless, analysing the graphic representation in [57] it is possible to notice that principally for more recent recessions (1990, 2000 and

2008), the model's results do not reach 0.4 of probability results, anticipating the event. This is a point in favor of this work where in both case study A to detect bear markets and case study B where one has to detect price decreases above 15%, several results given by models in pre-downfall zones were above 0.4 and some even above 0.5.

In the work by Barbosa [42] is possible to make a more complete analytic comparison because the same criteria evaluation regarding metrics and graphic representation. However, because the same Machine Learning models are used, the results obtained in both works should be similar.

Firstly, regarding the TP results, the recall metric will be used for comparisons. In [42] in tests for six and twelve months anticipation of events, the best results for the recall are 0.79 and 0.83 respectively by the random forest model, while in this work are 0.87 and 0.80 by the logistic regression model anticipating -20% decreases. Although very similar results between works, it can not be despised that in this work around more 15 years of time-series data is tested - 1985-2018 in [42] and 1970-2019 in the present work - and therefore although the best results for the recall metric are split between both work, with more test date the number of TP to hit increases and there is a greater tendency to lower the recall metric values. Comparing the AUC metric, the best values are split between both works but all values are still quite similar - values between 0.87 and 0.90.

It is fair to state that economy and stock markets are not the same. Stocks trade on sentiment, corporate earnings, and trends. The economy is a collection of goods and services produced within a specific period of time. NBER defines economy recessions as “a significant decline in economic activity spread across the economy, lasting more than a few months, normally visible in real GDP, real income, employment, industrial production, and wholesale-retail sales” [88]. All these indexes are directly measured through indicators, and if one or more starts to fall, it is a good sign that a recession could be coming. Generally speaking, a recession tends to correspond to a bear market and there are evidences of correlation between GDP and the stock market [89], however, there were times in history where this correlation did not occur, and one or few external factors greatly influenced the stock market variations. An example of this situation are the case of the market volatility caused by FED Funds Effective Rate increase, exposed above in this section (figure 4.17). Concluding, anticipate bear markets could be possible as it is demonstrated by this work results. However, due to the stock sentiment component leading to bigger volatility, it makes it harder, comparing to economy recessions, to get the exact intensity with which one or more indicators can impact the stock market.

4.2.4 Case Study C: Tuning of XGBoost hyperparameters using a Genetic Algorithm

This case study will focus on the XGBoost algorithm's results improvement, through the use of a Genetic Algorithm (GA).

One of the major questions in this case study will be in deciding which fitness function is more appropriate to evaluate and select the parent chromosomes that will be used for posterior crossover and create the following generation. It was made several tests for the same test case, with recall (Threshold=0.35) and the AUC metric, to make an analytic comparison. The criteria to choose which fitness functions should be used for tests, was based on the importance of each metric regarding the goal in this work. Also it was tested other fitness functions and did not obtain as good results as the recall and AUC, and therefore were excluded of this case study.

In the table 4.7 as it can be seen, the not-optimized version of XGBoost got the best values in accuracy, precision and AUC. Although the values are not too much higher than the ones of the other versions of XGBoost, this can be explained because these metrics take into account the values of FP in his calculation, that sometimes in this works are not always undesirable. It is possible to check greater values of recall (without counting the "bear periods") and AUTP metrics, in all of the optimized versions of XGBoost. This is a good sign of optimization since as it was already explained, these metrics give the best insight regarding TP results and for what is aimed in this work which is the anticipation of big downfalls in the S&P 500.

Table 4.7: Value of metrics testing different versions of the XGBoost algorithm, with tests for prediction of bear markets with 12 months in advance.

	Metric:	Accuracy		Precision		Recall (w/ falls)	Recall (w/o falls)		AUTP	AUC
Model - 20% downfalls, 12 months lag	**Threshold:**	0.5	0.35	0.5	0.35	0.5	0.5	0.35		
Default										
XGBoost		**0.8372**	**0.8164**	0.6667	**0.6026**	**0.6187**	0.3065	0.4194	0.3300	**0.8372**
Optimized, Fitness = Recall, TH=0.35										
XGBoost		0.8129	0.7659	0.6048	0.5040	0.6000	**0.3573**	0.5500	**0.4031**	0.8333
Optimized, Fitness = Auc										
XGBoost		0.8342	0.7600	**0.6679**	0.4957	0.5899	0.3290	**0.5581**	0.3883	0.8128
Impact on Ensemble										
Default XGBoost										
Ensemble (LR, RF, XGB)		0.8336	**0.8074**	0.6333	**0.5628**	0.6978	0.4185	0.6427	0.4379	0.8680
Optimized, Fitness = Recall, TH=0.35										
Ensemble (LR, RF, XGB) w/ Optimized XGBoost		0.8347	0.7753	0.6328	0.5144	**0.7115**	**0.4565**	0.6855	**0.4554**	**0.8694**
Optimized, Fitness = Auc										
Ensemble (LR, RF, XGB) w/ Optimized XGBoost		**0.8356**	0.7708	**0.6399**	0.5084	0.6950	0.4526	**0.6968**	0.4524	0.8662

The graphic representations below show the results only for the XGBoost algorithm optimized with recall metric as fitness function), and also the real impact in the Ensemble approach with the three algorithms (figures 4.19 and 4.20). As it is possible to observe by the graphic results, the optimized hyperparameters version obtained greater results of probability in TP zones and could antecede with more advance the market downfalls of 20% in the events of 1987 and 2000, and this leads to a positive impact on the Ensemble approach. Regarding the other 20% downfalls and the FP results, both versions obtained similar results, however the optimized hyperparameters version has a own FP right before 1990, which match with the US recession and a significant market price decrease as well.

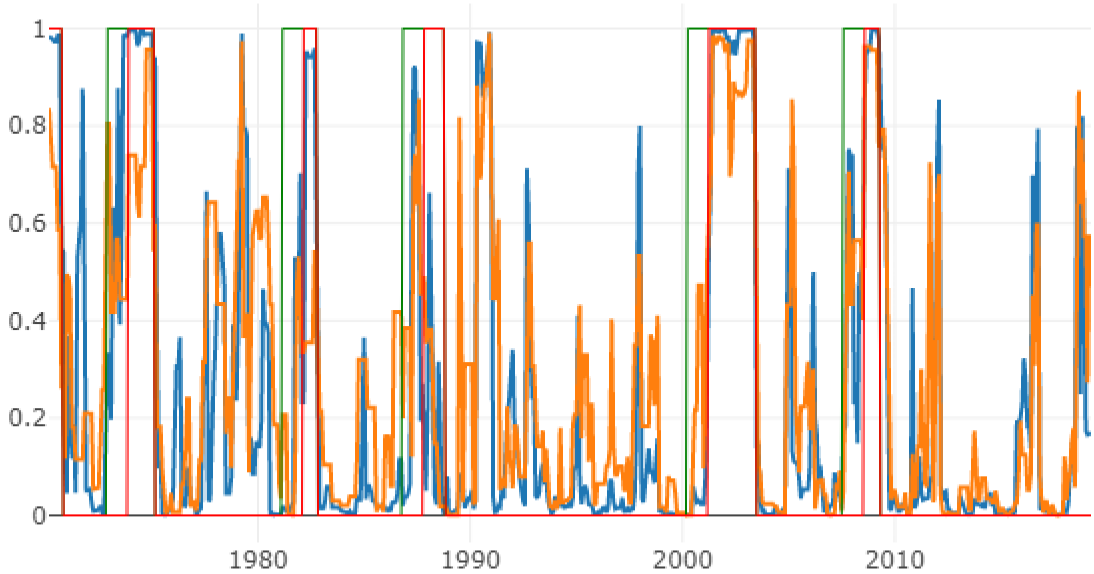


Figure 4.19: Example of plot with results for XGBoost (20% downfalls with 12 months lag), with versions: default (blue probability results) and optimized (orange probability results) hyperparameters.

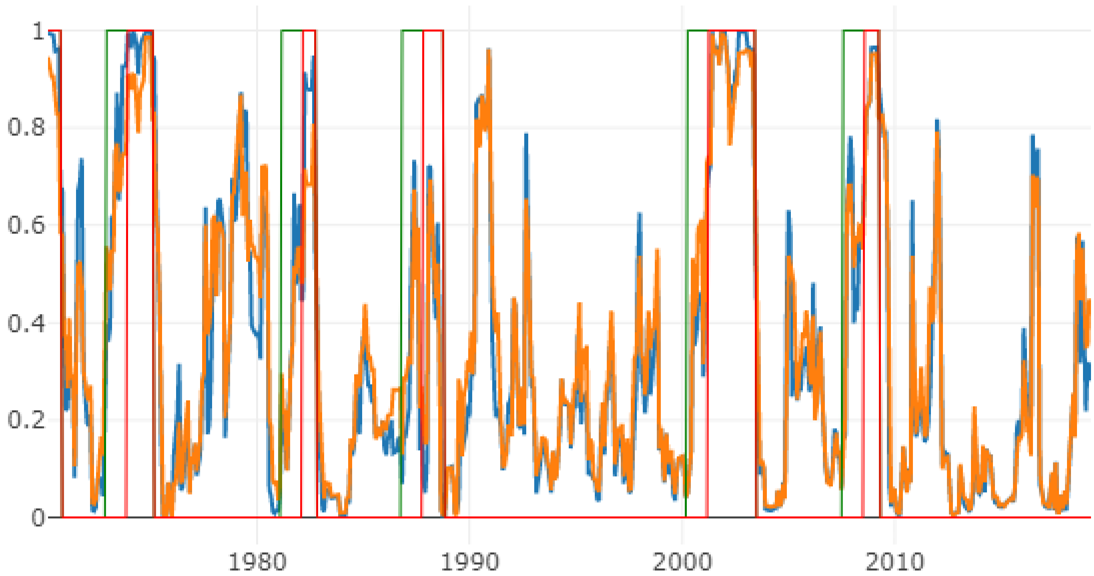


Figure 4.20: Example of plot with results (20% downfalls with 12 months lag) for average of three algorithms (logistic regression, random forest, XGBoost), with default (Blue) and optimized (Orange) hyperparameters.

In the figure 4.21 is observed tests for the 17.5% price decreases cases, comparing the default version of XGBoost with an optimized using AUC metric as fitness function. Although the results of the version with default hyperparameters reached greater values of probability more near to the downfall mark in some cases, it is possible to observe some earlier triggers by the optimized version in the falls of 1978, 1982, 1987, 1998 and 2001. This is considered positive results as well because it means that the optimized version could forecast some market downfalls earlier. Most of the other results in this test case are similar for both versions.

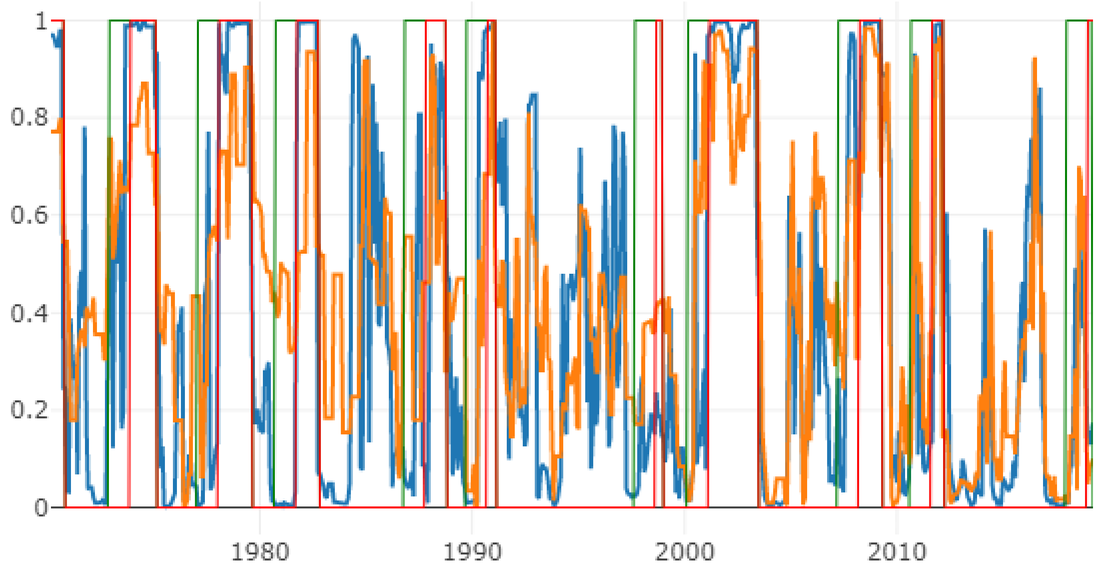


Figure 4.21: Example of plot with results for XGBoost (18% downfalls with 12 months lag), with versions: default (blue probability results) and optimized (orange probability results) hyperparameters.

In the table 4.8 are the metrics' values for the different versions of XGBoost and impact on the ensemble. As in the test case for downfalls of 20%, for the recall and AUTP metrics, both optimized versions also obtained greater values, demonstrating a better performance with the TP results, in particular for the optimized version with AUC as fitness function, in this case. In the other metrics the best values are shared with the default version and the optimized versions. This is also important in the sense that the optimized version does not increase that much the volatility in the results, although as was already referred, the FP results are not always undesirable.

Table 4.8: Value of metrics testing different versions of the XGBoost algorithm, with tests for prediction of -17.5% market price decreases, with 12 months in advance.

	Metric:	Accuracy		Precision		Recall (w/ falls)	Recall (w/o falls)		AUTP	AUC
Model - 18% downfalls, 12 months lag	**Threshold:**	0.5	0.35	0.5	0.35	0.5	0.5	0.35		
Default										
XGBoost		0.7169	**0.6820**	0.6961	**0.6082**	0.5749	0.2439	0.3171	0.2853	0.7317
Optimized, Fitness = Recall, TH=0.35										
XGBoost		0.6847	0.5712	0.6459	0.4921	0.5466	**0.3333**	0.5853	0.3927	0.7000
Optimized, Fitness = Auc										
XGBoost		**0.7475**	0.6458	**0.7207**	0.5549	**0.6478**	0.3325	**0.5854**	**0.3994**	**0.7655**
Impact on Ensemble										
Default XGBoost										
Ensemble (LR, RF, XGB)		0.7204	**0.6722**	0.6876	**0.5870**	0.6093	0.3000	0.4837	0.3889	0.7883
Optimized XGBoost, Fitness = Recall, TH=0.35										
Ensemble (LR, RF, XGB) w/ Optimized XGBoost		0.7288	0.6373	**0.7285**	0.5495	0.6318	**0.3559**	0.5041	**0.4267**	0.7929
Optimized XGBoost, Fitness = Auc										
Ensemble (LR, RF, XGB) w/ Optimized XGBoost		**0.7305**	0.6695	0.6913	0.5783	**0.6437**	0.3415	**0.5854**	0.4230	**0.8056**

Summing up, although there are a lot of similar results between the two versions (with default and optimized hyperparameters) proposed in this case study, it is possible to observe by the metrics and graphic representations that the version with optimized hyperparameters offers greater results for the same time spans. It also could precede with more advance some of the wanted events, constituting this way a more reliable alternative methodology.

Chapter 5

Conclusions

In this work was investigated the possibility of using a machine learning approach to detect downfalls in the S&P 500 index, resorting to economic variables. To approach this study three different case studies were formulated: one to anticipate bear markets 6 and 12 months ahead, another to anticipate other big marke price decreases (-18% and -15%), with the same anticipation and finally, in an attempt to improve the results obtained in previous case studies, the third case study is focused on hyperparameters' tuning in the XGBoost algorithm.

Reasonable good results were obtained for all the algorithms used, considering that all or almost all the falls give the probability of falling values greater than 0.4, having the tests for 12 months anticipation training, better results, than for 6 month training. Several economic variables were obtained but empirically was denoted that by reducing the number of these variables in classification it was possible to obtain better results.

As criteria evaluation, the metrics chosen was the accuracy, precision, recall, with thresholds of 0.5 and 0.35, and the AUTP (Area under True Positive) and AUC (Area Under the ROC curve). The recall and AUTP metrics were the most considered in this work since it gave a better insight on the TP results and do not have in account the FP results for its calculations. These FPs are not always undesirable since sometimes can alarm to other big market price decreases or to economic instability. The AUC metric is also of great relevance in this work since it evaluates the predictions having in account different thresholds. The Accuracy and Precision metrics although are still important to evaluate the volatility in the results of the different algorithms.

Regarding the recall and AUTP metrics, the logistic regression model had the best results in all the case studies. With 0.87 and 0.80 in the recall detecting bear markets with 6 and 12 months of anticipation, respectively. In the AUTP, logistic regression model had the best results with 0.62 and 0.64 detecting bear markets with the same anticipation patterns (6 and 12 months). Regarding the AUC metric, the Ensemble approach model with 0.89 had the best result anticipating bear markets with 6 months anticipation. The remaining best results of AUC in the other test cases are split between the Ensemble and logistic regression models. In the Accuracy and Precision metrics, the Ensemble approach (combining logistic regression, random forest and xgboost) and XGBoost models had the best results in the detection of bear markets. Ensemble With 0.88 in accuracy with previous 6 months bear markets detection and 0.79 in precision for the 12 months anticipation of -18% market price decreases, are examples of best results in these metrics. The XGBoost best results in these metrics are reflected in the graphic representation where the results show less variations and volatility through the time-series in negative labeled zones. The ensemble approach, was considered the most balanced method since it combines the highest results of each algorithm in the different metrics, and having the best and second best results in almost all metric's fields. However, regarding the goal pretended to achieve in this work, is fair to state that the Logistic Regression outperformed the other algorithms. It had greater values of probability and presented results with more antecedence in the detection of bear markets itself.

The case study that joins the prediction of smaller price decreases than bear markets, has higher results of probability in most of the events. However, the volatility of these results increased with the decrease of the fall that is tried to predict because the model is less and less able to distinguish between different patterns in the time-series.

In the last case study there was the attempt to optimize the results obtained for the XGBoost in the previous case studies, with hyperparameters' tuning, through a Genetic Algorithm. Although it was obtained a lot of similar results comparing to the "default" version of XGBoost, it was possible to obtain reasonable better results in the metrics of recall (e.g. 0.42 in default version to 0.55 the optimized for recall with threshold=0.35 not counting "bearish zones" as TP), and AUTP (e.g. 0.33 with default version for 0.40 in optimized versions), demonstrating a better performance with the TP results. Through the graphic representation is possible to observe a larger anticipation and higher values of probability of downfall, specially in the bear markets of 1987 and 2001.

Regarding the most important features in the models' production for downfall events prediction, it was made the features' importance in both case studies, and the economic variables that stood out were: i) the Conference Board Consumer Confidence index, ii) Number of 5% market pullbacks, before it reaches the last bull peak before the bear trend, iii) Rule of 20 and Shiller P/E, both variables of stock's valuation category, and iv) Misery Index, which has in account the inflation and unemployment rate for its calculation.

The prediction of Bear Markets did not reveal to be an easy task and one of the reasons why is because the periods where these kind of events happen, are not the only ones where are unusual feature's variations in the market. For example, US economy recessions often lead to significant stock market instability but not necessarily lead to bear markets. That is the reason why in several case studies there were FP results with more evidence on certain dates. However, this demonstrates the models' ability to capture other market declines and economic instability.

The idea of trying to predict with certainty the exact time before an event like this is going to happen, applying different types of model's training datasets (six or twelve months before), is difficult to achieve as shown in the results. However the methodology used in this work demonstrated a useful tool that through the analysis of some economic variables can trigger and alarm for significant market downfall events in the S&P 500 in the medium and long term. It is also worth mentioning the possibility of having this tool only using a personal computer with implemented programming techniques.

5.1 Future Work

This work was done in an exploratory manner and from it some things can be done to achieve better results than those obtained for this specific work or to obtain similar successful results focused on the economic/financial area. Some ideas for future work based in this work, are:

- Test the same approach for different stocks, not also in the US but also in other countries. E.g. US Nasdaq Index, EURO STOXX 50 (50 large blue chip companies in the Eurozone), Germany DAX (30 companies weighted by the market cap);

- Try to address in different ways the economic variables used (e.g. different transformations, get more data), in order to have models based in more variables and this way more reliable;

- Experiment the use of this indicator as an investment tool, to make short in periods of big probability of a downfall happens and comparing with other investment strategies (e.g. Buy and Hold) throughout time;

- Improve the GA, trying to use different fitness functions, different types of crossover (single-point, two-point and k-point crossover), different types of selection (E.g. Elitism and tournament), hypermutation, meta-optimization, and try to make it more efficient;

- Tuning the hyperparameters of other algorithms, specially the Random Forest which obtained the poorer results in most metrics, and try to utilize different approaches in tuning such as Bayesian Optimization;

- Approach this problem not as of binary classification nature, trying to distinguish different patterns regarding different economic conjecturs. E.g. Beginning of a market downfall, very close to a market downfall, in the bear zone itself, etc.

These are some of the possible and advised future developments that any individual following this work could take for the future.

Bibliography

[1] B. I. Jacobs and K. N. Levy. The complexity of the stock market. *The Journal of Portfolio Management*, 16(1):19–27, 1989. ISSN 0095-4918. doi: 10.3905/jpm.1989.409244.

[2] OECD. *OSLO Manual 2018*. OECD, 4th edition edition, Oct. 2018. Guidelines for Collecting, Reporting and Using Data on Innovation, 4th Edition.

[3] S.-S. Chen. Predicting the bear stock market: Macroeconomic variables as leading indicators. *Journal of Banking & Finance*, 33(2):211 – 223, 2009. ISSN 0378-4266. doi: https://doi.org/10.1016/j.jbankfin.2008.07.013.

[4] R. Shiller. The coming bear market? *Project Syndicate site*, 21, 2017. URL `https://www.project-syndicate.org/commentary/us-stock-volatility-bear-market-by-robert-j--shiller-2017-09`.

[5] W. B. national accounts data and O. N. A. data files. Global gdp this year. `https://data.worldbank.org/indicator/CM.MKT.TRAD.CD`, 2019. Combined GDP of all countries, from Jan. 1 2019 up to now.

[6] W. F. of Exchanges database. Stocks traded, total value (current us$), 2018. URL `https://www.worldometers.info/gdp/`.

[7] M. Kumar and T. M. Forecasting stock index movement: A comparison of support vector machines and random forest. *SSRN Electronic Journal*, 01 2006. doi: 10.2139/ssrn.876544.

[8] J. Coakley and A.-M. Fuertes. Valuation ratios and price deviations from fundamentals. *Journal of Banking & Finance*, 30(8):2325 – 2346, 2006. ISSN 0378-4266. doi: https://doi.org/10.1016/j.jbankfin.2005.08.004.

[9] G. Perez-Quiros and A. Timmermann. Firm size and cyclical variations in stock returns. *The Journal of Finance*, 55(3):1229–1262, 2000. doi: 10.1111/0022-1082.00246.

[10] S. Edwards, J. G. Biscarri, and F. P. de Gracia. Stock market cycles, financial liberalization and volatility. *Journal of International Money and Finance*, 22(7):925 – 955, 2003. ISSN 0261-5606. doi: https://doi.org/10.1016/j.jimonfin.2003.09.011. Regional and International Implications of the Financial Instability in Latin America.

[11] B. J. de Almeida, R. F. Neves, and N. Horta. Combining support vector machine with genetic algorithms to optimize investments in forex markets with high leverage. *Applied Soft Computing*, 64:596 – 613, 2018. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2017.12.047.

[12] P. Loungani, M. Rush, and W. Tave. Stock market dispersion and business cycles. *Economic Perspectives*, pages 2–8, 1991. URL `https://ideas.repec.org/a/fip/fedhep/y1991ijanp2-8nv.15no.1.html`.

[13] S. Fischer and R. C. Merton. Macroeconomics and finance: The role of the stock market. *Carnegie-Rochester Conference Series on Public Policy*, 21:57 – 108, 1984. ISSN 0167-2231. doi: https://doi.org/10.1016/0167-2231(84)90005-8. URL `http://www.sciencedirect.com/science/article/pii/0167223184900058`.

[14] R. Barro. The stock market and investment. *The Review of Financial Studies*, 1990.

[15] G. H. Moore and J. P. Cullity. Analysis. *Challenge*, 31(2):49–50, 1988. doi: 10.1080/05775132.1988.11471239.

[16] T. Z. Tan, C. Quek, and G. S. Ng. Brain-inspired genetic complementary learning for stock market prediction. In *2005 IEEE Congress on Evolutionary Computation*, volume 3, pages 2653–2660 Vol. 3, Sep. 2005. doi: 10.1109/CEC.2005.1555027.

[17] G. Armano, M. Marchesi, and A. Murru. A hybrid genetic-neural architecture for stock indexes forecasting. *Information Sciences*, 170(1):3 – 33, 2005. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2003.03.023. Computational Intelligence in Economics and Finance.

[18] J. A. Ou and S. H. Penman. Financial statement analysis and the prediction of stock returns. *Journal of Accounting and Economics*, 11(4):295 – 329, 1989. ISSN 0165-4101. doi: https://doi.org/10.1016/0165-4101(89)90017-7.

[19] O. Yefimochkin, R. Neves, and N. Horta. Fundamental: Using macroeconomic indicators and genetic algorithms in stock market forecasting. Master's thesis. Insituto Superior Técnico, Master Thesis.

[20] L. A. Teixeira and A. L. I. de Oliveira. A method for automatic stock trading combining technical analysis and nearest neighbor classification. *Expert Systems with Applications*, 37(10):6885 – 6890, 2010. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2010.03.033.

[21] A. Gorgulho, R. Neves, and N. Horta. Applying a ga kernel on optimizing technical analysis rules for stock picking and portfolio composition. *Expert Systems with Applications*, 38(11):14072 – 14085, 2011. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2011.04.216.

[22] G. Bontempi, S. Ben Taieb, and Y.-A. Le Borgne. *Machine Learning Strategies for Time Series Forecasting*, pages 62–77. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-36318-4. doi: 10.1007/978-3-642-36318-4_3.

[23] R. Adhikari and R. K. Agrawal. An introductory study on time series modeling and forecasting. *arXiv preprint arXiv:1302.6613*, 2013.

[24] C. Chatfield. *The analysis of time series: an introduction*. Chapman and Hall/CRC, 2003.

[25] G. Kitagawa and W. Gersch. A smoothness priors–state space modeling of time series with trend and seasonality. *Journal of the American Statistical Association*, 79(386):378–389, 1984.

[26] B. M. Pavlyshenko. Machine-learning models for sales time series forecasting. *Data*, 4(1):15, 2019.

[27] I. Georgoula, D. Pournarakis, C. Bilanakos, D. Sotiropoulos, and G. M. Giaglis. Using time-series and sentiment analysis to detect the determinants of bitcoin prices. *Available at SSRN 2607167*, 2015.

[28] R. P. Bunker and F. Thabtah. A machine learning framework for sport result prediction. *Applied Computing and Informatics*, 15(1):27 – 33, 2019. ISSN 2210-8327. doi: https://doi.org/10.1016/j.aci.2017.09.005.

[29] I. G. Maglogiannis. *Emerging artificial intelligence applications in computer engineering: real word ai systems with applications in ehealth, hci, information retrieval and pervasive technologies*, volume 160. Ios Press, 2007.

[30] B. G. Tabachnick and L. S. Fidell. *Using multivariate statistics*. Allyn & Bacon/Pearson Education, 2007.

[31] M. K. David G. Kleinbaum. *Logistic regression*. Springer, 2002.

[32] C.-Y. J. Peng, K. L. Lee, and G. M. Ingersoll. An introduction to logistic regression analysis and reporting. *The journal of educational research*, 96(1):3–14, 2002.

[33] A. Dutta, G. Bandopadhyay, and S. Sengupta. Prediction of stock performance in the indian stock market using logistic regression. *Int. J. Bus. In!*, 7:105–136, 01 2012.

[34] A. Liaw and M. Wiener. Classification and regression by random forest. *Journal Forest*, 23, November 2001.

[35] L. Khaidem, S. Saha, S. Basak, S. Kar, and S. Dey. Predicting the direction of stock market prices using random forest. April 2016. Cornell University work paper.

[36] L. Breiman. Bias, variance, and arcing classifiers. Technical Report 460, Statistics Department, University of California, 1996.

[37] R. Mitchell and E. Frank. Accelerating the xgboost algorithm using gpu computing. *PeerJ Computer Science*, 3:e127, 2017.

[38] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang, and Y. Si. A data-driven design for fault detection of wind turbines using random forests and xgboost. *IEEE Access*, 6:21020–21031, 2018. doi: 10.1109/ACCESS.2018.2818678.

[39] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu, and J. Peng. Xgboost classifier for ddos attack detection and analysis in sdn-based cloud. In *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 251–256, Jan 2018. doi: 10.1109/BigComp.2018.00044.

[40] P. Probst, B. Bischl, and A.-L. Boulesteix. Tunability: Importance of hyperparameters of machine learning algorithms. *arXiv preprint arXiv:1802.09596*, 2018.

[41] D. Beasley, D. R. Bull, and R. R. Martin. An overview of genetic algorithms: Part 1, fundamentals. *University computing*, 15(2):56–69, 1993.

[42] R. L. Barbosa. Ensemble of machine learning algorithms for economic recession detection. Master's thesis, Instituto Superior Técnico, 2018.

[43] V. H. Shah. Machine learning techniques for stock prediction. *Foundations of Machine Learning—Spring*, 1(1):6–12, 2007.

[44] World Development Indicators from The World Bank Group. Gdp ranking, 2018. URL `https://datacatalog.worldbank.org/dataset/gdp-ranking`.

[45] T. Durden. 13 of 19 bear market indicators have now been triggered - bofa. Zero Hedge, June 2018. URL `https://www.zerohedge.com/news/2018-05-31/13-19-bear-market-indicators-have-now-been-triggered-bofa`.

[46] T. Durden. 11 of 19 bear market indicators have now been triggered: Bofa, Dec. 2017. URL `https://www.zerohedge.com/news/2017-12-21/11-19-bear-market-indicators-have-now-been-triggered-bofa`.

[47] Google Finance. Standard & poors' 500. `https://www.google.com/finance`, Oct. 2019.

[48] M. Egan. Warning: 14 of 19 bear market signals triggered, bank of america says. CNN Business, Oct. 2018. URL `https://edition.cnn.com/2018/10/23/investing/dow-jones-stocks-bear-market/index.html`.

[49] T. Durden. Goldman's bear market indicator shows crash dead ahead, asks "should we be worried?". Zero Hedge, May 2018. URL `https://www.zerohedge.com/news/2018-09-05/goldmans-bear-market-indicator-shows-crash-dead-ahead-asks-should-we-be-worried`.

[50] F. R. B. of St. Louis. 10-year treasury constant maturity minus 3-month treasury constant maturity (t10y3m). retrieved from FRED, Federal Reserve Bank of St. Louis, Jan. 2019. URL `https://fred.stlouisfed.org/series/T10Y3M`.

[51] Edward Yardeni, Yardeni Research. `https://www.yardeni.com/`. Online; accessed throughout 2019.

[52] H. Li, Z. Yang, and T. Li. Algorithmic trading strategy based on massive data mining. 2014. Stanford University work paper.

[53] G. Creamer and Y. Freund. Predicting performance and quantifying corporate governance risk for latin american adrs and banks. *SSRN online repository*, 2004.

[54] P. Carmona, F. Climent, and A. Momparler. Predicting failure in the us banking sector: An extreme gradient boosting approach. *International Review of Economics & Finance*, 2018.

[55] S. Dey, Y. Kumar, S. Saha, and S. Basak. Forecasting to classification: Predicting the direction of stock market price using xtreme gradient boosting, October 2016.

[56] M. Y. Yu. *Predicting the Volatility Index Returns Using Machine Learning*. PhD thesis, 2017. University of Toronto.

[57] W. Liu and E. Moench. What predicts u.s. recessions? *International Journal of Forecasting*, 2014. doi: http://dx.doi.org/10.2139/ssrn.2495083.

[58] Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. *Ecological Modelling*, 154(1):135 – 150, 2002. ISSN 0304-3800. doi: https://doi.org/10.1016/S0304-3800(02)00064-9.

[59] C.-J. Lu. Integrating independent component analysis-based denoising scheme with neural network for stock price prediction. *Expert Systems with Applications*, 37(10):7056 – 7064, 2010. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2010.03.012.

[60] Y. Zhang and L. Wu. Stock market prediction of s&p 500 via combination of improved bco approach and bp neural network. *Expert Systems with Applications*, 36(5):8849 – 8854, 2009. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2008.11.028.

[61] B. Vanstone and G. Finnie. An empirical methodology for developing stockmarket trading systems using artificial neural networks. *Expert Syst. Appl.*, 36(3):6668–6680, Apr. 2009. ISSN 0957-4174. doi: 10.1016/j.eswa.2008.08.019.

[62] A. Canelas, R. Neves, and N. Horta. A sax-ga approach to evolve investment strategies on financial markets based on pattern discovery techniques. *Expert Systems with Applications*, 40(5):1579 – 1590, 2013. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2012.09.002.

[63] Y. Xia, C. Liu, Y. Li, and N. Liu. A boosted decision tree approach using bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications*, 78:225 – 241, 2017. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2017.02.017.

[64] J. Nobre and R. F. Neves. Combining principal component analysis, discrete wavelet transform and xgboost to trade in the financial markets. *Expert Systems with Applications*, 125:181 – 194, 2019. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2019.01.083.

[65] N. Mori, M. Takeda, and K. Matsumoto. A comparison study between genetic algorithms and bayesian optimize algorithms by novel indices. pages 1485–1492. GECCO '05 Proceedings of the 7th annual conference on Genetic and evolutionary computation, 01 2005.

[66] Federal Reserve Bank of St.Louis Database, FRED. https://fred.stlouisfed.org/. Online; accessed throughout 2019.

[67] Yahoo Finance Database. https://finance.yahoo.com/. Online; accessed throughout 2019.

[68] Bloomberg L.P., Bloomberg Database. https://fred.stlouisfed.org/. Online; accessed throughout 2019.

[69] Thomson Reuters Eikon Database. https://eikon.thomsonreuters.com/. Online; accessed throughout 2019.

[70] Plotly python open source graphing library. URL https://plot.ly/python/. Online; accessed throughout 2019.

[71] Python data analysis library — pandas: Python data analysis library. URL https://pandas.pydata.org/. Online; accessed throughout 2019.

[72] Api reference: pandas.dataframe. URL https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html. Online; accessed throughout 2019.

[73] statsmodels time-series analysis. URL https://www.statsmodels.org/dev/generated/statsmodels.tsa.stattools.adfuller.html. Online; accessed throughout 2019.

[74] scikit-learn: Machine learning in python. URL https://scikit-learn.org/stable/. Online; accessed throughout 2019.

[75] ipywidgets library documentation. URL https://ipywidgets.readthedocs.io/en/latest/#ipywidgets. Online; accessed throughout 2019.

[76] X. Library. Xgboost parameters. URL https://xgboost.readthedocs.io/en/latest/parameter.html. Online; accessed throughout 2019.

[77] A. E. Eiben, J. E. Smith, et al. *Introduction to evolutionary computing*, volume 53. Springer.

[78] C. Board. Consumer confidence survey®, 2019. URL https://www.conference-board.org/data/consumerconfidence.cfm.

[79] M. B. B. Insider). The '1994 moment' is keeping more and more bond traders awake at night, Jan. 2013. URL https://www.businessinsider.com/1994-federal-reserve-tightening-story-2013-1.

[80] A. Musacchio. Mexico's financial crisis of 1994-1995. May 2012. Harvard Business School Working Paper.

[81] M. Boesler. What happened to markets the last 15 times the fed tightened..., Jan. 2013. URL https://www.businessinsider.com/history-of-federal-reserve-tightening-2013-1.

[82] R. Dutt. Five events that shook world economy in 2016, December 2016. URL https://www.theguardian.com/business/2016/jan/09/2016-world-tumbles-back-economic-crisis.

[83] L. Elliott. Is 2016 the year when the world tumbles back into economic crisis?, Jan. 2016. URL https://economictimes.indiatimes.com/news/international/business/five-events-that-shook-world-economy-in-2016/articleshow/56182833.cms.

[84] D. J. Marotta. Black monday bear: The bear market of 1987, May 2018. URL https://www.forbes.com/sites/davidmarotta/2018/05/21/black-monday-bear-the-bear-market-of-1987/#7636612344b0.

[85] A. G. Malliaris and J. L. Urrutia. The international crash of october 1987: Causality tests. *The Journal of Financial and Quantitative Analysis*, 27(3):353–364, 1992. ISSN 00221090, 17566916.

[86] R. J. Shiller. Portfolio insurance and other investor fashions as factors in the 1987 stock market crash. *NBER Macroeconomics Annual*, 3:287–297, 1988.

[87] J. Itskevich. What caused the stock market crash of 1987?, Dec. . URL https://historynewsnetwork.org/article/895.

[88] N. B. of Economic Research. Us business cycle expansions and contractions. URL https://www.nber.org/cycles.html.

[89] J. T. Machado, G. M. Duarte, and F. B. Duarte. Identifying economic periods and crisis with the multidimensional scaling. *Nonlinear Dynamics*, 63(4):611–622, 2011.