Thesis Title: Bear Market Prediction Using Logistic Regression, Random Forest, and XGBoost

Pedro Jorge Bastos
pedro.jorge.bastos@tecnico.ulisboa.pt
Instituto Superior Técnico, Lisboa, Portugal

November 2019

Abstract

The stock market is considered one of the most complex systems in the world, consisting of 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 significant 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 higher 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, XGBoost, Genetic algorithm.

1. Introduction

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. 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. 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 lies the base of this thesis.

The bear market definition is a market loss of at least 20% in the market price and is a critical element of determining stock market's 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. The subject treated in this work is the previous detection of Bear Markets and other significant decreases in stock prices through the use of Machine Learning algorithms. These algorithms will be used (and optimized), and resorting to economic variables, it is aimed at the construction of models that will somehow be in accordance with the specifications proposed for this work - predict market downfalls with some advance.

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 predict big downfalls 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.

This work is composed of the following five chapters:

- Chapter 1 - Introduction
  Presents an overview of the work, providing its topic overview, objectives, and contributions.

- Chapter 2 - Background
  Sets the theoretical basis of the concepts used in this thesis. Includes also the state of art regarding this subject.
- Chapter 3 - Proposed System Architecture
  It reveals the proposed system architecture and implementation of several methodologies and approaches in order to achieve and accomplish these work objectives.
- Chapter 4 - Results
  The chapter will commence with a brief explanation of the metrics used as criteria evaluation, followed by the methodology used summarized, and finally, the results obtained in the different case studies prepared for this thesis.
- Chapter 5 - Conclusion
  The final section of this work where will be summarized the work done, the several results obtained and the conclusions drawn from these.

2. Background and State of art

2.1. Stock Market's Variations and Analysis

The stock market is a public, controlled, secure, and managed environment for securities trading. It is defined as non-stationary because the distribution of financial time-series is changing over time and 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 stock market index and exchange rates, difficulting the prediction of financial market' s movements [13]. To try to predict these variations, firstly, as is stated in [8], 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 [8]. However, the media nowadays delineate a "classic" or "traditional" bear market as a 20% decline in stock prices [21]. The same happens in the bull market, with the change of stock prices in the opposite direction (an increase of 20% in stock price). It is usual to consider all the business variations as a cycle because it alternates between these two phases. Beyond these, it is also likely to add into consideration the points in between, where the market is "sideways".

Several works have been done with the premise that stock market's predictions based on past data are 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) [2].

- 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. The fundamental analysis is based on several factors which can be summarized in three main strands: Economic Analysis, Industry Analysis, and Company 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. One of the best known technical indicators is the simple moving average.

2.2. Analysis of Time Series

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. Study time series can be made for several purposes, such as the future's forecasting based on prior knowledge, which is what will be attempted in this work [5].

In time series analysis, there is a concern with decomposing the variation in the series in four different components: Seasonality, Trend, Other Cyclical variations, and Other irregular variations [1][7].

Time series forecasting methods can be broadly split 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 downfalls 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).

2.3. ML Models

2.3.1 ML approach

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 and fits this way into Classification problem type (Supervised Learning). Supervised Learning is when it is used as a benchmark a set of examples and their class label (training set), the process consists in associating the different unlabelled objects to a specified output.

To deal with this type of problem, 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, 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).

2.3.1.a Logistic Regression - 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 [22] (expression 1). Logistic regression solves these problems by applying the logit transformation to the dependent variable.

$$Logit(Y) = \ln \frac{p_i}{1 - p_i} = \beta_0 + ... + \beta_p * x_{jp} = z_i \quad (1)$$

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 1, it is possible to derive 2 that describes the relationship between z_i and the probability p_i of the event to occur

From the expression 1, it is possible to derive 2 that describes the relationship between z_i and the probability p_i of the event to occur.

$$p_i = Probability(Y = outcome\, of\, interest \mid X = set\, of\, predictors\, x_{ij}) = \frac{1}{1 + e^{-z_i}} \quad (2)$$

The outcome of this function can only be between 0-1 range, and adapting to the present work will represent the probability of a stock market downfall event occur.

2.3.1.b Decision Trees - Decision trees are a popular tool used for classification and prediction purposes. Two well-known methods of ensemble trees are boosting and bagging of classification trees [15].

2.3.1.b.a Random Forest (Bagging Trees) - The idea of bagging is based on creating several subsets of data from the training sample chosen randomly with replacement. As a result, we end up with an ensemble of different models. The average of all predictions from the different trees is used, being more robust than a single decision tree. The Random Forest algorithm 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.

2.3.1.b.b XGBoost (Boosting Trees) - In boosting, successive trees give extra weight to points incorrectly predicted by earlier predictors. In the end, a weighted vote is taken for prediction [15]. Gradient Boosting algorithm is one of the variations of boosting.

The boosting algorithm will run for **M** boosting iterations, consisting in the following steps [17], that can be executed for several iterations until residuals have been minimized as much as possible:

- A model F_m is defined to predict the labeled variable y ($F(x) = \hat{y}$);
- This model will be associated with a residual $f_m(x) = y - F_{m-1}$ 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.

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 [9, 17, 26]. Each XGBoost iteration is given by the expression 3:

$$Obj(\Theta) = L(\Theta) + \Omega(\Theta) \quad (3)$$

Being $L(\Theta) = l(y_i, \hat{y}_i)$, the loss function that calculates the difference between the prediction and the true label, and $\Omega(\Theta)$ the function that describes the complexity of the estimator (tree), and will regulate the model overfit.

2.3.2 Genetic Algorithms

Some ML algorithms (such as random forest and XGBoost), already own first-level model parameters, which are defined during the training process, and besides these, it is also possible to define the hyperparameters in such a way that it maximizes the results [20]. For an appropriate selection of hyperparameters, it is possible to resort to other optimization algorithms that will dictate the best hyperparameters combinations. like Genetic Algorithms (GA) GAs are based on the genetic processes of biological organisms and premised on the principles of natural selection and ”fittest survival”, the GAs can evolve solutions to real-world problems.

The process will be executed by several ”chromosomes”, and consists of the following steps: Train, Selection, Crossover, Mutation. This process will be executed until it found (ideally) the converged value of ”fitness” (figure 1).

2.4. State of art

2.4.1 Stocks and Economic Variables

The US economy it is considered one of the strongest at the global level and consequently, may cause significant effects on all other economies

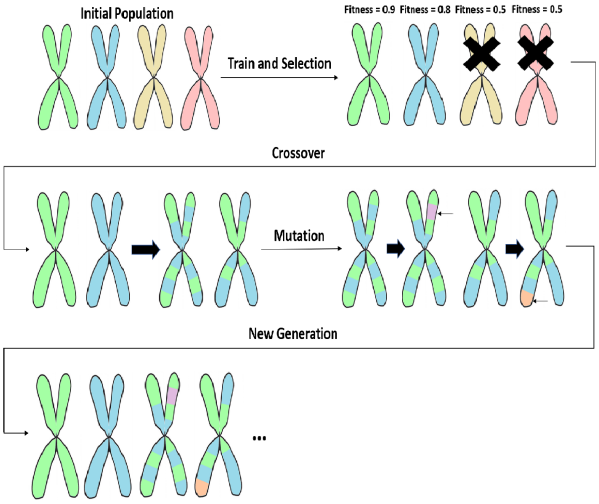

Figure 1: Process of a genetic algorithm.

worldwide. S&P 500 was the index chosen to analyze in this work because it is the broadest measure of the US economy among the major indices, due to the number of important entities it comprises.

It was discovered from researches, several indicators used by relevant finance entities (Bank Of America Merryl Linch, Goldman Sachs Inc.) that demonstrated to correlate with bear markets, since most of them were triggered at such times. Furthermore, other economic variables used as indicators in [3, 8] and [24] also showed correlation and 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.

2.4.2 Models and Results

In the work [14], "*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 - [3, 6, 10].

In [6] is achieved 94% accuracy using XGBoost to predict the bank failure in the USA, between 2001-2015 and was the best algorithm in this work comparing with results obtained with other different models such as the random forest, and logistic regression, where obtained 92% and 84,21% of accuracy, respectively.

[3], 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 when predicting recessions with no lag with the logistic regression model (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 in tests. Both works of [3, 16], make similar approaches like the one adopted in this work, like the graphic representation of the time series predictions and some of the metrics used as criteria evaluation (e.g. AUC and recall) but instead of trying to predict the downfalls in a singular stock market, they went for the predictions of U.S economy recessions.

In the work [11] is used XGBoost to predict financial securities trends and had 87% accuracy for the 60/90 days predictions. In [25], using XGBoost as well, achieved 85% accuracy in predicting the Chicago Board Options Exchange (CBOE) Volatility Index.

In the work [23], 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. In the work [19], the XGBoost hyperparameters were optimized, resorting to the use of a Genetic Algorithm. 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 Genetic algorithms in finance's purposes, the choice for this work was on this kind of approach. Also, in the work [18], it is made several comparisons between both approaches, and the Bayesian Optimization ended up having worse results, which helped to support the choice made.

Considering all the works presented, XGBoost and Random Forest had the best results and seemed to be a reasonable choice for the Machine Learning approach in the binary classification problem of this work. The use of Logistic Regression, although often observed, does not give very often the best results. However, it is considered for the approach in this work in order to have a linear model, differing from the other non-linear models chosen.

3. Proposed System Architecture

An architecture containing modules will be used in order to get different sections connected and working independently. The modules that constitute the system are the following four: Data module, Transformation Module, Classification/Validation Module, and User Interface Module (figure 2)

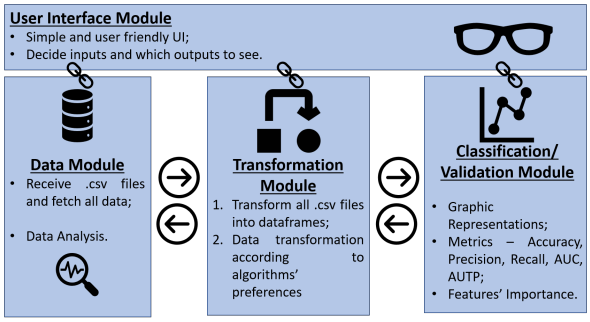

Figure 2: Work module-based architecture diagram.

3.1. Data Module

The quantity and quality of data gathered is one of the essences for this type of work. There are websites such as the Federal Reserve Bank of St. Louis website (FRED) [12], and computer terminals with access to Bloombergś databases [4], where it is possible to export financial data from. Hereupon, this module basically contains all the .csv files obtained, and the user will be able to analyze it (Analyst tool) and make several choices for further utilization. The Analyst tool will allow the analysis of different data: With each variable individually will be possible to have a description with different fields (e.g., max, min, mean) and have a graphic representation with the downfalls' dates marked; From a general point of view will be possible to analyze the correlation between all variables.

3.2. Transformation Module

After aiming all the data needed for this work, there will be made different transformations for the several data obtained. These are some of the tasks that will be assigned to the Transformation Module:

- Convert the .csv files in *Python* dataframes for further use;
- Analyze and mark downfall events dates in the S&P' 500 index dataframe and add a lag to these so in the classification it would be possible to correlate the different features with time spans pre-downfall;
- In order to make a multivariate forecast of the S&P 500 index, firstly it will be needed to verify if each variable used is stationary, and if not, it will be necessary to proceed to a transformation with a simple moving average or obtaining the variation of the variable in question.
- All data will be gathered/merged in one dataframe and then divided into input and output, in order to facilitate the classification part.

3.3. Classification/Validation Module

This module will be responsible for receiving the dataframes regarding the different variables chosen as features (input), and the S&P 500 index already marked with the downfalls in which the classification labels will be based on (output), and in the end produce forecasts of downfall events in the S&P 500 index, as accurate and as far in advance as possible.

3.3.1 Classification Process with Time-Series Cross-Validation

The user will be getting the response of classification for the three different ML models (Logistic Regression, Random Forest, and XGBoost), plus the average of these (Ensemble approach).

It will be made a cross-validation time-series method to ensure that the models produced are timelessly validated with out-of-sample tests. Data will be split into time units (e.g., days, months, etc.). 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.

In the end of all iterations of the cross-validation, the final four (Logistic Regression, Random Forest and XGBoost and Ensemble approach) results' dataframes will detain the probabilities of downfall events from 1970 to 2019 through a merging process as well as four python dictionaries with the feature importance for all the models produced in each iteration of the cross-validation method.

3.3.2 Validation of Results

The results will be displayed using the method *plot_results()*, which makes the graphics with the results for all the models throughout time, from 1970 until 2019, with the market downfalls and respective lags marked. It will also be calculated the results for several metrics used, such as the confusion matrix, accuracy, precision, recall, AUC, and AUTP (Area Under True Positive) for an analytic examination of the results (method *metricsResults()*). Through the use of *features' Importance* methods on *sklearn* 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 *feature importance* for the four models in each iteration in time where a new model was trained.

4. Results & Metrics

4.1. Metrics

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.

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. The following metrics explained are calculated using these four confusion matrix concepts.

4.2. Accuracy

Accuracy is defined as in 4

$$Accuracy = \frac{TP + TN}{FP + FN} \tag{4}$$

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

4.2.1 Precision

Precision is defined as in 5

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

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.

4.2.2 Recall

Recall is as defined as in 6

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

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.

4.2.3 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.

The curve will be set, having into account different thresholds and is a performance measurement for classification problems.

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.

4.2.4 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.

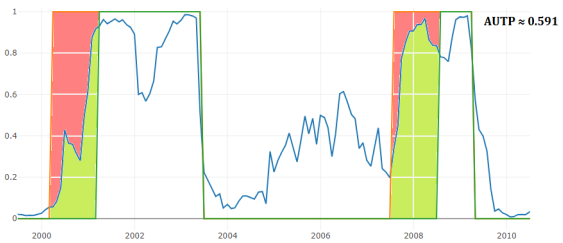

Figure 3: Display of algorithm results and important areas for AUTP result

$$AUTP = \frac{Algorithm's\ results\ inside\ TP}{Total\ area\ inside\ TP} \tag{7}$$

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 as positive labels. Ideally, the whole area with positive labels had to be cover by probability = 1 to have the maximum value of AUTP.

4.3. Methodology

Firstly there will be the need to convert all the raw data (index and economic variables) into *python* dataframes and handle the missing values so the different algorithms can use it. Before it enters the classification section, the data will be split into input (features) and output (labels). To get the anticipations in predictions, a lag will be added to the stock's dataframe, so that way, the models can try to correlate the different economic features used with periods chosen before the actual downfall occurs. Empirically and with the help of the metrics used, it was observed that reducing the number of economic variables led to better results in all algorithms.

The classification process will be made resorting to a time-series cross-validation method, with out-of-sample data tests, that will make as many iterations as decided by the user and split the data that number of times. In each iteration, it will be compared to the first date of the split dataframe with the stock downfalls' known dates, and if it had already passed any of these, the output used in the classification would be updated.

After this, it will be made the model's training. Three different Machine Learning algorithms will be used - Logistic Regression, Random Forest, XGBoost - and the ensemble of these three, to produce models to predict the wanted events (market downfalls), and the results acquired will be probabilities of an event happening, for several instants

in a time-series dataframe. Ideally, it is desired that these results would be maximum probabilities ($\approx$ 1) in pre-downfalls' periods and null probabilities of downfalls in the remaining times. In this process, the features' importance values when training the model, and the predictions made by the models will be saved in different dataframes for further utilization.

Using the dataframes containing the models' results, it will be plot a graphic representation of time-series with the results from 1970 to 2019 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, recall, precision, AUC, and AUTP.

It will also be tested the possibility of optimizing the algorithm's results through the optimization of the hyperparameters resorting to an implemented GA (Case study C). The only difference in the implementation of this optimization will stand on subdividing the training subset into training and validation and making several iterations of the Genetic Algorithm - process training-selection-crossover-mutation - until it finds a solution with hyperparameters that ideally will allow the optimization of the results obtained in previous case studies.

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

In this study case, it was obtained results trying to forecast downfalls with 6 and 12 months of anticipation in the S&P 500 in three different algorithms and for the average (ensemble) of all together. In total, eight different results were collected and will be analyzed separately hereinafter.

Table 1: Test in 1970-2019 for average model with 12 months lag

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

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. The Logistic Regression and the Ensemble also dispute the best value in the AUC metric. In this case study, it is possible to observe that the LR obtained the best results in both recall and AUTP metrics. The success in anticipating the event will be more reflected in these metrics of recall (TP rate) and AUTP because they do not have into account the FP results for its calculations. The more balanced test results were the ones from the Ensemble approach since it obtained most of the second-best results in all metrics.

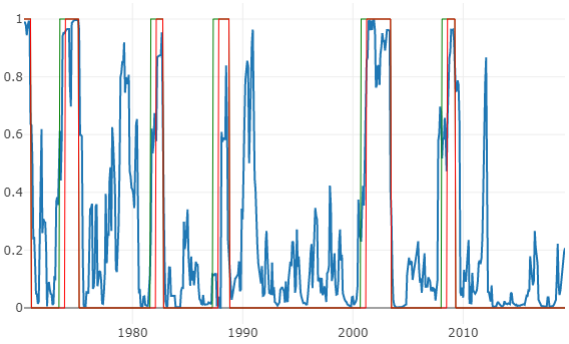

Figure 4: Results in 1970-2019 predicting 20% downfalls, with Ensemble model and 6 months lag

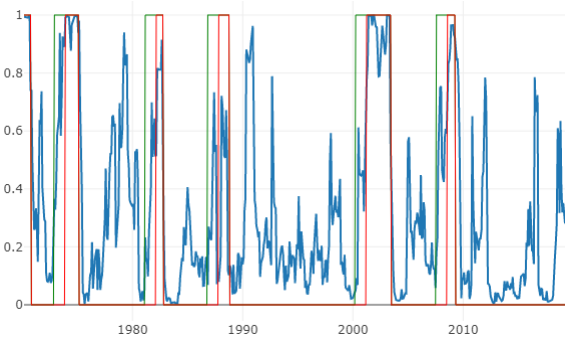

Figure 5: Results in 1970-2019 predicting 20% downfalls, with Ensemble model and 12 months lag

Observing the results in the graphic representation for the Ensemble of the three algorithms (figures 4 and 5) demonstrates 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's lack of FP, it "cleans" the negative labeled zones, having less FP than the LR and RF algorithms.

Training the data having the positive labels 12 months before the bear market zones, it is possible to observe by the graphic representations of the results that besides 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, than in training with six months of downfall's lag. The use of the Ensemble approach can distinguish better between the more significant and smaller downfall events. However, the Logistic Regression presented the best results in the positive labeled zones, with higher values and more

antecedence when anticipating.

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

The only difference within the whole process to the Case Study A will be on the output of the data set used. In addition 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.

However, the results for the several algorithms did not happen to be way different than in case study A. The Logistic Regression managed again to outperform the remaining models, with the best results in recall and AUTP metrics, and in some case tests, even had had the best results in all the metrics evaluated.

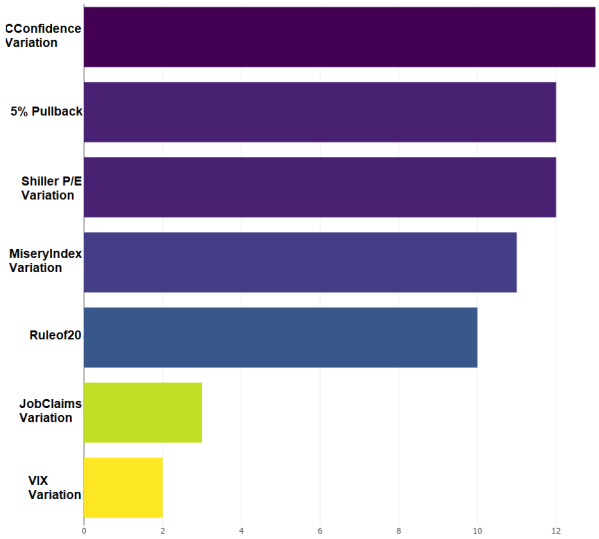

Figure 6: Average Feature Importance of the three models trained (LR, RF and XGB) to predict -18% market decreases with 12 months of anticipation. In the time-series instant: 30-02-2002 (after 1987 bear market).

Regarding the feature importance, for both cases study A, four economic variables stood out from the remaining. 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 into consideration when training the models. In the case study B, it was denoted that another economic variable was also taken into more consideration in models' training, comparing to the previous case study: the Misery Index.

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

This case study will focus on the XGB algorithm's results to improve through the use of a Genetic Algorithm (GA). The implementation of this case study is similar to the previous ones, but instead of using the algorithm's *python* methods with default hyperparameters, it will first go through a process to choose the hyperparameters and expecting, in the end, to optimize the results.

As it is possible to observe by the visible results (figure 7), the optimized hyperparameters version obtained higher results and could antecede with more advance the market downfalls of 20% in the events of 1987 and 2000. Regarding the other 20% downfalls and the FP results, both versions obtained similar results. However, the optimized hyperparameters version has its FP a little time before 1990, which matches the US recession and a significant market downfall as well.

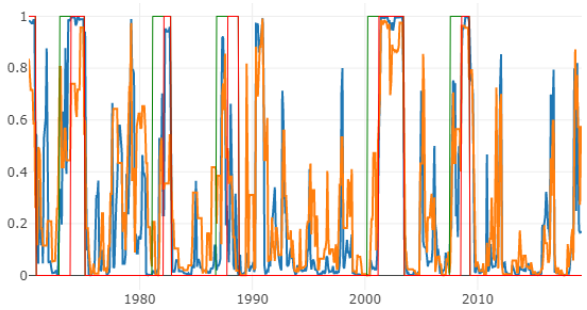

Figure 7: Example of plot with results for XGboost (20% downfalls with 12 months lag), with versions: default (blue probability results) and optimized (orange probability results) hyperparameters

The results for metrics were also compared between versions, and the optimized versions (for different fitness functions) obtained better results regarding the metrics which give a better insight about the TP results (recall and AUTP), although the results of the other metrics are in some fields higher in the default version.

Summing up, by the observation of the metrics and the graphic representations, it is possible to conclude that the version with optimized hyperparameters offers greater results for the same periods, and it can precede with more advance some of the wanted events, constituting this way an alternative methodology more reliable.

5. Conclusions

This work investigates 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 significant market 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 twelve months anticipation training, better results, than for six-month training.

The recall and AUTP metrics were the most considered in this work since it gave a better insight on the TP results and did not have in account the FP results for its calculations, which are not always undesirable since sometimes can alarm to other significant market price decreases or to economic instability. The AUC metric is also of great relevance in this work since it evaluates the predictions having into account different thresholds. The Accuracy and Precision metrics, although they are still essential to evaluate the volatility in the results of the different algorithms.

Regarding the recall metric, 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 six and twelve months of anticipation, respectively. In the AUTP, the logistic regression model had the best results with 0.62 and 0.64 detecting bear markets with the same anticipation patterns (six and twelve months). Regarding the AUC metric, the Ensemble approach model with 0.89 had the best result anticipating bear markets with six months anticipation. The remaining best results of AUC in the other test cases are split between the Ensemble and LR 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 six months bear markets detection and 0.79 in precision for the twelve 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 fewer 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, it is fair to state that the Logistic Regression outperformed the other algorithms since it had higher values of probability and presented results with more antecedence in the detection of bear markets itself.

In the last case study, there was an attempt to optimize the results obtained for the XGB in the previous case studies, with hyperparameters' tuning, through a GA. Although it was obtained a lot of similar results compared to the "default" version of XGB, 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, it is possible to observe more substantial anticipations and higher values of probability of downfall, especially 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 time spans where these kinds of events happen, are not the only ones where are unusual feature's variations in the market, for example, US recessions usually leads 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 some 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 (6 or 12 months before), is challenging 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 an alarm for 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.

References

[1] Ratnadip Adhikari and Ramesh K Agrawal. An introductory study on time series modeling and forecasting. 2013.

[2] G. Armano, M. Marchesi, and A. Murru. A hybrid genetic-neural architecture for stock indexes forecasting. *Information Sciences*, 170(1):3 – 33, 2005. Computational Intelligence in Economics and Finance.

[3] Rodrigo Lopes Barbosa. Ensemble of machine learning algorithms for economic recession detection. Master's thesis, Instituto Superior Técnico, 2018.

[4] Bloomberg terminal Bloomberg L.P., 2019.

[5] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. *Machine Learning Strategies for Time Series Forecasting*, pages 62–77. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[6] Pedro Carmona, Francisco Climent, and Alexandre Momparler. Predicting failure in the us banking sector: An extreme gradient boosting approach. *International Review of Economics & Finance*, 2018.

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

[8] Shiu-Sheng Chen. Predicting the bear stock market: Macroeconomic variables as leading indicators. *Journal of Banking & Finance*, 33(2):211 – 223, 2009.

[9] 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.

[10] Germán Creamer and Yoav Freund. Predicting performance and quantifying corporate governance risk for latin american adrs and banks. 2004.

[11] Shubharthi Dey, Yash Kumar, Snehanshu Saha, and Suryoday Basak. Forecasting to classification: Predicting the direction of stock market price using xtreme gradient boosting, 10 2016.

[12] Federal Reserve Bank of St.Louis FRED.

[13] Manish Kumar and Thenmozhi M. Forecasting stock index movement: A comparison of support vector machines and random forest. *SSRN Electronic Journal*, 01 2006.

[14] Haoming Li, Zhijun Yang, and Tianlun Li. Algorithmic trading strategy based on massive data mining. *Stanford University*, 2014.

[15] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *Forest*, 23, 11 2001.

[16] Weiling Liu and Emanuel Moench. What predicts u.s. recessions? 2014.

[17] Rory Mitchell and Eibe Frank. Accelerating the xgboost algorithm using gpu computing. *PeerJ Computer Science*, 3:e127, 2017.

[18] Naoki Mori, Masayuki Takeda, and Keinosuke Matsumoto. A comparison study between genetic algorithms and bayesian optimize algorithms by novel indices. pages 1485–1492, 01 2005.

[19] João Nobre and Rui Ferreira Neves. Combining principal component analysis, discrete wavelet transform and xgboost to trade in the financial markets. *Expert Systems with Applications*, 125:181 – 194, 2019.

[20] Philipp Probst, Bernd Bischl, and Anne-Laure Boulesteix. Tunability: Importance of hyperparameters of machine learning algorithms. *arXiv preprint arXiv:1802.09596*, 2018.

[21] Robert Shiller. The coming bear market? *Project Syndicate site*, 21, 2017.

[22] Barbara G Tabachnick and Linda S Fidell. *Using multivariate statistics*. Allyn & Bacon/Pearson Education, 2007.

[23] Yufei Xia, Chuanzhe Liu, YuYing Li, and Nana Liu. A boosted decision tree approach using bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications*, 78:225 – 241, 2017.

[24] Edward Yardeni. Yardeni research. Zero Hedge.

[25] Michael Yangmeng Yu. *Predicting the Volatility Index Returns Using Machine Learning*. PhD thesis, 2017.

[26] 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.