Hybrid System of Classification Algorithms Voting System with Genetic Algorithm for Dividend Stocks Ranking

Rodrigo Henriques da Silva Lopes
rodrigo.h.lopes@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

June 2022

Abstract

In this thesis, a voting system is proposed with classification algorithms (XGBoost and SVM) that intends to analyze the financial data of the companies that constitute the S&P500, and predict which ones (of those that present a streak of increasing the dividends paid) will continue that streak and which ones will break it. The algorithms will have a parameter optimization phase, done by a Genetic Algorithm and grid search, and then move on to the prediction phase. They will be tested in the years 2017, 2018 and 2019, and will create a system that assigns a rank to each stock, thus making it easier to choose the best ones to invest in. On top of this, a sliding window method will be implemented, where the system will be evaluated for a sequence of test years, making it easier to get an accurate and very reliable system. This sliding window contains all the classification and prediction modules in order to sequentially run the system several times, thus improving its predictions with each iteration.

Keywords: Financial Analysis, Dividends, XGBoost, SVM Genetic Algorithm, Grid Search

1. Introduction

The stock market, is a well studied market, where many investors have already participated, as it has always attracted many types of investors. This market has many components and consequently changes due to many factors, which causes many investors, too often, to lose with investments . Still, if one researches and studies the market enough, there is proof that one can profit a lot as well.

These investors differ a lot in types of preferred investments that they tend to keep more often in their portfolios. One way to invest, for example, is in dividend-paying companies. In this case investors will be looking for companies that will continue to pay dividends and keep them rising steadily. This is not always easy to predict, as there are many factors that can lead a company to change its dividend policy.

Another big part that motivated this project was the work that has been done on machine learning(ML) and supervised leaning algorithms, which over the years, many ways have been found to make them more efficient and more able to predict future results. These algorithms can be used to predict a number of different aspects in a large range of different situations, which is why they are becoming increasingly popular when it comes to financial investments. And in a market like the stock market, where there are many different types of data that can be fed to these algorithms, it's normal that in some cases they can be quite efficient. Even so, their predictions are not certain, and probably never will be, and that's why studies such as this project are continuously being done - In order to better optimize prediction processes like this.

In the case of this project, we will apply classification algorithms to the dividend stocks market, in order to try to predict the future situations of companies' dividend policies.

2. State-of-the-Art

2.1. Dividends

In 2001, [7] conducted a study that showed a decrease in the percentage of dividend paying companies. It showed that between the years of 1978 and 1999, the proportion of companies that effectively payed dividends, went down, from 66.8% to 20.8%. Their study presented a variety of factors that could have caused this decrease in dividend paying companies (e.g., "lower transaction costs for selling stocks for consumption purposes" and "larger holdings of stock options by managers who prefer capital gains to dividends"). However, in 2004, [5] continued to investigate the subject in question, and found that, although, there was a decrease in the percent-

age of companies that payed dividends, the aggregate amount of dividends being paid was, in fact, increasing, which happened due to the fact that the majority of payers that stopped paying dividends, corresponded to relatively small firms, and the increase of dividend payments, by much larger firms, "swamp" the reductions of smaller firms dividend payments.

As stated by [1], there's a very important statement when discussing dividend policy, that says that in case of a dividend increase there will be a positive reaction by the market and all its constituents, and respectively, in case of a dividend decrease by a company, then the market will react very negatively, as it could mean the some problem relative to the company in question may be resurfacing.

In order to better understand dividend smoothing and what drives certain companies to do so, some studies were made by [10] on a sample of more than two-thousand firms from around the world. As a result from those studies, [10] cites that "Managers of firms with low market-to-book ratios, less cash, low dividend payouts, and few tangible assets engage greater dividend smoothing", while firms that are in their early stages, usually show less dividend smoothing.

Corporate Social Responsibility(CSR), has become more important and relevant over time. Hence, a study was made by [3], in the interest of getting a better understanding on the matter. The study showed that companies with lower CSR appeared to give out smaller dividend payments, than companies with a higher CSR. Low CSR firms also take less time to adjust their dividend payments, which make them less stable then dividends given by high CSR firms. Finally, [3] showed that firms that are involved in controversial products or services, for example, alcohol, usually give less dividends.

In 2020, [**?**] studied the effects of the COVID-19 pandemic on US-firms' dividend payouts. Results showed that, out of a pool of 1400 dividend paying firms, about 15.2% cut their dividend payments, and 6.6% omitted their dividend payments completely.

Dividend investing has become more and more common each year, and in 2021, [4] developed a study with the sole purpose of comparing several portfolios according to different dividend investing strategies to a classical SP500 portfolio. [4] used portfolios such as DC20(Dividend Constant 20 years) and DR10(Dividend Raise 10 years). These portfolios were tested in several decades and the results implied that dividend growth portfolio strategies outperformed the typical S&P500, when measuring certain performance metrics such as the annual yield and return, and Sharpe Ratio. The one portfolio strategy that stood out the most was the DC20 as the DR10 sometimes had very little companies and could show some levels of discrepancy, unlike the DC20.

2.2. Machine Learning

SVM has been very commonly used in all sorts of classification problems. One typical situation where one might use an algorithm such as this, is in financial related predictions. It is very common to use SVM fed with technical indicators in order to predict future stock prices of one or more companies [8]. And in 2007, [8], conducted a study where, instead of resorting to technical indicators, fed the algorithm with the information from the companies' financial statements instead. This Fundamental Analysis type of approach showed great results, and the SVM model appeared to display a better accuracy than those models who analyzed technical information. Then, in 2008, [6] developed a study, where instead of trying to predict stock prices, wanted to see if one could predict a company's future dividend policy, and if such could be done with good accuracy performances when resorting to an SVM model. [6] came to the conclusion that, even though the algorithm appeared to have great levels of accuracy in its predictions, it also had a substantial error associated with it. The experiment was done using several kernels as well, so as to get the best out of the algorithm and eventually the kernel with best results was the RBF kernel. Furthermore, in 2010, [2] tested several machine learning classification techniques on how well they would preform in predicting future dividend policies for several companies. The one that appeared to have the best performances out of the rest, was the SVM algorithm.

In order to test the performance of several ML algorithms, [9] gathered the financial information of several Taiwanese companies, and ran them through various ML algorithms, both supervised and unsupervised ML algorithms, and even tested with an hybrid form of DBN-SVM, to see which would give the best predictions. Inside the supervised learning algorithms were the SVM and XGBoost, which incidentally were the ones that showed some of the best performances. An also good approach was the hybrid algorithm, as it made better predictions than those of the SVM, although, the XGBoost was the one that evidently stood out from all of them. During recent years, with more incoming studies on XGBoost, it became a very popular algorithm in financial prediction, and in 2021, [11] started to investigate the performance of certain ML algorithms when trying to predict future dividend policies, and of course, one of them would be XGBoost itself, with the other one be-

ing Multi-Layer Neural Networks (MLNN). The algorithms would receive data from the companies' financial statements(e.g., cash, sales, earnings per-share, etc), and try to predict if a company would pay dividends in the future or not. Even though both algorithms showed good results, XGBoost was the one that stood out the most, specially when being run with carefully chosen parameters. But this isn't the only environment for which the XGBboost evidently exceeds in comparison to other machine learning algorithms. In 2020, [12] conducted a study to see if machine learning algorithms could detect and prevent financial fraud, a problem that very present and important in the world of finance. The study involving XGBoost also considered several other ML algorithms, such as Naive Bayes and Logistic Regression, but XGBoost clearly outperformed the other algorithms, and again stood out as being the most trustworthy in financial related predictions.

3. Implementation

Figure 1: Graphical Representation of the Overall Architecture of the Implemented System

The first part of this project is based on obtaining the data that will be needed for the project and processing it so that it is ready to serve as input for the classification modules.

3.1. Data Loading Module

First there's the *Data Loading* module. This module is used to collect Financial data from all companies in the S&P500 index, do some minor processing in the database to adjust the data to what is needed and append all different types of data collected during this phase. The main database used was the CRSP/Compustat Merged Database, and was used to collect nearly all the data needed. This database contained all information from the Income Statement and Balance Sheet from all the desired companies. The downloaded document was in .csv format Where each column was either a data identifier(date of the statement, quarter, year, tic of the company, etc) or a specific financial stat of the company(Revenue, Cost of goods sold, Long term debt, etc) and each row was associated with a specific record, identified by the data identifiers. The dataframe obtained from this *.csv* file had 32020 rows and 376 columns.

3.2. Scraper Module

The next step is to download all the data needed from the multiple databases. This process is implemented in the *Scraper* module. This module is divided into two parts, one scrapes the *Macrotrends* database and the other scrapes Yahoo Finance. Firstly, the former, focuses on scraping the financial data that weren't already in the data retrieved from the *CRSP/Compustat* merged database, such as Cash Flow from operating and from investing activities, and the Free Cash Flow(this financial stat, came as a Per-Share value, so the number of shares outstanding had to be retrieved as well so as to calculate the absolute value of Free Cash Flow). Secondly, the *Yahoo Finance* scraper went on to retrieve the price and the dividends from the Yahoo Finance's Historical Data. The price retrieval is a very straight forward process. Through the python library *yfinance*, and by defining an interval of time, and the tic of a specific stock, one can get the historical prices of said stock. The next step is to retrieve the dividend history of a stock from *Yahoo Finance* as well. This part is not as straightforward as getting a stock's price. The initial part is very similar to getting the price data, one inserts the time period to which one wants to get the data from, and the tic of the company, and *yfinance* gets you the dividend history. Although, as you download dividends for each of the target companies, one needs to check if there are any stock splits during the time period where one is getting the dividends. If there aren't any stock splits recorded in the selected time period, then the dividends retrieved from *Yahoo Finance* are returned like that, without need for any type of adjustment. On the contrary, if there is in fact a record of a stock split or more for a certain stock, in the time period that was previously selected, then the process differs from the example before. Typically when a stock split occurs, the dividend also splits roughly in the same ratio than that of the stock split. This can easily cause a problem when

trying to calculate dividend streaks, so in order to reduce this problem, the dividend payments before the stock split occurred are divided by the ratio of the stock split in question. In the figure 2 bellow, there is a graphical representation of the process.

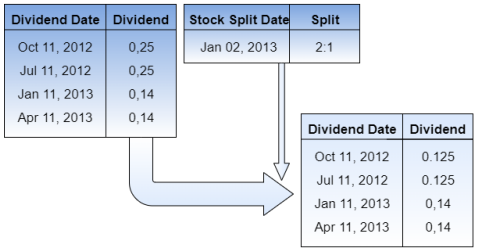

Figure 2: Graphical representation of how one preforms a data split correction for dividends

This module works semi-parallel to the Data Loading module, as it continuously send the scraped data to the Data Loading module in order to load the various scraped financial data into a single dataframe.

3.3. Ratio Generator Module

Afterwards, the *Ratio Generator* module exerts its functions of calculating the desired financial ratios, the dividend streak indicators and the labels later used for classification. In order to make it easier to follow the dividend streaks of the various companies at any given date, certain flags have been created. In order to know if a company is currently on a one, or three year streak of maintaining the dividends paid, or of increasing the dividends paid, these flags tell if a company, in the case of the flag of consecutive dividend increases, has a dividend paid higher than the one of the previous year, and consequently the one of the previous year being higher than the one of two years ago.

In respect to what the algorithm will try to predict(labels), what matters is whether some company in question, will continue its dividend streak(whether it is of dividend increase or maintenance), or if it will break it in the following year.

There´s a more succinct explanation below, with the aid of graphical representation, to better perceive this process. Since all companies are treated individually but equally, this explanation will be an example of what one would do for each of the companies in the present data.

Step 1: First of all, in order to calculate the dividend streaks, we just need to use the annual amount of dividends paid, since that will be the way to verify the companies' streaks. Then, for each instance (i.e. for each quarter/year pair), we check if in that quarter, the company in question has paid dividends.

Step 2: Afterwards, if the company has paid dividends, then the next step is to verify if this value is just an isolated value or if it comes from some kind of streak from previous years. In order to check what the present picture looks like, one compares the present value of the dividend with those of previous years.

Step 3: After assigning the corresponding flags for all the data entries, then the next step is, for all the instances that are already on a dividend streak, to check whether they will continue that same dividend streak in the following year or if they will break it. In order for the label to be set as **True**, the following has to happen: $Div_{year} < Div_{year+1}$ in case of checking for dividend increase, and $Div_{year} \leq Div_{year+1}$ in case of dividend maintenance.

3.4. Data Processing Module

After generating the financial ratios, all the data needed for the classification algorithms to use is now available. The last step left to complete the data handling section will take place in the *data processing* module. This module can be divided into two parts. A first pre-processing part, where the data will go through a series of filtering and elimination processes, and a second part where the data will be divided into several datasets (i.e. training, validation and test datasets).

The separation of the data can be seen in figure 3 for the validation phase. And in figure 4 when the model is going through the testing phase.

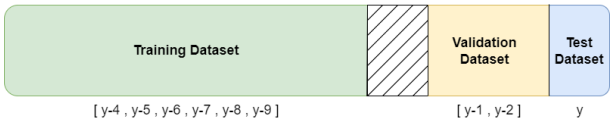

Figure 3: Separation of the Data with the *Walk-Forward* method - Validation Stage

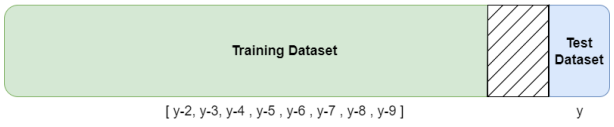

Figure 4: Separation of the Data with the *Walk-Forward* method - Testing Stage

Now that the datasets are completed, the next part is the classification phase. In this part, the data will be fed to the algorithms, and with some aid from the GA for validation purposes, the classification part of the project will handle all the data

analysis and predictions. This is the most crucial part of this project, has it will have the most impact on how well the system will preform.

The two algorithms used in this project will be XGBoost and SVM, and the way they are used is described bellow:

3.5. XGBoost Module

This module will implement the **XGBoost Classifier**. It receives the training dataset, the evaluation dataset and the hyper-parameters of the algorithm, builds the classifier and returns the data predictions and its probabilities.

The XGboost's hyperparameters serve the purpose of regulating the learning process in order to optimize the algorithm.

XGBoost is a very powerful machine learning algorithm, as such, it has a wide range of hyperparameters that can be adjusted. Since it is a decision tree algorithm, the hyperparameters will define certain tree characteristic metrics, such as the maximum depth of each tree, the amount of trees, certain decision metrics involved in the splitting of specific nodes, and so on.

As aforementioned, there is a variety of different hyprparaenters that can be used to tune this algorithm, and they can be broken into three different categories: **General Parameters**, **Booster Parameters** and **Learning Task Parameters**. Below in figure 5, there's the list of hyperparameters used in this project, with a brief description and its category.

Parameter Type	Parameter Name	Brief Description
General Parameters	nthread	Number of cores used for parallel processing
Booster Parameters	learning_rate	step size shrinkage used in update to prevent overfitting
	scale_pos_weight	Manages class imbalance. Helps the algorithm to converge
	min_child_weight	Minimum sum of weights required in a child node
	max_depth	Maximum tree depth
	gamma	Loss reduction needed to make a split on a leaf node of the tree
	subsample	Defines ratio of observations to be randomly sampled
	colsample_bytree	Defines the subsample ratio of columns for each tree
	lambda	L2 regularization of weights
	alpha	L1 regularization of weights
	n_estimators	Defines the number of weak learners
Learning Task Parameters	objective	Loss function to be minimized
	seed	Random number seed. Typically used for getting reproducible results

Figure 5: Table with all XGBoost parameters used in this project, and a brief description on each of them

3.6. SVM Module

In this model, the SVM algorithm will be implemented. As with the XGBoost module, described above, it will also receive the data already separated and ready for validating and testing the model.

The performance of this classification algorithm, as you would expect, depends heavily on the choice of its parameters. The parameters that will be used in this project for this algorithm, are the following, and can be seen in table 6.

Parameter	Description	Values Tested
Kernel	The type of kernel used in the algorithm	[RBF, Polynomial]
C	Regularization parameter for the algorithm	[1, 10, 100, 1000]
Gamma	Kernel coefficient	[0.1, 0.01, 0.001, 0.0001]
Degree	Degree of the polynomial kernel funcion(only applicable in polynomial kernel).	[2, 3, 4, 5]

Figure 6: SVM parameters used and considered values

When in the process of calculating the optimal parameters for the SVM model, a grid search approach was used. Firstly, the two kernel were separated as the polynomial has an extra parameter associated. Below, in the figures 7 and 8 are the results for the different kernels with various sets of parameters, when evaluating them according to the ROC-AUC evaluation metric, for the testing year 2019.

	$C = 10^0$	10^1	10^2	10^3
$\gamma = 10^{-1}$	0.612	0.605	0.599	0.597
$\gamma = 10^{-2}$	0.665	0.660	0.669	0.625
$\gamma = 10^{-3}$	0.674	**0.682**	0.668	0.648
$\gamma = 10^{-4}$	0.662	0.660	0.672	0.679

Figure 7: Results of a grid search preformed on SVM - RBF Kernel used

	n=2				n=3				n=4			
	$C = 10^0$	10^1	10^2	10^3	10^0	10^1	10^2	10^3	10^0	10^1	10^2	10^3
$\gamma = 10^{-1}$	0.616	0.622	0.597	0.609	0.605	0.562	0.540	0.540	0.613	0.602	0.585	0.585
$\gamma = 10^{-2}$	0.592	0.599	0.616	0.622	0.639	0.648	0.643	0.605	0.686	0.683	**0.689**	0.678
$\gamma = 10^{-3}$	0.506	0.538	0.592	0.599	0.609	0.611	0.670	0.639	0.494	0.484	0.595	0.675
$\gamma = 10^{-4}$	0.491	0.502	0.506	0.538	0.565	0.577	0.607	0.609	0.482	0.483	0.533	0.529

Figure 8: Results of a grid search preformed on SVM - Polynomial Kernel used

The best results for each of the kernels can be seen in bold in each of the tables above(7 and 8) and is 0.682 for the BRF kernel and 0.689 for the Polynomial kernel. Therefore the polynomial is going to be the one that is going to be used for future purposes, and with the following parameters : $n = 4, \quad C = 10^2, \quad \gamma = 10^{-2}$

First the algorithms have to go through a validation phase, this is where the algorithms analyze the data served as input over and over again in order to find the optimal values for their parameters. This is done through the use of a genetic algorithm, and grid search depending on the algorithm. The genetic algorithm will serve the purpose of finding the optial parameters for the XGBoost algorithm,

for each year that is needed. In the case of the SVM, a grid search is going to be used with the purpose of finding a set o parameters for the algorithm, that can apply to every year that is requires, as such, this grid search needs to be extensive and will be separated in two parts, one for each considered kernel(Polynoial and RBF).

Then, after the optimal parameters have been found, the algorithms are ready to make their predictions. Each algorithm will return their own predictions along with a probability assigned to each prediction.

3.7. Voting System

Afterwards, comes the voting phase. This is where the algorithms come to an agreement on which predictions to use. This is done through analyzing both the predictions and probabilities of each algorithm, and deciding on what should be done relative to the classification problem at hand. Every data entry is treated independently, and the voting system analyzes each model's predictions, and consequently saves the prediction of the model that has a higher probability value. An example of this process is represented below in figure 9 for 5 distinct data entries.

SVM		XGBoost		Voting System
Prediction	Probability	Prediction	Probability	Final Decision
1	0.945	1	0.863	1
1	0.899	0	0.736	1
1	0.986	0	0.993	0
0	0.763	0	0.837	0
0	0.803	1	0.798	0

Figure 9: Functioning of the implemented Voting System

3.8. Stock Ranking Module

After the voting, phase, a final prediction will now take place. This prediction will enter the ranking system and begin by evaluating all the companies, and according to the predictions, compare them all and finally assign a score to each. This score will be used later to make a ranking of the best companies in the dataset, in terms of future dividend payments. It will receive the predictions and probabilities coming from the algorithms or the voting system and merge them with the test dataset (in order to know if a certain company, in fact, will stop presenting dividend payments in the following year), and assign a rank to each company.

This rank is associated to the probabilities that the algorithms associated to each prediction, and as in this project, the data are of quarterly statements, the rank of a company will be relative to the four quarters of the year that is being evaluated. The way this happens is represented in the table below 10.

Stock 1		Stock 2		Stock 3		Stock 4	
Quarter	Score	Quarter	Score	Quarter	Score	Quarter	Score
Q1	0.932	Q1	0.959	Q1	0.932	Q1	0.846
Q2	0.930	Q2	0.906	Q2	0.899	Q2	0.903
Q3	0.990	Q3	0.924	Q3	0.851	Q3	0.912
Q4	0.903	Q4	0.927	Q4	0.947	Q4	0.899
Final Score = 0.939		Final Score = 0.929		Final Score = 0.907		Final Score = 0.890	
Ranking = 1st		Ranking = 2nd		Ranking = 3rd		Ranking = 4th	

Figure 10: An example on how the Stock Ranking module would preform

3.9. Sliding Window Module

Finally, the modules mentioned above in the classification part of the system, are all functioning inside the *Sliding Window* module, where the system repeats all the steps all the while increasing the years where the datasets are placed, along the entire functioning period. With the conclusion of this module, the system returns the stock rankings, in respect to the years inside this same period of time, therefore assigning a rank that takes into account all the years as to give a more complete solution.

4. Results

4.1. Case Study 1 - Single Algorithm System vs Voting System in Single Window

The first part of this case study is to analyze how each of the algorithms performs in respect to the metrics chosen and show the improvements of the voting system in comparison to just using each algorithm separately. The algorithms used, SVM and XGBoost had their parameters chosen with the aid of a grid search and a genetic algorithm respectively. Afterwards, with the set of parameters chosen for each, they went through a series of performance metrics such as *F-score*, *PR-AUC* and *ROC-AUC*.

A comparison between both algorithms performance is needed and both of their ROC curves can be observed below in figures 11 and 12.

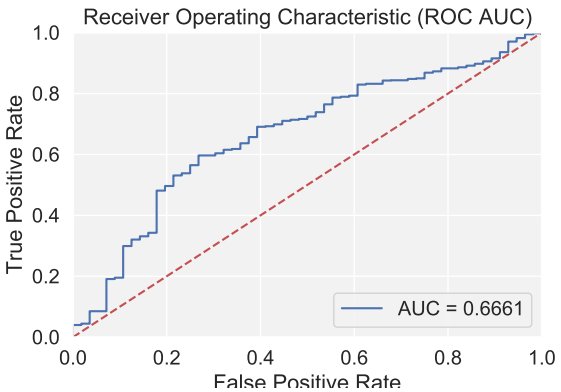

Figure 11: ROC-AUC Graphic of SVM predictions - Testing Year = 2017

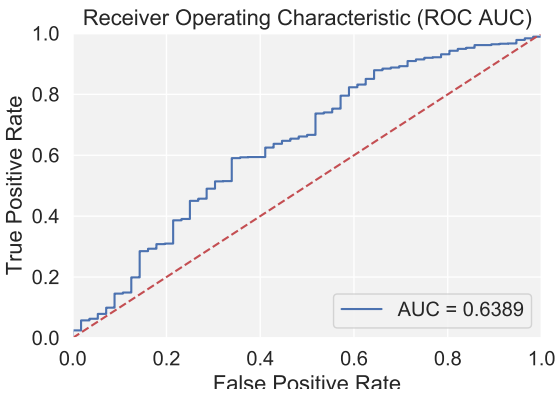

Figure 12: ROC-AUC Graphic of XGB predictions - Testing Year = 2017

One can already observe here which algorithm showed better results in the chosen testing year, and in this case was the SVM, as it presented a better ROC-AUC value than XGBoost. The testing year was chosen randomly from a selection of years that on which the tests were performed on the algorithms. In this year, the SVM might have better results, but in a different year, XGBoost could have a better performance, as the data that defines the performance of both algorithms varies for each year. This is the main reason why the voting system was implemented, as it can get the best predictions from both algorithms and create a better predictor. The resulting predictions from the voting system, generated as well a ROC curve that can be observed below in figure 13

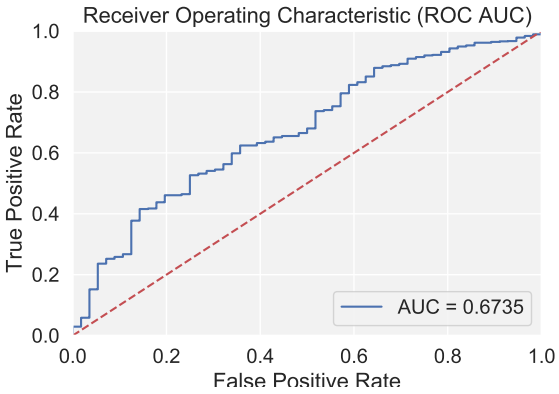

Figure 13: ROC-AUC Graphic of the Voting System predictions - Testing Year = 2017

As we can see from the figure 13, the ROC curve resulting from the voting system, although it may not show as good an improvement as expected, it still outperforms the algorithms when they try to predict the results independently, which is a good result overall.

A more complete demonstration of the results is shown bellow:

- **SVM -** [F-score = 0.9708 ; ROC-AUC = 0.6661; PR-AUC = 0.9725]
- **XGB -** [F-score = 0.9635 ; ROC-AUC = 0.6389; PR-AUC = 0.9678]
- **Voting System -** [F-score = 0.9721 ; ROC-AUC = 0.6735; PR-AUC = 0.9741]

It is apparent that the voting system present demonstrates superior results in all the metrics chosen. Both in F-Score and PR-AUC, the algorithm has a better performance which means that it has an easier time recognizing when an observation is actually positive or negative, thus having fewer false positives and false negatives, which is quite important in the problem presented.

4.2. Case Study 2 - Stock Ranking with Voting System

In this case study, we will evaluate the companies present in the dataset, and use the stock ranking module, in order to get a better understanding of which are the best companies in terms of future continuation of the dividend payout increase. The metric used to better optimize the parameters of the XGBoost was ROC-AUC, and this was the most regularly used metric in this project, as it always showed better results than the others. As to the SVM parameters, the same ones are used in all the studies performed, seeing that the grid search used to calculate the optimal set of parameters has been validated and trained in many different situations, already with the aim of being used in the following processes. The main part of this case study is to find the best ranked companies, as such, the information obtained by the system served to construct a bar chart of the results from the stock ranking process.

The main part of this case study is to find the best ranked companies. After this process, the information obtained by the system served to construct a bar chart so as to get a better understanding of how the results would vary for these companies and whether any would in the future leave their dividend streak.

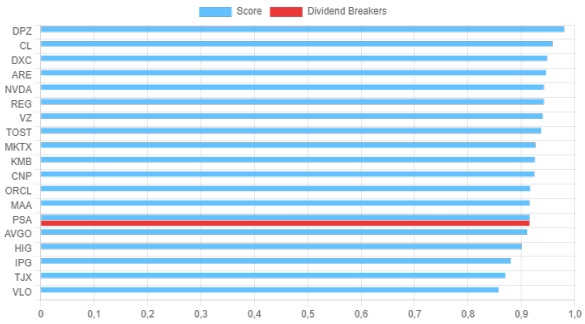

Figure 14: Single Window - Top 20 Stocks with corresponding Ranking - Stocks that brake their dividend the following year are signaled with a red bar

The results did not deviate much from what was expected, as the companies do indeed score high and there is not much variation, which would be expected from the companies placed at the top of the rank, given the percentage it represents in the total number of companies processed and analyzed. There is, however, one company (PSA) that will break its dividend streak the following year, which should not happen to one that is so high in the total ranking.

Even so, considering the percentage it represents, one company does not carry that much weight in the total system analysis. However, an analysis of the financial data of this company is a process that should be carried out in order to try to understand why this outlier was present in the section of the top ranking stocks analyzed.

It was verified that, even though PSA has broken its dividend increase streak, it has maintained its dividend payments until now without ever reducing or completely withdrawing them, which means that the red flag that it appeared to be is not as serious as it could be. However, it still means that the system implemented can be improved, and way to do so is to perform a more through analysis on these red flags, try to understand why the algorithm decided the way it did, and finally, attempt to correct these mistakes.

The next companies that are going to be analyzed are the ones on the bottom of the ranking system. The voting system has classified these companies in this way, in order to show that investments made in these companies are not so reliable, even if they continue to pay dividends at the next observation. It means that one, or several, financial ratios are at values that the system associates with companies that will break their streak of dividend payments.

Below, in figure 15, is the graph representing these 20 companies, and as one might expect, more than one will break its streak at the next observation.

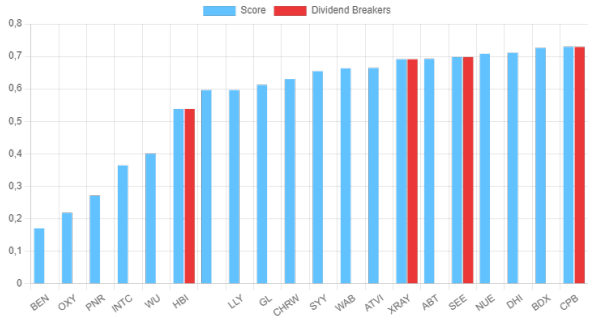

Figure 15: Single Window - Bottom 20 Stocks with corresponding Ranking - Stocks that brake their dividend the following year are signaled with a red bar

After gathering the top 20 stocks, the next step is to analyze whether investing in those stocks is a good investment or not. In order to evaluate how good an investment is, the ROI(Return on Investment) is going to be calculated for a portfolio with all those companies, in the year corresponding to the one used in testing. Additionally, not only will the ROI for the entire year be calculated as the one for a 6 month period, in order to perceive the evolution of the ROI.

In order to evaluate this investment the ROI will be compared to the one of the S&P500 for the same year, to understand if the voting system is a viable option, or if it is redundant because it shows worse results than that of S&P500 investing. The ROI values for both investment strategies are represented in the figure 16 below.

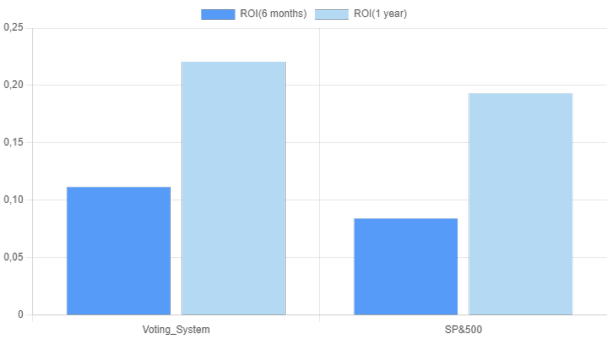

Figure 16: Single Window - ROI comparison for the Voting System strategy vs Classical S&P500 typical investment strategy

As one can see, the voting system investment strategy, outperforms the classical S&P500 investment. This is a positive result as it shows the viability of the voting system.

In the next case study, the voting system will be tested in a sliding window, in order to see if it can improve its current performance.

4.3. Case Study 3 - Sliding Window Stock Ranking and Performance Analysis

In this last case study, the voting system will again make its predictions, thus building a ranking of the companies. However, now, instead of just doing it once, it will do it several times sequentially, within a sliding window. This method is described in section **??**, and will consist in repeating the previous process, adding in each iteration a new ranking to the previous ones, always calculating the average of the rankings according to a SMA(Simple Moving Average).

The sliding window used in this project is in respect to three evaluation years([2017, 2018, 2019]). The metrics used to optimize the parameters of the XGBoost algorithm was again the ROC-AUC, as it appeared to lead the algorithm in a better direction, performance-wise, and again the SVM parameters will stay as in the previous examples.

This method, was proposed, as it should turn the predictions into a more reliable source for future investment strategies. In a sliding window The system as it continuously runs the modules needed to build the rankings, the worst ranked stocks will begin to leave the ranking it self in order to discard any stock, that at any point in time appeared to be inconstant and unreliable.

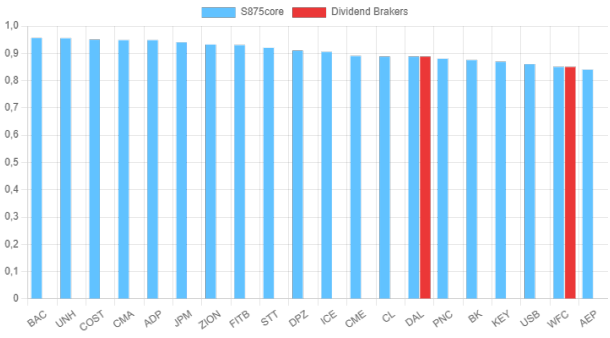

Figure 17: Sliding Window - Top 20 Stocks with corresponding Ranking - Stocks that brake their dividend the following year are signaled with a red bar

The resulting stock ranking doesn't appear to be what it should. There are two of the top 20 best ranking stocks that will in fact break their dividend streak, therefore, they shouldn't be here in the top 20 best stocks. The sliding window was supposed to show more reliable results, but this graph shows exactly the opposite. This graphic should've looked better than the one in figure 14, but as one can observe, it doesn't.

As one would except, accordingly to the graphic of this sliding window module, in figure 18, there is evidently some major downside to using this system, something that wasn't at all expected. In this case study, the S&P500 investing strategy outperforms the voting system with a sliding window, therefore, the system implemented cant be considered as a major investing tool, as one could just as simply invest in the S&P500, and get better results, with a smaller risk.

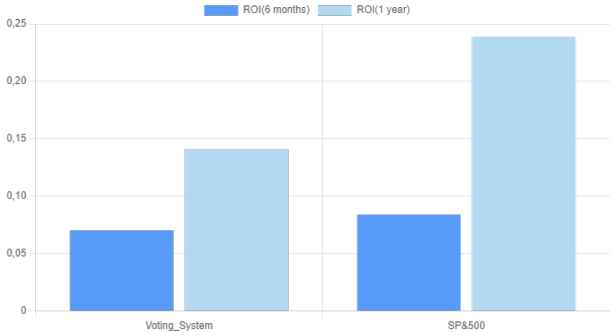

Figure 18: Sliding Window - ROI comparison for the Voting System strategy vs Classical S&P500 typical investment strategy

This case study didn't go as well as it should've. There are some improvements that need to be done in order for this system to start showing some better results. improvements will be discussed in the section 5, but as of now, this system isn't a very viable option.

5. Conclusion and Future Works

5.1. Conclusion

The final system, presents a voting system composed of the SVM and XGBoost algorithms, where parameter optimization is done through a Genetic Algorithm and a Grid Search respectively. This system was mainly tested in the years 2017, 2018 and 2019 and trained/validated in previous years, since 2008 onward. The algorithms received datasets composed of the financial data of the companies that make up the S&P500, which was used to train them, so as to create a model that will predict continued increases in dividend payments.

The algorithms also took part in a voting system that provided the system's final predictions. This voting system that was suggested, had both algorithms make their predictions and decide which one to use for the final prediction of the system.

The genetic algorithm used to optimize the XGBoost parameters showed a very good ability to find parameters that maximized the performance of the algorithm for several different testing years. Additionally, the grid search method showed some good

results, but still seems to be an aspect that can be improved.

5.2. Future Work

The implemented voting system showed improvements over using algorithms individually, which was the expected result back in the beginning of this project. However, the sliding window module fell short of its expectations as the system did not behave as expected. A system such as this can always improve, as so, some improvement ideas are shown bellow.

- The first improvement that could be implemented is bringing a GA approach into the SVM's parameters optimization. The grid search would still be used, with a short list of values for each parameter, and sequentially, a GA could be applied in order to search for a more optimal set of parameters in the feature space around the parameters given by the grid search.

- A fairly obvious improvement that could be done to this system is adding more different algorithms to the voting system. The current voting system already showed some improvements, and adding a few more, could only help the system adapt, and make better predictions over time. An algorithm that was considered for this system but ended up not being used is the Random Forrest.

- Finally, another good improvement would be to add more labels to the dataset. this way, one could get a better understanding on a few more important events by attempting to predict them. This could help get more information on how the companies would behave in the future, therefore, increasing the user´s financial knowledge on the financial market, and make better investment decisions in the future.

References

[1] F. Allen and R. Michaely. Chapter 25 dividend policy. In *Finance*, volume 9 of *Handbooks in Operations Research and Management Science*, pages 793–837. Elsevier, 1995.

[2] J. K. Bae. Forecasting decisions on dividend policy of south korea companies listed in the korea exchange market based on support vector machines. *J. Convergence Inf. Technol.*, 5(8):186–194, 2010.

[3] M. Benlemlih. Corporate social responsibility and dividend policy. *Research in International Business and Finance*, 47:114 – 138, 2019.

[4] M. Berre. *Investing in dividend growth stocks: analysis of portfolio performance using asset pricing models*. PhD thesis, 2021.

[5] H. DeAngelo, L. DeAngelo, and D. J. Skinner. Are dividends disappearing? dividend concentration and the consolidation of earnings. *Journal of Financial Economics*, 72(3):425–456, 2004.

[6] Y. Ding, X. Song, and Y. Zen. Forecasting financial condition of chinese listed companies based on support vector machine. *Expert Systems with Applications*, 34(4):3081–3089, 2008.

[7] E. F. Fama and K. R. French. Disappearing dividends: changing firm characteristics or lower propensity to pay? *Journal of Financial Economics*, 60(1):3–43, 2001.

[8] S. Han and R.-C. Chen. Using svm with financial statement analysis for prediction of stocks. *Communications of the IIMA*, 7(4):8, 2007.

[9] Y.-P. Huang and M.-F. Yen. A new perspective of performance comparison among machine learning algorithms for financial distress prediction. *Applied Soft Computing*, 83:105663, 2019.

[10] D. Javakhadze, S. P. Ferris, and N. Sen. An international analysis of dividend smoothing. *Journal of Corporate Finance*, 29:200 – 220, 2014.

[11] S. Ozlem and O. Tan. Predicting dividend payout policy of turkish firms using xgboost and mlnn algorithms. In *8th. INTERNATIONAL MANAGEMENT INFORMATION SYSTEMS CONFERENCE*, 2021.

[12] L. Shimin, X. Ke, Y. Huang, and S. Xinye. An xgboost based system for financial fraud detection. In *E3S Web of Conferences*, volume 214, page 02042. EDP Sciences, 2020.