Selection of Sustainable Dividend Stocks Combining XGBoost with Genetic Algorithm

Francisco José Couto da Silva

Thesis to obtain the Master of Science Degree in Electrical and Computer Engineering

Supervisor: Professor Rui Fuentecilla Maia Ferreira Neves

Examination Committee

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

January 2021
Declaration

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

To professor Rui Neves, who made this dissertation possible and was always available to guide me through the crossroads. His feedback and understanding on the subject made me feel confident and encouraged throughout these months.

To all my friends, who have constantly showed their unconditional support. I hope I can share many more happy moments with you all.

To my grandparents and parents, who always seem to find the right words at the right times. I can’t thank you enough for the trust and confidence you put in me, and I owe much of what I’ve accomplished during these past 5 years to you.
Abstract

This work presents an approach which combines Extreme Gradient Boosting (XGBoost) with Genetic Algorithm (GA) to create a novel dividend stock selection system, capable of identifying the safest dividend growth stocks and avoid dividend cuts, using real regularly updated financial data of firms with ongoing dividend streaks from the S&P500. The proposed implementation defines new training windows yearly, on which XGBoost learns to classify how likely it is that a company ends their dividend streak in the space of one year, based on quarterly observations. The probability scores are then used to rank the companies under evaluation. For each window, the GA is used to find the best set of hyperparameters, based on the performance on a defined validation period of observations, in terms of ROC and PR AUC scoring. The system is able to yearly update the rankings and stock selections by generating new models and combining the classification scores with prior ones, in a sliding window fashion. The results were evaluated by analyzing the performance of the top ranking stocks in the year following scoring, using one or several combined models to generate scores. In this last case, an elitism parameter is introduced to the system, reducing the number of stocks returned by dropping the worst ranked stocks every year. For 2019, the system was able to select as much as one half of S&P500 stocks defined as having ongoing dividend streaks, while avoiding more than 80% of the companies with unsustainable dividends. Furthermore, the top performing stocks consistently generated annual total returns that outperformed those of the S&P500.

Keywords

Dividend Investing; Dividend Policy; XGBoost; Genetic Algorithm; Fundamental Analysis; S&P500.
Resumo

Esta tese introduz uma abordagem que combina Extreme Gradient Boosting (XGBoost) com o Algoritmo Genético num sistema com o objectivo de selecionar ações geradoras de dividendos do índice S&P500, identificando aquelas com maior sustentabilidade no crescimento dos dividendos regulares e evitando aquelas cujos pagamentos podem ser interrompidos a curto prazo. A implementação proposta define anualmente janelas de treino-teste, em que o algoritmo XGBoost é utilizado para aprender e classificar se empresas na amostra irão manter o aumento dos dividendos no ano seguinte, com base em observações financeiras trimestrais. Em cada janela, o algoritmo usa o modelo para classificar um ano de observações, usando as probabilidades geradas para ordenar as empresas em termos de sustentabilidade. O Algoritmo Genético é usado em cada período como método de optimização dos parâmetros do XGBoost, de acordo com a performance num período de validação, com base nas métricas ROC e PR AUC. Os rankings gerados são atualizados anualmente gerando novos modelos e combinando as probabilidades geradas com as anteriores. Os resultados foram avaliados analisando a performance das ações escolhidas no ano seguinte aos rankings serem gerados, usando um ou múltiplos modelos para ordenar e selecionar ações. Neste último caso, foram adicionalmente introduzidos parâmetros de elitismo, eliminando iterativamente as piores ações de cada ano de modo a reduzir o número de ações retornadas. Para os períodos de teste, os melhores resultados mostram que o sistema é capaz de escolher até metade das ações de dividendos listadas, evitando acima de 80% daquelas com dividendos insustentáveis. Adicionalmente, as ações mais pontuadas geraram consistentemente retornos totais superiores aos do índice S&P500.

Palavras Chave

Dividendos; Política de Dividendos; XGBoost; Algoritmo Genético; Análise Fundamental; S&P500.
# Contents

1 Introduction  
1.1 Motivation .......................................................... 3  
1.2 Objectives .......................................................... 3  
1.3 Contributions ....................................................... 4  
1.4 Thesis Outline ...................................................... 5  

2 Background and Related Works  
2.1 Stock Markets ....................................................... 7  
2.2 Dividends ........................................................... 8  
2.2.1 Dividend Policy .................................................. 9  
2.2.2 Why Do Companies Pay Dividends? ......................... 11  
2.2.3 Dividend Raisers ............................................... 11  
2.2.4 Dividend Investing ............................................ 14  
2.3 Market Analysis ................................................... 16  
2.3.1 Fundamental Analysis vs Technical Analysis ............. 16  
2.3.2 Financial Statements ......................................... 17  
2.3.3 Financial Ratios ............................................... 18  
2.4 Machine Learning ................................................. 19  
2.4.1 Supervised Learning Algorithms ............................ 19  
2.4.2 Extreme Gradient Boosting .................................. 21  
2.5 Genetic Algorithm ................................................ 24  
2.6 Related Works .................................................... 26  
2.6.1 Works on Dividend Policy ................................... 27  
2.6.2 Works Dividend Policy Prediction and Machine Learning ...................... 28  

3 Implementation .................................................... 32  
3.1 Architecture ........................................................ 33  
3.2 Data Loading Module ............................................. 35  
3.3 Market Scraper Module ............................................ 37
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4 Ratio Generation Module</td>
<td>38</td>
</tr>
<tr>
<td>3.5 Data Update Module</td>
<td>41</td>
</tr>
<tr>
<td>3.6 Preprocess Module</td>
<td>41</td>
</tr>
<tr>
<td>3.7 XGBoost Module</td>
<td>44</td>
</tr>
<tr>
<td>3.8 Genetic Algorithm Module</td>
<td>46</td>
</tr>
<tr>
<td>3.9 Stock Ranking Module</td>
<td>49</td>
</tr>
<tr>
<td>3.10 Sliding Window Module</td>
<td>49</td>
</tr>
<tr>
<td>3.11 User Interface Module</td>
<td>52</td>
</tr>
<tr>
<td>4 Results</td>
<td>53</td>
</tr>
<tr>
<td>4.1 Data Overview</td>
<td>54</td>
</tr>
<tr>
<td>4.2 Single Window Dividend Increase Stock Ranking</td>
<td>57</td>
</tr>
<tr>
<td>4.3 Sliding Window Dividend Increase Stock Ranking</td>
<td>62</td>
</tr>
<tr>
<td>4.4 Sliding Window Dividend Maintenance Stock Ranking</td>
<td>69</td>
</tr>
<tr>
<td>4.5 Summary and Final Remarks</td>
<td>71</td>
</tr>
<tr>
<td>5 Conclusion</td>
<td>75</td>
</tr>
<tr>
<td>5.1 Future Work</td>
<td>77</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Dividend payout date ordering</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Dividend payout ratio</td>
<td>10</td>
</tr>
<tr>
<td>2.3</td>
<td>Dividend payout history of S&amp;P500 Dividend Aristocrats</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>Dividend payout history of S&amp;P500 dividend streak breakers</td>
<td>14</td>
</tr>
<tr>
<td>2.5</td>
<td>Dividend yield value trap</td>
<td>15</td>
</tr>
<tr>
<td>2.6</td>
<td>General boosting ensemble training process</td>
<td>22</td>
</tr>
<tr>
<td>2.7</td>
<td>Genetic algorithm gene encoding</td>
<td>25</td>
</tr>
<tr>
<td>2.8</td>
<td>GA crossover techniques</td>
<td>26</td>
</tr>
<tr>
<td>3.1</td>
<td>System architecture</td>
<td>33</td>
</tr>
<tr>
<td>3.2</td>
<td>CRSP/Compustat dataset sample</td>
<td>36</td>
</tr>
<tr>
<td>3.3</td>
<td>Dividend split adjustment</td>
<td>38</td>
</tr>
<tr>
<td>3.4</td>
<td>Sample dividend data returned by market scraper module.</td>
<td>38</td>
</tr>
<tr>
<td>3.5</td>
<td>Data update module flowchart</td>
<td>42</td>
</tr>
<tr>
<td>3.6</td>
<td>Data partitioning of a single training window</td>
<td>44</td>
</tr>
<tr>
<td>3.7</td>
<td>GA Flowchart</td>
<td>47</td>
</tr>
<tr>
<td>3.8</td>
<td>Encoding of hyperparameters in Genetic Algorithm</td>
<td>47</td>
</tr>
<tr>
<td>3.9</td>
<td>ROC and P-R AUC curves and score interpretation</td>
<td>48</td>
</tr>
<tr>
<td>3.10</td>
<td>Example of stock ranking</td>
<td>50</td>
</tr>
<tr>
<td>3.11</td>
<td>Sliding window module and training window definition</td>
<td>50</td>
</tr>
<tr>
<td>3.12</td>
<td>Example of sliding window iterative process of score combination</td>
<td>52</td>
</tr>
<tr>
<td>4.1</td>
<td>Yearly distribution of dividend streak interruptions</td>
<td>55</td>
</tr>
<tr>
<td>4.2</td>
<td>Top and bottom ranking firms (single training window, ROC AUC GA objective)</td>
<td>57</td>
</tr>
<tr>
<td>4.3</td>
<td>Top and bottom ranking firms (single training window, Precision-Recall Area Under Curve (AUC) GA objective)</td>
<td>58</td>
</tr>
<tr>
<td>4.4</td>
<td>Cumulative distribution of dividend streak breakers in rankings (single training window)</td>
<td>58</td>
</tr>
</tbody>
</table>
4.5 Returns of top ranking firms (single training window) ................. 60
4.6 Feature importance scores in terms of weight and average gain ................. 62
4.7 Sliding window returns \((p = 1)\) .......................................................... 67
4.8 Selected stocks and respective scores for \(p = 0.1\) and \(p = 0.2\) ................. 68
4.9 Comparison between single window and sliding window rankings, in terms of cumulative distribution of dividend streak breakers in rankings ........................................ 68
4.10 Sliding window returns, with dividend maintenance as target label ................. 73
4.11 Selected stocks and respective scores with dividend maintenance as target label, for \(p = 0.1\) and \(p = 0.2\) .......................................................... 74
List of Tables

2.1 Dividend streak indexes .................................................. 12
2.2 Fundamental vs technical analysis ...................................... 17
2.3 Summary of dividend policy state-of-the-art studies ................. 30
2.4 Summary of state-of-the-art studies in machine learning and predictive techniques applied to dividends .................................................. 31

3.1 Features generated by the ratio generation module .................... 40
3.2 XGBoost tree boosting parameters ........................................ 45

4.1 Grouped summary statistics of dividend increasing and cutting firms .................. 56
4.2 Results of sliding window for the three evaluation years \( p = 1 \) .................. 63
4.3 Final sliding window results, tested with a range of \( p \) thresholds ............ 65
4.4 Results of sliding window for three evaluation years, with dividend maintenance as target label \( p = 1 \) ........................................ 70
4.5 Sliding window results, with dividend maintenance as target label and for a range of \( p \) values 71
# Acronyms & Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>ANN</td>
<td>Artificial Neural Networks</td>
</tr>
<tr>
<td>AUC</td>
<td>Area Under Curve</td>
</tr>
<tr>
<td>CART</td>
<td>Classification and Regression Trees</td>
</tr>
<tr>
<td>CHAID</td>
<td>Chi-squared Automatic Interaction Detection</td>
</tr>
<tr>
<td>CRSP</td>
<td>Center for Research in Security Prices</td>
</tr>
<tr>
<td>DEAP</td>
<td>Distributed Evolutionary Algorithms in Python</td>
</tr>
<tr>
<td>DER</td>
<td>Debt-to-Equity Ratio</td>
</tr>
<tr>
<td>DJIA</td>
<td>Dow Jones Industrial Average</td>
</tr>
<tr>
<td>DPR</td>
<td>Dividend Payout Ratio</td>
</tr>
<tr>
<td>EBIT</td>
<td>Earnings Before Interest and Taxes</td>
</tr>
<tr>
<td>EBITDA</td>
<td>Earnings Before Interest, Taxes, Depreciation and Amortization</td>
</tr>
<tr>
<td>FCF</td>
<td>Free Cash Flow</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>GAAP</td>
<td>Generally Accepted Accounting Principles</td>
</tr>
<tr>
<td>GAKR</td>
<td>Genetic Algorithm Knowledge Refinement</td>
</tr>
<tr>
<td>GBM</td>
<td>Gradient Boosting Machine</td>
</tr>
<tr>
<td>GICS</td>
<td>Global Industry Classification Standard</td>
</tr>
<tr>
<td>KI</td>
<td>Knowledge Integration</td>
</tr>
<tr>
<td>LDA</td>
<td>Linear Discriminant Analysis</td>
</tr>
<tr>
<td>LTD</td>
<td>Long Term Debt</td>
</tr>
<tr>
<td>MBR</td>
<td>Market to Book Ratio</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Squared Error</td>
</tr>
<tr>
<td>NASDAQ</td>
<td>National Association of Securities Dealers Automated Quotations</td>
</tr>
<tr>
<td>NOA</td>
<td>Net Operating Assets</td>
</tr>
<tr>
<td>NYSE</td>
<td>New York Stock Exchange</td>
</tr>
<tr>
<td>REIT</td>
<td>Real Estate Investment Trusts</td>
</tr>
<tr>
<td>ROA</td>
<td>Return on Assets</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristic</td>
</tr>
<tr>
<td>ROE</td>
<td>Return on Equity</td>
</tr>
<tr>
<td>ROI</td>
<td>Return on Investment</td>
</tr>
<tr>
<td>ROIC</td>
<td>Return on Invested Capital</td>
</tr>
<tr>
<td>SDD</td>
<td>Special Dividend Distribution</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
</tr>
<tr>
<td>TTM</td>
<td>Trailing Twelve Months</td>
</tr>
</tbody>
</table>
Introduction

Contents

1.1 Motivation ................................................................. 3
1.2 Objectives ................................................................. 3
1.3 Contributions .............................................................. 4
1.4 Thesis Outline .............................................................. 5
In the world of financial markets, there are a large number of popular strategies investors use to generate profit. These strategies are adopted and used by different types of investors, according to their preferences in terms of risk tolerance, return timeframe and available funds. Among the most notorious investment strategies are growth, value and income investing.

Growth strategies revolve around purchasing more volatile and often overpriced stocks, with high future growth perspectives, in hopes of profiting of said growth. On the other hand, value oriented approaches are long term strategies which consist on buying stocks with strong fundamentals, in theory underpriced, so that when the market catches up with real value the investor can profit on this adjustment. Lastly, income investing relies on building investment portfolios which generate regular streams of income, suited for investors looking to accumulate wealth over time. Income sources include bonds or dividend paying stocks – stocks which generate regular income to their shareholders.

In particular, dividend stocks are one of the most widely used and recognized sources of passive income by investors. Companies which pay dividends on their stock typically do so in a regular fashion and tend to increase the payouts at a steady rate. While dividend investing is associated with income investing, it is also associated with other traditional strategies that take a more value oriented approach. These are based on identifying and selecting underpriced stocks which pay regularly high-quality dividends, in hope that both the dividend appreciates over time along with the price. Other variations of this strategy focus on dividend growth investing, which simply look for stocks having strong outlooks for future dividend growth. In any of the cases, dividend investors strive to avoid at all costs firms stale and low quality cash dividends.

In recent years, with the increasing availability of financial data and the growth of learning based Artificial Intelligence (AI) algorithms, much work has been put into creating systems which use fundamental and technical analysis to help investors taking profit in financial markets according to the strategies taken, improving their decision process [1, 2]. Other predictive systems have been used as pre-emptive tools, such as in the wide bankruptcy prediction field, which help investors identify poor fundamental structures before-hand and minimize loss [3, 4].

In the same vein, this work proposes and implements a learning-based system using Extreme Gradient Boosting (XGBoost) combined with Genetic Algorithm (GA) optimization, with the main goal of aiding the stock selection process made by dividend stock investors. The proposed system uses regularly updated fundamental and market based historic data to yearly train XGBoost models to classify if dividend paying firms are likely to interrupt their dividend streak in the short-term. The GA is used to find the best hyperparameters for each model, based on validation period data prior to the evaluation year. XGBoost outputs classification scores to the financial observations generated by firms throughout the year, which are then used to rank the stocks in terms of sustainability and dividend growth safety. These models can be used independently or combined in a sliding window fashion, by using the moving average of scores
to iteratively update the rankings. The use of rankings allows investors to freely select the number of
top scoring dividend stocks, according to their portfolio diversification requirements, and to monitor and
update their selections according to the results generated each year.

1.1 Motivation

Dividend stocks are seen as relatively safe investment options by most investors, when compared to the
typical growth stocks. Even though there is some truth to the idea that dividend stocks are less volatile
and more conservative options, dividend investing is not as straightforward as it may look. This is mostly
due to the sheer diversity of dividend paying firms progressing through different financial stages, shifting
their dividend policies accordingly.

The key to the success of a dividend growth stock portfolio lies in the selection of dividend stocks,
in which the investor aims to pick the stocks that will keep growing their dividend at a sustainable rate
and avoiding those which in the near future may decrease or even omit their payouts. In particular,
these are among the most troublesome events for dividend investors since, in addition of reducing the
level of passive income, drive stock prices down – specially in companies with long histories of dividend
increases and maintenance. This selection process therefore involves a careful analysis of the financial
position and overall fundamentals of a firm – making the act of predicting less than safe dividends
non-trivial for investors. As a consequence, the main motivation for this work was to develop a system
capable of performing such the task of selecting dividend stocks, by ranking them in terms of future
sustainability.

The second main motivation for this work was the implementation of the system using state-of-
the-art gradient boosting algorithm XGBoost combined with GA optimization. This approach is largely
unexplored in the fields of financial markets, while no significant studies in this context were applied to
dividend policy.

1.2 Objectives

The main goals of this work are to:

- Develop several GA optimized XGBoost models to periodically score and rank dividend paying
  stocks from the S&P500 index;

- Use the rankings to identify, among companies with ongoing dividend streaks, those which are the
  most unsustainable and preemptively detect dividend cuts;

- Identify the main financial drivers and predictors of future dividend cuts;
• Study the effects of using different scoring methods for the GA and using a single or multiple training period windows to score the stocks;

• Assess both the capital and dividend returns of the selected stocks, benchmarking them with the overall S&P500 compound index returns for the same period of time.

### 1.3 Contributions

The main contributions proposed by this dissertation are the following:

• The combined application of XGBoost and Genetic Algorithm (GA) to the dividend policy prediction field. While XGBoost has notoriously achieved outstanding results in neighboring fields, such as bankruptcy prediction, no significant work has been done in the field of dividend policy and sustainability prediction;

• A novel ensemble-like approach for stock selection, based on multiple model classification scores to yearly select the safest dividend stocks and reduce the risk of experiencing dividend cuts. This approach stands out from the existing works on dividend policy prediction, creating a system solely focused on dividend safety from an investment standpoint;

• Identification of the most relevant financial factors in predicting dividend policy changes in the S&P 500 in the recent years.
1.4 Thesis Outline

This document is organized as follows:

• Chapter 2 goes over the necessary background on stock markets, dividend policy and fundamental analysis - key concepts used in the development and implementation of the system. This chapter also reviews relevant studies made in the fields of dividend policy, financial determinants of dividend payouts and the application of AI techniques in stock markets, with focus on the predictability of dividends

• Chapter 3 goes over the implementation of the system in detail. It first describes the high-level architecture, then detailing each of the building blocks and design choices

• Chapter 4 describes the results obtained by executing the system in different setups and case study scenarios. These results are also analyzed in terms of the performance of the selected top ranking stocks in the periods following scoring and compared between setups

• Chapter 5 offers a brief recap on this dissertation, a set concluding remarks and takeaways, as well as a list of future work suggestions.
2 Background and Related Works

Contents

2.1 Stock Markets .................................................. 7
2.2 Dividends ....................................................... 8
2.3 Market Analysis ................................................. 16
2.4 Machine Learning .............................................. 19
2.5 Genetic Algorithm ............................................. 24
2.6 Related Works .................................................. 26
This chapter details background knowledge to this dissertation. Sections 2.1 through 2.5 go over essential concepts on stock markets, dividend policy, fundamental analysis, Machine Learning (ML) and the GA. Lastly, Section 2.6 reviews relevant literature on dividend policy and on the application of ML techniques to dividend policy prediction.

2.1 Stock Markets

The stock market is where the selling, buying or trading of stocks occurs. These stocks are fraction units of ownership, issued by publicly traded companies, which allow private shareholders to take a position in the company and be entitled to a piece of the earnings. The organizations that allow the trading of listed stocks and set the foundations and regulations of stock markets, gathering buyers and sellers, are referred to as stock exchanges.

From a company standpoint, private firms turn to the stock market in order to raise capital to fund their activities and gain more exposure, which can build trust among the public. This serves as an attractive alternative to debt financing, where companies borrow money from investors, which has to be paid in a given time and at a given interest rate. On the other hand, after going public, as ownership is distributed, firms allow themselves to lose some degree of control to the public shareholders. Additionally, they are expected by the government and other entities to report their activity in standardized documents, with quarterly and yearly financial reports.

The price of stocks is subjected to the market forces, being speculated among investors and reflecting the sentiment they show towards each of those companies – a supply and demand mechanism. The trades are typically facilitated by a stockbroker, who aids the exchange between buyers and sellers. The supply and demand nature of stock markets makes it so that a stock with increased buyers and few sellers will naturally increase its trading price, where periods of high demand create uptrend markets, being the opposite also true, since when the supply exceeds the demand, prices tend to fall.

There is a large number of stock exchanges around the world, often categorized by the country in which they operate, typically with stocks from that same region. Stock exchanges are often evaluated in terms of capitalization, which is the sum of the individual market capitalization of the firms listed, given by the individual share price multiplied by the outstanding shares in the market. The largest stock exchanges, in terms of market capitalization, are located in the USA. Among the most notable examples of stock exchanges are:

- New York Stock Exchange (NYSE): the largest stock market and one of the largest financial markets in the world. It is a physical exchange market, with a trading floor located at 11 Wall Street. It implements a typical auction system, where buyers and sellers buy directly from one another;
- **National Association of Securities Dealers Automated Quotations (NASDAQ)**: the largest electronic stock exchange in the world, with second largest market capitalization, second to NYSE. The trades are done within a computer network, where buyers and sellers trade stocks through a dealer, contrary to the NYSE.

Stocks are often grouped in baskets by different market segments, such as company size or sector – these are defined as stock indexes. An index helps evaluate how a certain segment of the market is performing. Each stock index can be seen as a portfolio of the underlying stocks, each of them weighted by a specific value. The index value is therefore gauged by the weighted average value of these selections of stocks. The weighting strategy of the index can be based on market capitalization, price or fundamental factors.

The segments by which indices are grouped are referred to as the “coverage type” and they include exchanges, sectors or regions. Some notorious stock indices include:

- **S&P500**: Gathers 500 largest firms listed on USA stock exchanges, weighted by capitalization;

- **Dow Jones Industrial Average (DJIA)**: Gathers 30 largest firms listed on USA stock exchanges, weighted by price.

In the context of this work, the focus is put on public firms publicly traded within USA stock exchanges, in particular constituents of the S&P500 index.

### 2.2 Dividends

Dividends are periodical payments made by publicly listed firms to their shareholders, as accorded by the board of directors of the firms. These payments are typically made in cash and are used as means to distribute the company profits when they excel. Dividends are declared on a per-share basis, which means that the payment received by each shareholder is determined by the number of shares owned. They can also be paid in other forms besides cash, such as with stock or property dividends.

The distribution of dividends follows a particular order of events, which must be closely followed by the shareholders, as illustrated in Fig. 2.1:

- **Announcement Date**: also called the declaration date, is the date when the dividend payment is approved by the board of directors and announced to the shareholders. This announcement includes references to the amount and date of payment. Stock prices rise on the announcement date by the value of the dividend announced;

- **Ex-dividend Date**: the date after which the stock will start trading without the dividend, decreasing the stock price by the dividend value, as such. A shareholder that buys the stock on the ex-dividend date is no longer eligible to receive the dividend payout;
• **Record Date**: the date set by the company when the shareholder must be found in the company’s books in order to receive the payment. The ex-dividend date is set from one to three business days prior to the record date by the stock exchange, so that when a buying order is placed, the settlement period allows the shareholder to be in the company’s books in time for the record date;

• **Payment Date**: the date when the company sends the corresponding payment to the shareholders.

![Diagram of dividend payment dates]

**Figure 2.1**: Ordering of significant dividend payment dates, to be monitored by dividend investors.

The issuance of stocks, as described in Section 2.1, is made in the form of common stock or preferred stock. While not all companies issue preferred stock, their share owners are paid a prioritized fixed dividend, whereas with common stock, they are not guaranteed to receive any dividends. In literature, dividend payout studies focus on common stock dividends, which are dependant on the dividend policy of the firm [5]. As such, when stating that a firm does pay dividends, that is referring to common stock dividends.

### 2.2.1 Dividend Policy

The dividend policy is the set of decisions and rules taken by the board of directors with respect to dividend distribution. Dividend policy can also be defined in simpler terms, such as the particular decision to maintain, increase or decrease the dividend in a single year or in a horizon of years.

Companies opt for different policies, with different scheduling and rates of payout. They determine the amount of earnings surplus the company is willing to distribute to the shareholders, instead of retaining or reinvesting them into the growth of the business, under the retained earnings account. The percentage of earnings (net income) a company pays in the form of dividends is commonly called the Dividend Payout Ratio (DPR), represented in Fig. 2.2.
Even though dividend policies are company specific in terms of amount and rates of change, they often align with one of the three following groups:

- The **irregular policy** states that the company is in no obligation to pay dividends, so the board of directors decide to pay them after extraordinary events – for example, following a year of abnormal profits. This is a typical of companies with more volatile cash flows and earnings;

- The **stable policy** sets a fixed percentage out of the earnings to be distributed, periodically, as dividends. This means that the investor takes more risk, as the dividends will fluctuate according to the profits of the company;

- The **regular policy** is the most common among dividend payers and the one that aligns with this study. The companies following this policy focus on delivering a steady and yearly schedule of dividend payments which in principle do not strongly respond to quarterly declines in earnings, meaning they point at long term stability. Their dividend can be annual, semi-annual or, more commonly, a quarterly dividend. The companies that commit to this strategy increase their dividend at a steady rate and are therefore more reliable. One important caveat regarding regular dividend policy is that typically, after initiating regular dividend payouts, there is added pressure for the firm to not only keep paying them, but also increase them from time to time, which has to be sustained by the underlying firm performance.

It is also common for regular paying firms to issue special non-recurring distributions of dividends in addition to their regular dividend, showing that in some cases the policy groups described do indeed overlap.

Most research done in the field of dividends points out that the typical regular payers are large and
more established companies, transitioning from growth into a maturity phase [6]. In contrast, growth oriented companies or companies in more competitive sectors tend to shy away from dividends. This happens since these companies need to reinvest most of their profits due to high expenses, which consist on R&D, operational and expansion costs [7]. These companies could therefore be more inconsistent and even have years of loss, which does not allow them to commit to dividends.

2.2.2 Why Do Companies Pay Dividends?

In the context of this work, it is important to understand what drives companies to commit to dividend payments in the first place and to distribute the earnings to their shareholders, when they could simply retain them into growing or adding more value to the company.

One of the main reasons for which managers pay shareholders a cut of the company’s profit is simply to reward them for the ownership of the stock and therefore the trust they deposited in the company. This reward can be seen as an encouragement for the investor to invest in stable dividend paying companies, that may not show the same growth as other classes of non-paying firms, but still are highly profitable and have high quality earnings.

Another reasoning behind paying dividends is given by the signalling theory, which claims that firms that commit to regular dividends and increase them regularly send a positive signal to the market about future performance of the firm [5]. According to the theory, the bigger the increase, the larger the signal of confidence on future earnings increase from the board of directors, which on the other hand may prevent smaller firms of sending a similar signal of confidence. Although this theory has widespread popularity, studies concluded that traditional signalling theory does not experimentally translate well into future profitability of the firm [6, 8].

Finally, dividend payouts can be thought of as a way of reducing agency costs [5], which are the costs of monitoring the managers. The classical principal-agent problem – where the principal is the shareholder and the agent is the manager – arises when the interests of the shareholder and the manager do not overlap, which may occur, for example, when the managers are investing company funds in projects that the shareholder does not consider to be worth investing, or in general when they pursue personal interests. Paying out dividends can mitigate this problem, since it prevents asset structures, mainly high cash and low debt, that give the managers the power to make value reducing investments [9].

2.2.3 Dividend Raisers

This chapter relates to the class of regular dividend payers who follow a policy of increasing their dividend for consecutive years in a row. Such a trend of yearly increases maintained for a significant period of years is often described as a dividend streak in most literature. There are several stock indexes
available for dividend investors that track down the dividend growth history. Notoriously, the S&P500 Dividend Aristocrats gathers firms from the S&P500 index that have increased their dividend for the last consecutive 25 years, that is, with a 25 year streak [10], plus some added liquidity requirements. Aside from this index, there are others that measure the streaks of dividend increases, with different requirements, some of them presented on Table 2.1.

Table 2.1: Popular dividend streak indexes, which trace companies with consecutive annual dividend growth.

<table>
<thead>
<tr>
<th>Name</th>
<th>Maintenance</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dividend Aristocrats</td>
<td>Standard &amp; Poors</td>
<td>25 year streak, S&amp;P500 member, Liquidity requirements</td>
</tr>
<tr>
<td>Dividend Champions</td>
<td>DRiP</td>
<td>25 year streak</td>
</tr>
<tr>
<td>Dividend Achievers</td>
<td>Indxis</td>
<td>10-24 year streak; Liquidity requirements</td>
</tr>
<tr>
<td>Dividend Contenders</td>
<td>DRiP</td>
<td>10 to 24 year streak</td>
</tr>
<tr>
<td>Dividend Challengers</td>
<td>DRiP</td>
<td>5 to 9 year streak</td>
</tr>
</tbody>
</table>

Most of these companies in the 25+ streak group are blue chip established companies with a solid track of performance and strong competitive advantages. Specifically, the S&P500 Dividend Aristocrats index has historically outperformed the overall S&P500 market return, with lower volatility. In the last 10 years, the Aristocrats averaged a 14.74% return per year, against the 13.56% of the S&P500.

The majority of the companies which have established this trend for over decades will most likely keep the streak, until it is absolutely impossible to do so, which is highly unlikely – historically, for the Dividend Aristocrats, less than 10% of the companies get dropped per year [11]. As such, an unexpected shift in policy breaking the trend would be a major negative signal for these long time dividend raisers.

To illustrate these groups of companies, the dividend history of two notable constituents of the S&P500 Dividend Aristocrats is represented in Fig. 2.3, Walmart (WMT) and Procter & Gamble (PG) – both of which time-tested dividend stocks. WMT is often referred to as recession-proof, withstanding bear markets and increasing their dividend in unfavourable economic circumstances for more than 40 years. These competitive advantages are mostly due to their massive scale and low distribution costs which allow them to practice low prices and obtain large profitability and earnings, therefore sustaining their dividend. The other notable case is PG, on a 63 year dividend streak, the world’s largest consumer products company. It also enjoys a great deal of competitive advantages, specially in the brand building and distribution fields.
On the other hand, sometimes managers of these firms are reluctant about the current level of earnings being capable to support future payouts at the present rate [12], and so they cut their dividend payout and end their dividend streaks. Commonly, dividend cuts are regarded as a signal of financial distress approaching – some works establish that these events are the first state in a continuum of financial distress steps, being followed by defaults on loan payments and bankruptcy [13]. This plays to the conclusion that in the face of distress, companies first resort to dividend cuts.

Some notable cases of such firms include General Electric (GE) and Kraft Heinz (KHC), with dividends portrayed in Fig. 2.4, both regular dividend payers with years of consecutive dividend growth. GE made a dividend cut of 67% in 2009, during strong financial turmoil years. This followed a loss in competitive advantage over the recession years, which was reflected by poor balance sheet results that made the dividend cut rather predictable. After these years, GE returned to semi-regular dividend hikes up until 2018, when it announced that it would be again cutting the dividend, to fix excess leverage. In 2019, KHC also cut their quarterly dividend-per-share by 36%, after having been increasing payouts for
4 consecutive years. This dividend cut was said to be a part of deleveraging process which aimed to
give the company balance sheet strength. In addition, KHC had shown mediocre growth in revenues
dating back to the period that followed the merger between Kraft and Heinz, back in 2015.

![General Electric (NYSE: GE) Dividend History](image)

![Kraft Heinz (NYSE: KHC) Dividend History](image)

**Figure 2.4:** 20 year dividend payout history for regular dividend paying stocks which cut their dividend, corrected for stock splits and Special Dividend Distribution (SDD). Obtained from Yahoo Finance.

### 2.2.4 Dividend Investing

Buying and holding dividend paying stocks has been a popular strategy among income seeking investors
for a long time. This strategy focuses on a long term increasing streams of cash and relative low risk of
most of these dividend stocks.

Companies such as the ones described in Section 2.2.3 are highly popular between dividend growth
investors. As described, they are likely to continue their stream of payments, which allows these in-
vestors to compound their returns – meaning that the dividend received in each quarter is being actively
reinvested to buy more stock, which on the other hand increases the return of the portfolio.

According to [14], the pillars of modern dividend investing can be broken down in three:
- Paying a reasonably high dividend yield – annualized dividend divided per share price ratio, which estimates the proportion of returns on dividends from the investment;

- Having a reliable dividend paying strategy, that is, committing to regular dividends with yearly increases, excluding extraordinary dividends;

- Building a diversified portfolio in order to minimize exposure to unsystematic risk.

The dividend yield is in particular one of the first ratios looked at by dividend investors. Even though the bulk of companies in the 25+ year group are reliable and all around solid investment opportunities, it is important to keep in mind that their dividend yield are often not the best. Since companies in the Champions and Aristocrats group commonly yield less than 3% [11], which correspond to the higher maturity and slower growth rate of mature dividend programs, it is beneficial to also take in account companies with smaller streak periods, which correspond to newer dividend programs, often with more growth. One important note is that these companies are generally less safe, representing risky investments that can be value traps for investors. The value trap occurs in cases where the dividend yield is significantly higher than their industry average at the time, due to sudden price drops or policy shifts. These share price drops, coupled with high dividend payout ratios and weak fundamental trends in financial statements, may signal that although the yield is high at a given point in time, the company is not likely to sustain the payout increase, luring the investors into a dead end.

In order to illustrate these scenarios, Fig. 2.5 shows the relationship between the dividend yield and the stock price before the General Electric (GE) dividend cut. GE had a spike in yield prior to 2009, due to their price downfall, which was followed by a slash in dividend in 2009.

![Figure 2.5: Example of the relation between dividend yield and stock prices in periods prior to dividend cuts. GE cut their dividend after first quarter of 2009, after experiencing steady price downfall (Source: Yahoo Finance).](image)

In other cases, high yield periods may be potential entry point for investors, if the company has strong fundamental performance and is able to sustainably increase their dividend in the upcoming years. For
these reasons, from the investing standpoint, it is mandatory to evaluate the fundamentals to distinguish value traps – this is at core the sustainability issue of the present work, since in an ideal setup, we could be able to identify, out of these high yielding stocks, those which have the ability to meet their dividend commitment.

2.3 Market Analysis

Investing in company stock requires the investors to perform a market analysis, to find out which stocks will grant the best returns. One of the most widely used approaches is fundamental analysis.

In the fields of finance and accounting, measuring and evaluating the value of an asset, such as a stock, is done using fundamental analysis. This method of valuation is used to find if firms are over-performing or underperforming by analyzing financial statements to assess their intrinsic value. Fundamental analysis uses figures from these statements to produce relevant indicators that are then used to evaluate stocks. Individual firm fundamental analysis can be extended far beyond, to the industry sector or economic analysis, in a bottom-up or top-bottom fashion. Fundamental analysis is the preferred method of valuation for value driven and income investors, including dividend investors.

2.3.1 Fundamental Analysis vs Technical Analysis

Aside from fundamental analysis, the other popular method of valuation is technical analysis, which uses market based data such as volume and pricing to build indicators, assuming that all fundamentals are implicitly condensed onto the price. While the fundamental valuation of a firm can be out of sync with the current market price, fundamental analysts believe that the market will eventually catch up with the real value of a firm in the long run, which makes this valuation strategy beneficial for long-term investors. For stocks, this means that, rather than providing optimal entry and exit points for buying and selling, which is the approach taken by technical analysis, fundamental analysis focuses instead on understanding and forecasting the real value of the asset. This is both the main advantage and disadvantage of the method, since it is not applicable for short termed decisions in financial markets but offers a much more well founded valuation, which investors then use to profit on the eventual future market correction. Other large disadvantage of fundamentals is the accessibility to the data, since gathering accurate historical financial data from firms quarterly or yearly reports can be much more troublesome, when compared to market data. Table 2.2 shows the key differences between these two approaches.
Table 2.2: Comparative analysis of the differences between fundamental analysis and technical analysis.

<table>
<thead>
<tr>
<th></th>
<th>Fundamental Analysis</th>
<th>Technical Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Financial Statements and economic news</td>
<td>Historical stock prices</td>
</tr>
<tr>
<td>Used by</td>
<td>Value and long-term investors</td>
<td>Short-term driven investors</td>
</tr>
<tr>
<td>Time frame</td>
<td>Months or years</td>
<td>Mostly minutes and days</td>
</tr>
<tr>
<td>Strategy</td>
<td>Identify and buy strong stocks when undervalued</td>
<td>Find optimal entry and exit points</td>
</tr>
<tr>
<td></td>
<td>in the market</td>
<td></td>
</tr>
</tbody>
</table>

Since the objectives are focused on dividend sustainability and the fact that dividend investing is in itself a long term strategy, fundamental data will be used in this work.

2.3.2 Financial Statements

Financial statements are documents produced by managers regarding the performance and activity of their companies. These reports are key data sources to investors and the keystone to fundamental analysis, being typically released quarterly and annually.

The information included in these documents should be user readable, understandable and provide accurate information. In order to standardize reports made by different companies, their content must comply with the Generally Accepted Accounting Principles (GAAP), which define the accounting guidelines that must be followed. However, since they are written by company members who want to make them look attractive for investors and creditors, they are prone to financial engineering. For that reason, it’s important to be able to analyze them very thoroughly and carefully [15].

Financial statements come in three basic types:

- **Income Statement**: Provides information on how much cash the company earned during a set interval of time, typically a financial quarter or year. It starts from the revenue of the company over the corresponding operating period, which is subtracted from the cost-of-goods sold, resulting in the gross profit. The other class of expenses on the Income Statement are the operating expenses and depreciation costs. Taking out these figures, we get the operating profit, which is also called the Earnings Before Interest and Taxes (EBIT). This value is finally subtracted by interest expense and taxes, which result in the final net income (earnings) of the period. Another popular figure typically represented in this document are the Earnings Before Interest, Taxes, Depreciation and Amortization (EBITDA), obtained by adding back depreciation and amortization to the operating profit;

- **Balance Sheet**: Describes the assets and liabilities of the company in a fixed point in time. The
assets are the resources the company owns, which include items such as cash, property or equipment. They are divided in current and non-current assets, with respect to quickness they can be converted into cash. The second component of balance sheets are the liabilities, which represent the legal obligations and financial debt. They are divided in current and long-term liabilities, representing those coming due in the next year or longer, respectively. The total assets minus the total liabilities represent the net-worth of the company, known in accounting as the shareholders’ equity;

- **Cash Flow Statement**: Records the movements of cash in and out the company. As with the Income Statement, they are typically issued quarterly and yearly, accounting for a given period of time. They are composed of three sections: the cash flows from operating activities, cash flows from investment activities and cash flows from financing activities. Cash flows from operating activities are obtained by adding non-operating expenses to the total earnings. Cash flows from investing activities contain the gains and losses in investments. Lastly, cash flows from financing activities account for the amounts of cash in and out used to finance the business. Adding all the three items of the statement represents the total cash flow of the period.

### 2.3.3 Financial Ratios

Financial ratios are mathematical dependencies and relations between different items taken from raw quarterly or annual financial statements, described in Section 2.3. They are indicative of how a firm is performing in a given area, allowing for comparisons with other companies, inside industry sectors and between time periods.

Some of these ratios are major analysis points for investors when picking stocks, particularly for evaluating the sustainability of dividends. The main financial metrics to be used in this work will be further described in the following sections. These ratios are grouped in literature by different classes according to the performance measure they evaluate:

- **Profitability Ratios**: assess the profit generation capabilities of a firm based relatively to other financial metrics. They are commonly divided into margin ratios or return ratios. Margin Ratios are used to evaluate how well the company turns sales into profits and the overall capacity on money making. On the other hand, return ratios focus on the generation of return to shareholders, focusing on how well a company can convert assets, equity or capital into profit;

- **Liquidity Ratios**: measure the ability for a company to meet their short-term commitments without raising external capital, as well as their ability to avoid insolvency [15]. As the current and long term
liabilities come due, these ratios evaluate how cash reserves and asset liquidity can be used to fulfill those obligations – hence measuring the ability for internally raising cash from assets;

• **Leverage Ratios**: measure degrees of commitment, in terms of debt, to external sources of capital, showing how many assets are financed by shareholders equity against how many are from liabilities [15]. They can be seen as a measure of the risk, since high levels of debt can lead to distress and low credit scoring, but having low levels of leverage is also a symptom of short flexibility for borrowing;

• **Activity (Turnover) Ratios**: measure how well companies use their resources to generate revenues and cash, with focus on speed of circulation of assets in and out the business [15];

• **Market Ratios**: metrics which use the stock price together with fundamentals in order to assess the value of a publicly traded company. They are typically used to find if a firm is undervalued or overvalued, reason by which they are commonly called valuation ratios.

### 2.4 Machine Learning

Machine Learning (ML) is one of the largest fields in modern AI, which focuses on developing systems that learn and improve without being explicitly programmed to do so. ML is a statistical data driven AI approach since it uses often complex datasets to identify complex patterns and dependencies which are then used to extrapolate to future observations.

ML is divided into three main groups, with respect to the type of feedback used in the learning process:

• **Unsupervised Learning**: algorithms which learn from unlabeled data. These algorithms identify patterns and clusters in the input data that were previously unidentified;

• **Reinforcement Learning**: algorithms which learn from a series of penalties and rewards;

• **Supervised Learning**: the learning process is based on labeled datasets, where each of the learning patterns is associated with a simple input-output pair.

#### 2.4.1 Supervised Learning Algorithms

Supervised learning algorithms are the single largest family of ML algorithms and a popular field of research. The applicability of the “learn by example” framework and the increase in access to big structured datasets explain the recent rise in popularity. In particular, supervised learning algorithms
have been applied with success to several problems in the field of finance, such as bankruptcy prediction or stock market time-series prediction [4, 16].

The general supervised learning task involves a dataset of \( n \) data patterns, described as

\[
(x_1, y_1), \ldots, (x_n, y_n)
\]  

(2.1)

where \( x_i \) is an array with dimension \( m \), the number of features or attributes of the data, and \( y_i \) is the label value of each observation. Typically, the data patterns are grouped in matrix form, in the design matrix, given by \( X \in \mathbb{R}^{n \times m} \). Each of the data patterns in the design matrix is associated with labels, which are the targets of supervised learning. If this target value is finite and discrete, the supervised task is labeled as classification. If otherwise the outputs are continuous values, it is labeled as regression.

The goal of any supervised learning algorithm is to estimate an hypothesis function to approximate \( F(x) \), such that

\[
\hat{F}(x) = Y
\]  

(2.2)

where \( Y \) is the array of predicted labels. The learning process strikes to find a model \( \hat{F} \), in the search space of possible models, which performs well according to a given metric beyond just the training data [17]. This process is iterative and done through the evaluation of a loss function, which produces a score to be minimized or maximized. A typical loss function is the Mean Squared Error (MSE) between the predictions and true labels.

To assess performance, some data that is not used for training is kept on a separate test set, to be evaluated by the learned model. The quality of the model strongly depends on the observed extrapolation capabilities to this out-of-sample test data. This challenge is translated by the bias-variance tradeoff problem. Bias is known as the difference between the average prediction of a model, for different models obtained from the same algorithm on different training sets, and the actual value for the prediction. Models with high bias oversimplify the model and miss some important dependencies, which cause underfitting to the training data. On the other hand, variance is the variability of the prediction of a single data point for models obtained by the same algorithm on different training sets. High variance models are typically overly complex, fitting too well to the training data and corresponding noise, which causes a high generalization error. In general, there is trade-off between highly complex models with high variance and simple models with high bias [17]. This problem must be approached by changing the inner parameters of the algorithm, commonly designated as hyperparameters.

Supervised learning algorithms exist in high number and take different approaches to perform the tasks described above. One of the most common ways of grouping them is by their functionality and similarity in the learning and decision process:
• **Regression algorithms** relate a group of independent variables to a dependent target variable, the prediction label. Regression algorithms include linear and logistic regression;

• **Decision Tree algorithms** are classification and regression algorithms which partition the data based on consecutive conditions applied to the features. These conditions start from a root node, after which a sequence of tests is applied to the data, which decide the best splitting points using measures of node purity. The leaves represent the output given to a particular sample of data. Decision tree algorithms include Classification and Regression Trees (CART) and Chi-squared Automatic Interaction Detection (CHAID);

• **Artificial Neural Networks (ANN)** are inspired by the biologic connections of neurons in human brains. They are composed of units (perceptrons) connected by directed weighted links, used to recognize and predict patterns;

• **Bayesian algorithms** are a family of algorithms inspired by the Bayes rule. Notable Bayesian algorithms include Naive Bayes and Multinomial Naive Bayes;

• **Ensemble algorithms** rely on the use of several base learning algorithms, described in this context as “weak learners”, to build stronger predictive models. The outputs of the weak learners are combined in different ways, according to the ensemble technique used. Popular ensemble techniques include averaging, voting, bagging or boosting. Algorithms in this family include the Random Forest, a bagging technique, and XGBoost, which uses boosting.

### 2.4.2 Extreme Gradient Boosting

Extreme Gradient Boosting, or XGBoost [18], is an ensemble based supervised learning algorithm derived from the original Gradient Boosting Machine (GBM) algorithm [19].

Boosting is a technique of ensemble learning where each of the weak learners are trained sequentially. During this process, a first model is trained on the training data after which a second model will be retrained with the added objective of correcting the errors made by the previous model, and so on until the stopping criteria is met. In boosting ensembles, each of the weak learners are fit to weighted versions of training data. These weights are added to data samples which were misclassified by the previous model. When all of the boosting rounds are done, the weaker models are also weighted according to their performance and their outputs are combined to produce the final predictions. This sequential framework, described in Fig. 2.6, creates stronger models with lower bias and variance with increased the predictive performance when compared to a single weak learner.

The GBM, introduced by Friedman [19], combines traditional boosting with the Gradient Descent algorithm, by connecting stepwise additive expansions to the steepest descent minimization [19].
Figure 2.6: Description of the general boosting training process, where base learning models are sequentially fit to training data and errors are used as input for the following base learners.

The algorithm starts with the definition of a loss function and a base learner. Then, a weak learner is fit to the training data and the difference between predictions and actual outputs is calculated – the residuals. High values of the residual for a single data pattern mean that the model did not adjust to this particular point. The next model is fit to the residuals of the last, in an attempt to correct the errors made by the model, following the same strategy as the classical boosting approach. With GBM however, base models are fit to the pseudo-residual, the negative gradient value of the loss, since it pays less attention to outliers and is more general. At each boosting step, these new model fits are added to the previous models in a greedy stage-wise fashion in order to reduce the loss function. The optimization can be performed through Gradient Descent, an optimization algorithm which involves minimizing a loss function moving contrarily to the gradient, such that

$$F_{m+1}(x_i) = F_m(x_i) - \eta_m \frac{\partial J}{\partial F_m(x_i)}$$

(2.3)

where $F_m$ are the models obtained at boosting stage $m$ and $\eta_m$ is the step size of the optimization. This expression can be mathematically solved for any loss function $J$.

XGBoost uses Gradient Tree Boosting, based on the implementation by Friedman [19], setting the CART as base learners. CARTs differ from standard decision trees by associating a score to each of the leaf nodes, which give them extra flexibility to perform tasks beyond classification. The tree ensemble method is built based on $K$ additive functions, used to predict an output $\hat{y}_i$, described by Chen and Guestrin [18] as

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F}$$

(2.4)

where $\mathcal{F} = \{ f(x) = w_q(x) \}$ is the set of CARTs, given by $f_k$, which map a data pattern to a leaf node and corresponding score. Each of the trees is defined by their structure $q$ and has $T$ leaves weighted by $w$, such that $q : \mathbb{R}^m \rightarrow T, w \in \mathbb{R}^T$. 22
As with all the other supervised methods of learning described, obtaining a model is done through the optimization of a loss function. In this scenario, the objective function to be optimized is given by

\[ L(\phi) = \sum_i L(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \]  

(2.5)

where \( L \) is a differentiable and convex loss function. The second term represents a regularization term, given by

\[ \Omega(f) = \gamma T + \frac{1}{2} \lambda \| w \|^2. \]  

(2.6)

This term prevents overfitting by penalizing overly complex models, i.e trees with a high number of leaf nodes, giving priority to simpler base trees. The extent of this penalty can be controlled by increasing or decreasing \( \gamma \) and \( \lambda \).

Optimizing (2.5) is not possible in the Euclidean space, since the problem involves functions as parameters. This problem is solved using the GBM greedy additive training framework such that, at iteration \( t \), we optimize

\[ L^{(t)} = \sum_i L(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t). \]  

(2.7)

As described in the original derivation [18], this expression can be further simplified by taking the second-order approximation, removing constants and developing the regularization term 2.6, resulting in

\[ L^{(t)} = \sum_j \left[ \left( \sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left( \sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T. \]  

(2.8)

where \( g_i \) and \( h_i \) represent respectively the first and second order gradient statistics of the loss function and \( I_j = \{ i | q(x_i) = j \} \) is the set of training indices associated to leaf \( j \) by tree \( q \). The summation is now done over the leaf nodes instead of training patterns, since all training patterns associated with the same leaf get the same score. By equalling the derivative of (2.8) to zero we can compute the ideal weights for the leaf nodes and the best loss reduction,

\[ w_j^* = \frac{\left( \sum_{i \in I_j} g_i \right)^2}{\sum_{i \in I_j} h_i} \]  

(2.9)

\[ L^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^T \left( \sum_{i \in I_j} g_i \right)^2 + \gamma T. \]  

(2.10)

Equation (2.10) represents the quality of a fixed tree structure \( q \) with \( T \) lead nodes, and is obtained by
replacing Equation (2.9) in (2.8). These results provide a feasible way to compute the loss and optimal weights of the leaf nodes.

The next and final issue is the exploration of possible trees. This process is not done exhaustively over all possible tree structures, instead the tree is optimized level by level, making left and right splits and greedily picking those that minimize the loss function. This splits are done according to several criteria, as described by [18].

XGBoost presents other novelty features that make it stand out from traditional Gradient Boosting algorithms:

- Support for both Lasso and Ridge regularization, which helps reducing noise in the features and prevent overfitting. Lasso regularization can be additionally used for feature selection;

- Parallel split finding allows the algorithm to use multiple processors to speed up execution. While not building trees in a purely parallel way, XGBoost pre-sorts data entries before finding the ideal split, which makes it outperform traditional Gradient Boosting algorithms;

- Handling of missing feature values from sparsity-aware split finding, by learning the split direction that minimizes loss in such cases [18].

These factors, among others further described by Chen and Guestrin [18], made XGBoost the state-of-the-art GBM implementation both in terms of predictive and execution performance.

### 2.5 Genetic Algorithm

The Genetic Algorithm (GA) is one of the most used of the evolutionary computation algorithms, which draws inspiration from the biological evolution of species - as described by Charles Darwin “survival of the fittest” principle. It is a meta-heuristic that uses evolutionary operators such as selection, crossover, mutation and fitness to produce the solution to optimization and search problems [20]. These solutions typically consist on minimizing or maximizing one or multiple performance measures.

In the GA, each individual is encoded as a chromosome, represented by a fixed-length string, where each position represents a feature of the individual, the same way as genes do. Although binary values are typical for gene encoding, integers or floats can also be used. Encoding the potential solutions as chromosomes can be challenging, since this representation is merely abstract [20]. An example of chromosome representation is illustrated in Fig. 2.7.

The first step of the GA is the initialization, by generating \( n \) random individuals, which constitute the initial population. Each individual in the population has a fitness value, obtained from the GA objective function, also called the fitness function [17, 20]. A fitness function takes in an individual and returns
a numeric value, proportional to the quality of the solution, which should be minimized or maximized by the GA. This function should be defined by the programmer, according to the problem at hand.

The next step of the process is the selection of individuals to be used as parents for the next generation. Although, by traditional natural selection, the most fit individuals are to be chosen, this is not always ideal. Highly favoring the most fit individuals tends to diminish population diversity, which leads to premature convergence to sub-optimal solutions. Conversely, lowering the favourability to pick high fitness individuals leads to slow convergence. The most used selection processes for the GA are described below.

- **Random Selection**: Selection of a random individuals from the population;
- **Roulette Wheel Selection**: An individual is chosen with a probability proportional to their fitness value;
- **Tournament Selection**: Each time a parent is selected, a tournament of \( N \) random individuals from the population is performed and the most fit individual is picked for reproduction;
- **Stochastic Universal Sampling**: Improvement over traditional Roulette selection, providing minimum spread and bias.

The next step is the crossover of the selected parents. Crossover is the process by which two individuals combine their genes, creating offspring chromosomes. From the mating pool, two parents are picked at random. Then, crossover positions are defined along each of the parent’s chromosome strings, according to a crossover method. Finally, the child is generated by swapping each portion of the chromosomes along those positions. The most popular crossover strategies are represented in Fig. 2.8.

After the crossover, the resultant children are exposed to the mutation process. This step prevents the solutions to converge in local minima, reintroducing diversity in the population. Mutation is implemented by randomly choosing a gene from the chromosome string and changing its value. In binary
encoding, this change is typically flipping the bit, but in other implementations include switching gene’s positions or replacing a gene by drawing a random value from a probability distribution.

Both crossover and mutation occur at a given rate. Crossover rate is given by $P_c$, representing the probability that crossover will be performed by a pair. Setting a crossover rate below 1 ensures that some parents in the mating pool can be copied to the next generation of chromosomes. On the other hand, $P_m$ represents the probability that a chromosome will suffer a mutation on one of its genes. The values of $P_m$ should be kept low, since the contrary can lead to simple random search [20].

After the previous steps are concluded, the generated individuals replace the previous population. This process is repeated until a stopping criteria is met. Typical stopping criteria include timing limits, iteration limits or no improvements in the fitness value for a set number of iterations.

### 2.6 Related Works

This section reviews relevant studies on dividend policy and applications of machine learning on dividend policy predictability and other financial fields. Section 2.6.1 presents relevant works on dividend policy, including descriptive modelling studies on the financial determinants that are associated with the increase or decrease of dividend payouts, as well as with their sustainability. Section 2.6.2 reviews AI and ML based works in the fields of finance, with focus on dividend policy and financial distress pre-
dictability. Lastly, Tables 2.3 and 2.4 outline some of the most relevant studies described throughout this section.

2.6.1 Works on Dividend Policy

In order to find what drives managers to change their dividend policies, early studies made by Lintner, John [21] conducted interviews, finding that they mostly look at current earnings and to the target level of dividend payout when making the decisions, which were made conservatively. Brav et al. [22] furthered this study in a more modern setup, conducting interviews and surveys on public firms from several sectors, in order to find the most relevant factors that managers use to decide to raise or cut their dividend. They find that maintaining the historic track record, stability of future earnings projections and sustainable changes in earnings are among the key factors in the decision process. Brav et al. [22] also asserts that managers tend to avoid reducing the dividend yield, and for that they look at recent values from previous quarters, while also trying to maintain a smooth stream from year to year. Lintner, John [21] argued in favor of this smoothing in dividends, stating that it was an attempt to separate earnings volatility from the payout.

Miller [23] analyzed the “stickiness” of dividends under poor financial conditions, finding financial performance trends that drive dividend cuts. They find that firms which maintain or increase their dividend in a given year tend to have higher assets, sales and cash-flow growth. For distressed firms, their study points that approximately half of the group experiencing both negative profits and cash flows maintain or increase their payouts, mostly by drawing down on cash reserves and raising external capital. Although most of the performance measures are no longer statistically different, leverage and asset growth show significant differences between both groups, respectively higher and lower in cutters. This is consistent with previous studies by Grullon et al. [6], which also find statistically significant declines in leverage for dividend raising firms. DeAngelo and DeAngelo [9] also conducted a study on dividend payers which cut or omitted their dividend, finding that more than a half of the sample firms had eventually binding debt covenants in the year of the first dividend reduction, whereas others in the years of later reductions. These previous studies highlight the importance of leverage on dividend sustainability.

Furthermore, Gill et al. [24] analyzed a sample of financial reports in order to find which factors impacted the most in the magnitude of the dividend payout ratio on American firms. This study introduced the adjusted DPR as an indicator, by adding back depreciation value to net income, and divided sample data by sectors, mainly the service and manufacturing sectors. The results showed a positive correlation between adjusted payouts and profitability and negative correlation with the MBR across all sample. However, most metrics were strongly sector dependent, which shows the importance of contextualizing the performance according to this dependency.

Due to the importance of earnings as a determinant for dividend policy, other studies focused on
assessing earnings quality in regular dividend payers. Tong and Miao [25] measured quality in earnings with regard to discretionary accruals and in the mapping of accruals into cash flows, which are argued to reflect the current operating performance accurately. These enforce the previous studies, meaning that earnings and their quality are relevant determinants of dividend sustainability, since it is costly to support cash dividends that do not reflect underlying performance and cash flows [25].

Table 2.3 recaps relevant studies on drivers of dividend policy, along with the financial variables that most contribute to the conclusions of each. This is done in order to compile a set of fundamental drivers associated with dividend policy decisions and can be further used in this study as data features.

2.6.2 Works Dividend Policy Prediction and Machine Learning

ML predictive algorithms have been increasingly used in finance for the last decades, particularly in fields such as dividend policy and financial distress prediction - which are at the core of this work. This section goes over and recaps some of the most relevant studies in this context. Lastly, Table 2.4 outlines the studies discussed throughout the section, in terms of data sources, features and prediction targets.

One of the first models developed for dividend prediction was introduced by Marsh and Merton [26]. This was a simple yet effective econometric regressive model, which used past and present stock price data \( (P_t, P_{t-1}) \) and dividend data \( (D_t) \) to predict future dividends [26, 27]. Using historical price and dividend data, Kim et al. [28] applied ML techniques to achieve a similar goal, using CART with Knowledge Integration (KI) for predicting future dividends. This KI approach had the upside of deriving decision rules which could be interpreted, contrasting with other algorithms where knowledge is buried in weights and parameters. The KI algorithm extracted a set of rules from 4 types of datasets using the pruned trees, which were then integrated into 39 rules based on past and current dividend and price data from over 100 sample companies. This work showed successful results for different tolerance levels of accuracy, outperforming the regressive models developed by Marsh and Merton [26] and showing that using ML approaches can yield better results than traditional econometric approaches. Subsequent studies by Won et al. [29] used Genetic Algorithm Knowledge Refinement (GAKR), which consisted on running multiple rule based algorithms (CART, CHAID, C5.0 and QUEST) and using the Genetic Algorithm (GA) to generate the optimal subset of rules for dividend policy prediction. The prediction targets were defined as binary labels corresponding to dividend maintenance, if \( D_t \leq D_{t+1} \), or reduction, if \( D_{t+1} < D_t \), essentially designed to predict the sustainability of the current level of dividends. The results showed that the accuracy of the GAKR was on average higher than the underlying rule generating algorithms, with less generalization errors – showcasing the advantage of classifier ensembling.

A large percentage of the studies in the field dividend policy prediction are based on fundamental data instead of sole price and dividend historic data. These include Laoh [30], in which ANN and GA were applied to dividend prediction, using financial ratios of the payout year, described in table 2.4.
While this study showed that the GA can be effectively used to find the best model weights, the results obtained with the ANN did not outperform traditional linear regression in dividend prediction. Luebke and Rojahn [31] tested a set of ML algorithms to predict dividend change patterns of maintenance, increase and decrease. In particular, they compare the accuracy of the Support Vector Machine (SVM), CART and Random Forests with traditional methods such as Linear Discriminant Analysis (LDA) or Multinomial Logit, with the features described in Table 2.4. The best predictive results obtained with the ML algorithms achieved similar misclassification rates as the traditional methods. The CART models allowed to evaluate feature importance, which showed that net income, MBR and turnover growth rate had the most predictive power. Hobbs and Schneller [8], on the other hand, studied predictability of long term dividend streak maintenance. The study aimed to group dividend initiators in short term and long term payers, respectively those that are able to sustain their dividend for less than 3 years or more than 7 years. Hobbs and Schneller [8] were able to correctly label 150 out of 243 initiators, based on comparison between pre-initiation performance, using the volatility of profitability (risk adjusted Return on Assets (ROA) [6]) and logistic regressions using the features outlined in Table 2.4.

Dividend cuts and omissions can also be seen as one of the first measures in a sequence of events leading to bankruptcy, as described in Section 2.2.3. In this context, many state-of-the-art algorithms have been applied to fundamental financial datasets, as bankruptcy prediction became one of the most researched topics in the last years.

Mai et al. [32] employed ANN deep learning techniques using structured financial data, complemented by unstructured textual accounting fillings data, to predict bankruptcy of publicly listed American firms. Furthermore, Le et al. [4] applied a similar study to 3 balanced and imbalanced financial ratio based datasets, using XGBoost with GPU integration, comparing the proposed solution with other state-of-the-art gradient boosting algorithms. The results showed that XGBoost both consistently outperformed the other approaches in terms of ROC-AUC score and in terms of computation time. Huang and Yen [33] tested 6 state-of-the-art supervised and unsupervised algorithms for overall financial distress prediction based on financial ratio data and found XGBoost to be the overall better performing classifier in terms of accuracy, type I and II errors. These results suggest that, even though there is no record of XGBoost application to dividend prediction, it has achieved outstanding results with fundamental structured numerical data, namely in financial distress prediction problems.
Table 2.3: Summary of selected studies on dividend policy and driving financial factors behind dividend policy changes. The objectives and goals of each work are outlined, together with the financial features which showed more impact on the study.

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Sample data</th>
<th>Study outline</th>
<th>Financial variables associated with dividend policy</th>
</tr>
</thead>
<tbody>
<tr>
<td>[34]</td>
<td>NYSE &amp; AMEX quarterly dividend payers (1979-1991)</td>
<td>Analysis on financial information content of earnings and dividend changes</td>
<td>Short-term earnings and earnings growth</td>
</tr>
<tr>
<td>[22]</td>
<td>256 firms listed on NYSE, AMEX &amp; Nasdaq</td>
<td>Surveys of firm managers to find key drivers of dividend policy decisions</td>
<td>Length of dividend streak, earnings growth, dividend yield and dividend growth</td>
</tr>
<tr>
<td>[23]</td>
<td>Compustat firms, except financial and utilities sectors (1977-2007)</td>
<td>Study companies that cut or maintain dividends under cash shortage</td>
<td>Firm age, size, leverage, sales, operating cash-flows, dividend-to-assets, dividend streak length and short term growth rates of key financial measures (leverage, assets and sales)</td>
</tr>
<tr>
<td>Ref.</td>
<td>Data</td>
<td>Algorithm</td>
<td>Prediction targets</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>-----------</td>
<td>-------------------</td>
</tr>
<tr>
<td>[26]</td>
<td>CRSP data from NYSE (1926-1981)</td>
<td>Linear Regression</td>
<td>Dividend Payout in subsequent year</td>
</tr>
<tr>
<td>[28]</td>
<td>Firms listed on KRX (1980-2000)</td>
<td>CART with KI</td>
<td>1-year ahead dividend payout</td>
</tr>
<tr>
<td>[29]</td>
<td>Firms listed on KRX (1980-2000)</td>
<td>GAKR</td>
<td>1-year ahead dividend payout change</td>
</tr>
<tr>
<td>[31]</td>
<td>Firms listed on Prime Standard (2007-2010)</td>
<td>SVM, Decision Trees, Random Forests and LDA</td>
<td>Dividend Payout Change (Maintenance, Increase or Decrease)</td>
</tr>
<tr>
<td>[8]</td>
<td>CRSP monthly financial data (1962-2000)</td>
<td>Logistic Regression</td>
<td>Classification of firm into temporary (unsustainable) or permanent dividend payers</td>
</tr>
<tr>
<td>[32]</td>
<td>CRSP/Compustat USA firms (1994-2014)</td>
<td>Deep learning (ANNs)</td>
<td>1 to 3 year ahead bankruptcy status</td>
</tr>
<tr>
<td>[4]</td>
<td>3 datasets, including the USABDS of USA firms (1981-2009)</td>
<td>XGBoost</td>
<td>1-year ahead bankruptcy status</td>
</tr>
<tr>
<td>[33]</td>
<td>TEJ firms (2010-2016)</td>
<td>6 algorithms, including SVM and XGBoost</td>
<td>Distress state (bankrupt/recovery) in the sample period</td>
</tr>
</tbody>
</table>
Implementation

Contents

3.1 Architecture ................................................. 33
3.2 Data Loading Module ....................................... 35
3.3 Market Scraper Module ..................................... 37
3.4 Ratio Generation Module ................................. 38
3.5 Data Update Module ....................................... 41
3.6 Preprocess Module .......................................... 41
3.7 XGBoost Module ............................................ 44
3.8 Genetic Algorithm Module ............................... 46
3.9 Stock Ranking Module ..................................... 49
3.10 Sliding Window Module .................................. 49
3.11 User Interface Module .................................... 52
This chapter describes the proposed implementation of the dividend stock selection system, from the data extraction process up until the final stock ranking. Section 3.1 starts by describing the overall system architecture, while Sections 3.2 through 3.11 describe the building blocks of the system.

### 3.1 Architecture

The implemented system has the goal of selecting S&P500 dividend stocks by scoring and ranking them in their likelihood to increase their payout in the following periods. The generated rankings are intended to be used for stock selection, by picking stocks among the $n$ best ranked. This allows to filter and remove the dividend stocks with the most unsustainable dividends – which is a core concern in dividend investing.

Figure 3.1 describes the high level architecture of the system, including all of the layers and modules, as well as the circulation flow of data between them. The system is divided into three main layers: the data layer, the simulation layer and the user layer.

![Figure 3.1: Summarized high level description of the implemented system, including the modules and the flow of data communication between them.](image)

The data layer is responsible for the generation of the main dataset – this includes data collection, transformation and error-checking. The execution flow of the data layer is summarized by the following steps:

1. The data loading module retrieves quarterly financial statement records for S&P500 constituents, from 2005 up to 2018, saved locally. This will be the base of the initial dataset;

2. The split-corrected dividend history data is retrieved from the market scraper module, for all sample firms. The dividend payouts are then annualized and appended to the main dataset, according to
the observation year. The dividends are used to filter the sample, keeping only companies which have paid dividends in the past;

3. The closing stock prices are retrieved from the market scraper module and merged firm by firm, filling in the closing prices at the release date of each quarterly observation;

4. The industry sector codes are retrieved and appended to each observation, according to the sector of the firm that originated them;

5. The ratio generator module takes in the dataset and generates a set of financial ratios, to be used as the predictors of dividend sustainability. This module also attributes a set of flags to the observations, signalling the existence of dividend streaks;

6. The data is then further completed with recently released records, via the data update module. This module scrapes financial statement data from multiple available sources and repeats the steps 2-5, formatting the new observations in a similar fashion;

7. The data update module then merges the scraped observations with the previous data, which completes the definition of the raw dataset, to be used by the predictive layer.

The predictive layer has the goal of providing rankings of dividend stocks for different evaluation periods, using the XGBoost + GA approach. A step-by-step overview of the predictive layer is given below:

1. The preprocess module takes in the dataset and applies preprocessing transformations required to build the model, such as data filtering, missing value handling or standardization. The data is filtered in order to keep only observations belonging to firms with an ongoing dividend streak of at least 3 years. The data prediction label is also set in this step, to the 1-year ahead dividend change flag, defined in step 5 of the data layer. Finally, the features are also filtered such that only the relevant set of financial ratios (Table 3.1) and sectors are input to the algorithm;

2. In order to train and validate the model, the preprocess module splits the data into the training and validation periods – according to the evaluation year selected;

3. The GA module evolves a GA in order to find a set of XGBoost parameters that best suit the specific data window. The fitness is evaluated by fitting a model on the training data with the chromosome encoded parameters and evaluating performance on the out-of-sample validation set. This evaluation is done through two metrics: the Receiver Operating Characteristic (ROC) and Precision-Recall Area Under Curve (AUC);

4. The best parameters are used to retrain XGBoost, which is then used to score the evaluation period observations;
5. The stock ranking module groups and averages the scores given to a firm, for the evaluation period. The firms are ranked in ascending order, with higher scores representing the firms with riskier dividends, which are most likely to break their dividend streak in the year following evaluation;

6. According to a list of evaluation years, the sliding window sequentially shifts the training-validation-evaluation window and generates stock rankings for the next year, reiterating steps 1-5. The generated scores can be combined with the previous in a moving average fashion – such that previous models contribute to the final rankings. The sliding window can be seen as the top level module of the predictive layer.

Finally, the user layer defines an interface for the user to interact with the system. It is tightly coupled with the data and predictive layers, supporting the exchange of information throughout the steps described above. The user layer makes it possible to receive logs, statistics and results, as well as set up several parameters, such as the percentage of top ranking firms companies to retrieve or the evaluation periods to use.

The remainder of the chapter details the implementation and design choices of the modules described in this section.

### 3.2 Data Loading Module

This module of the system is responsible for reading the financial data from raw database records, gathering sample data for dividend paying firms, merging the data records with other relevant data and performing basic error correction.

One of the main sources of data for this work is the Center for Research in Security Prices (CRSP) and Compustat database, containing Income Statement, Balance Sheet and Cash Flow Statement data from the S&P500 index, from 2005 to 2018. The data was collected in the .csv format, represented by two files: annual financial statement data and quarterly financial statement data. In the context of this work, only quarterly data was used to build the data sample, comprised of raw data records with 39335 rows and 644 columns - each of the columns representing a financial statement item of a given firm quarter, identified by the respective Compustat mnemonic. Each row is uniquely identified by the ticker of the firm, by the date of the ending period of the financial quarter and release data of the financial report. This initial data structure is the baseline from which the data layer will build the final dataset, to be used by the algorithm.

The module starts by loading this dataset and filtering it, keeping only the financial items to be used to compute financial ratios, corresponding to those analyzed in Section 2.3.2. A slice of the output Pandas dataframe is presented in Fig. 3.2.
Next, the dividends are merged to each of the rows. The dividend data is retrieved from the market scraper module, further described in Section 3.3. The module returns the dividends firm by firm as a data series containing the ex-dividend date and dividend payout amount. The ex-dividend date is used to group the annual dividend payouts of each firm, which are then merged with the financial observations of the corresponding year, on the initial financial statement dataset. One issue regarding this implementation is the fact that companies with regular dividend policies can issue a Special Dividend Distribution (SDD), which cannot be distinguished in the dividend data records. In the context of this work, the focus is on regular dividend payments, so these extraordinary dividends are not to be considered. The following measures were taken to solve this issue:

- The dividend policy of each sample firm is identified, by computing the mode of the number of dividend payouts per year on the historical data, $n_{div}$. This allows to find if the firm follows an annual, semi-annual or quarterly dividend policy. If a firm pays a larger number of dividends than their regular policy, such that $n > n_{div}$, only the $n_{div}$ smallest payouts are kept and annualized - the rest are assumed to be SDDs.

- Any abnormally high dividend payout is dropped, with respect to the firm’s standard dividend payouts. For example, a single quarterly dividend 5 times the mean of the previous dividends paid is dropped, and is not factored into the annual dividend of the firm.

In the following step, the industry sectors of the firms are added to the data, according to the Global Industry Classification Standard (GICS) terminology – a widely used taxonomy system created by S&P, which classifies publicly listed companies into 11 sectors, which are also subdivided into different industries. The data is then merged with the stock price data, which is again retrieved from the market scraper module. The stock price of each firm is merged to the data by adding the closing price value on the release date of each financial statement to the corresponding observation.

The resulting dataframe is then passed to the ratio generation module, which returns the initial dataset plus the financial ratios, as described in Section 3.4. Finally, the output dataset is saved locally.
3.3 Market Scraper Module

The market scraper module retrieves historical price and dividend data, by web scraping data records from *Yahoo Finance*.

The price data is retrieved according to the selected historic period and firm ticker, specified as parameters. The scraping process itself is performed with *YahooQuery* Python library, which provides fast and simple access to *Yahoo Finance* data. Price data is available in several columns, which include:

- **Open price**: Initial price of the stock in a trading day;
- **Close price**: Final price of the stock in a trading day;
- **High**: Highest price of the stock during a trading day;
- **Low**: Lowest price of the stock during a trading day;
- **Adjusted Close**: Final price of the stock in a trading; day, adjusted for stock splits, dividends and other corporate actions.

Even though adjusted close prices depict a better historical view of the price value, being corrected for dividend payouts and splits, they may add look-ahead bias to the model – since the dividends are discounted to the present date. For this reason only the closing price was retrieved.

For dividends, the web scraping process was implemented manually, due to the need to perform data corrections and unavailability of libraries. The process begins by setting the number of historical days to retrieve, defining a start and end date accordingly, as well as a ticker – which are taken as parameters. Then, the URL subdomain is formatted according to the selected values and a *GET* request is issued to the web server, using the Python *Requests* module. The dividend *HTML* tables are parsed and converted into a *Pandas*, a data structure library for Python, dataframe type. If the table returns an error or has no contents, an empty dataframe is returned by the module instead.

Afterwards, the process described above is repeated for the stock split data. Stock splits affect the real value of dividends since *Yahoo Finance* keeps the original payout values - for example, in a 2:1 stock split\(^1\) the dividend is likely to be roughly halved past the split. To mitigate this problem, which is bound to affect the process of evaluating dividend streaks, the dividends are adjusted prior to the split by dividing them by the split fraction, as described in Fig. 3.3.

An example of the final error corrected dividend data returned by the module is represented in Fig. 3.4.

---

\(^{1}\) Stock splits divide existing shares by a given ratio, improving liquidity. In a 2:1 split, the shareholder receives two shares for each share he owns.
Figure 3.3: Adjustment applied to dividend payouts, according to the date of the stock split.

Figure 3.4: Sample of dividend series returned by the market scraper module. Each ex-dividend date is paired with the per-share dollar amount of the payout.

3.4 Ratio Generation Module

The ratio generation module takes the quarterly dataset and produces financial ratios and other derived data items from the financial statements. This module is responsible for building the features to be used by the algorithm in later steps of the data flow.

The choice of features was inspired on the studies described in Section 2.6, specifically the ratios highlighted in Tables 2.3 and 2.4, conditioned by the availability of the items necessary to compute them. The features are presented in Table 3.1, grouped by the performance measure they evaluate.

After the base features are generated, the module creates binary items to flag past streaks of dividend increase or maintenance. To identify past dividend streaks on each quarter, the threshold defined was 3 years of consecutive increasing annualized dividends. This includes both companies with long and well-established dividend growth programs and those with shorter dividend streaks, but which may be as likely to consistently grow their dividend in the future. The process used to identify the streaks on the data is described as follows:

1. The observations are grouped by the release date of the financial statements that originated them, e.g., 2018 observations include all observations generated from financial statements released during 2018;

2. The annualized dividend of each firm is computed for all the observation years and is joined to each observation, according to the corresponding year. For the sake of example, observations generated from financial statements released during 2018 are associated with the annualized cash...
dividend paid by the firm during 2018. This step is performed by the data loading module, when appending dividend data;

3. Dividend increase streaks are identified by comparing the annualized dividend associated to an observation with the one from one year before, and so on. Therefore, for a 3-year streak of dividend increase, the following must hold:

\[ D_t > D_{t-1} > D_{t-2} > 0. \]  

(3.1)

Where \( D_t \) is the total per-share dividend paid during year \( t \). Observations belonging to firms which fulfill eq. (3.1) are flagged in distinct column as having an ongoing streak, in the ratio generation module.

In addition, it is possible to generate similar variables to identify streaks of different size or using different criteria other than increase – such as the maintenance of the current level of dividends.

Since the goal of the system relies on predicting dividend increases on the sample firms, each observation in the dataset is also associated to a similar column, respective to future dividend increase. Positive labels indicate that the firm increases the dividend from their current observation year into the next one, such that

\[ D_t < D_{t+1} \]  

(3.2)

is true. Negative labels are therefore associated with the unsustainability of current dividends. Following the addition of all the data items the data is cleaned, removing unnecessary columns.
Table 3.1: Financial features created by the ratio generation module, as well as the acronyms used to encode them as features and the formulas used to compute them. The ratios are grouped by the financial performance measure evaluated, together with some of the studies described on section 2.6 which employ similar metrics. The resolution of data points \( t \) is the financial quarter.

<table>
<thead>
<tr>
<th>Profitability Ratios [4, 6, 8, 32, 36]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Gross Profit Margin ( gpmq )</td>
<td>((\text{Revenue} - \text{Cost of Revenue}) / \text{Revenue})</td>
</tr>
<tr>
<td>Risk adjusted ROA ( adj_roaq )</td>
<td>(\text{ROA} / \sqrt{[\text{ROA}]})</td>
</tr>
<tr>
<td>Return on Equity (ROE) ( roeq )</td>
<td>(\text{Net Income} / \text{Shareholders’ Equity})</td>
</tr>
<tr>
<td>Return on Invested Capital (ROIC) ( roiq )</td>
<td>(\text{Net Income} / \text{Invested Capital})</td>
</tr>
<tr>
<td>EBITDA Margin ( ebitdaq )</td>
<td>(\text{EBITDA} / \text{Revenue})</td>
</tr>
<tr>
<td>Cash Flow Margin ( cfmq )</td>
<td>(\text{Operating Cash Flow} / \text{Revenue})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Liquidity Ratios [4, 22, 32, 36]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Ratio ( crq )</td>
<td>(\text{Current Assets} / \text{Current Liabilities})</td>
</tr>
<tr>
<td>Quick Ratio ( qrq )</td>
<td>((\text{Receivables} + \text{Cash}) / \text{Current Liabilities})</td>
</tr>
<tr>
<td>Cash Ratio ( csrq )</td>
<td>(\text{Cash and Equivalents} / \text{Current Liabilities})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Leverage Ratios [4, 9, 23, 32, 36]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Debt Ratio ( drq )</td>
<td>(\text{Liabilities} / \text{Assets})</td>
</tr>
<tr>
<td>Debt-to-Equity Ratio (DER) ( derq )</td>
<td>(\text{Liabilities} / \text{Shareholders’ Equity})</td>
</tr>
<tr>
<td>LTD-to-EBITDA Ratio ( ltdaq )</td>
<td>(\text{Liabilities} / \text{EBITDA})</td>
</tr>
<tr>
<td>Long Term Debt (LTD) ( / \text{Assets})</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Leverage Growth Ratios [9, 23]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Debt Ratio Growth ( dr_g4q )</td>
<td>([\text{Debt Ratio}(t) - \text{Debt Ratio}(t-4)] / \text{Debt Ratio}(t-4))</td>
</tr>
<tr>
<td>LTD Growth ( ltd_g4q )</td>
<td>([\text{LTD}(t) - \text{LTD}(t-4)] / \text{LTD}(t-4))</td>
</tr>
<tr>
<td>DER Growth ( der_g4q )</td>
<td>([\text{DER}(t) - \text{DER}(t-4)] / \text{DER}(t-4))</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Turnover Ratios [4, 31, 32, 36]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Inventory Turnover ( invxq )</td>
<td>(\text{Revenue} / \text{Mean}[\text{Inventory}(t), \text{Inventory}(t-1)])</td>
</tr>
<tr>
<td>Receivables Turnover ( rectrq )</td>
<td>(\text{Revenue} / \text{Mean}[\text{Receivables}(t), \text{Receivables}(t-1)])</td>
</tr>
<tr>
<td>Asset Turnover ( attq )</td>
<td>(\text{Revenue} / \text{Mean}[\text{Assets}(t), \text{Assets}(t-1)])</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Earnings Quality Ratios [25]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Accruals Ratio ( accrq )</td>
<td>([\text{NOA}(t) - \text{NOA}(t-1)] / \text{Mean}[\text{NOA}(t), \text{NOA}(t-1)])</td>
</tr>
<tr>
<td>Sloan Ratio ( sloanq )</td>
<td>((\text{Net Income} - \text{Operating Cash Flow} - \text{Investing Cash Flow}) / \text{Assets})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Growth Rates [22, 23, 34]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Revenue Growth ( rev_g4q )</td>
<td>(\text{Revenue}(t) / \text{Revenue}(t-4))</td>
</tr>
<tr>
<td>Net Income Growth ( ni_g4q )</td>
<td>(\text{Net Income}(t) / \text{Net Income}(t-4))</td>
</tr>
<tr>
<td>Dividend Growth ( div_g4q )</td>
<td>(\text{Dividend}(t) / \text{Dividend}(t-4))</td>
</tr>
<tr>
<td>Sustainable Growth ( sust_gr )</td>
<td>(\text{ROE} \times (1 - \text{DPR}))</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Market Valuation Ratios [8, 28, 29, 31]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Market to Book Ratio (MBR) ( mbr )</td>
<td>((\text{Shares Outstanding} - \text{Price}) / (\text{Assets} - \text{Liabilities}))</td>
</tr>
<tr>
<td>Dividend Yield ( dy )</td>
<td>(\text{Dividend-per-share} / \text{Price})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dividend Coverage Ratios [6, 23]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dividend Payout Ratio (DPR) ( dpr )</td>
<td>(\text{Dividend} / \text{Net Income})</td>
</tr>
<tr>
<td>Dividend-to-FCF ( divfcf )</td>
<td>(\text{Dividend} / \text{Free Cash Flow (FCF)})</td>
</tr>
<tr>
<td>Dividend-to-Assets ( diva )</td>
<td>(\text{Dividend} / \text{Assets})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Size Ratios [23, 35]</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Firm Size ( sizeq )</td>
<td>(\log(\text{Assets}))</td>
</tr>
</tbody>
</table>
3.5 Data Update Module

As described in 3.2, the initial data collection stored in disk includes only financial statements released up to 2018. In order to update the data used by the system, the data update module retrieves new financial statements from the web and appends them to the existing data.

The sources of web data used were both Macrotrends and Yahoo Finance. Macrotrends was selected since it publishes quarterly financial statements results upon release for free, for the large majority of sample firms. Yahoo is, as previously mentioned, the choice for price and dividend data, due to their availability and reliability. The execution of the module begins by iteratively going through all the firms in the original dataset and scraping financial statements for each of them. The module defines methods to scrap each of these statements, by issuing GET requests using the Requests Python library. After retrieving the web objects, the BeautifulSoup Python library is used to parse the contents of the page, which are converted into a dataframe. In order to reduce the size overhead, the module only pulls the financial items needed to compute the fields described in Table 3.1. During execution of the data scraping, it is common for the operations to block or be interrupted, due to the query limits imposed by the websites. In order to solve this, after the data for a single firm is gathered, the whole dataset is saved locally. The system resumes by checking the firms that have already been scraped and determines if the previous run was incomplete based on this list, picking up were it left off – avoiding restarting the process, in case of failure. The resulting dataframe, containing the financial quarterly data from 2018 onwards, is stored locally in disk memory.

Next, the data update module merges the previously obtained web scraped data with the processed dataset obtained from the data loading module. The processed dataframe is first read and, for all firms in the original sample, the dividend and price data is retrieved, using the methods described in sections 3.2 and 3.3. Next the web scraped fields are used to compute the financial ratios described in table 3.1 – which is again handled by the ratio generation module. The data is then merged, such that the final dataset includes the complete up to date records of dividend payers and the respective financial ratios under analysis, from 2005 to the present. This dataset is stored in disk as a Pickle file.

The simplified flowchart of the sequence of processing steps described in this section is represented by Fig. 3.5.

3.6 Preprocess Module

The preprocess module defines a set of procedures to process, filter and split the data, which are necessary to train and build the predictive model, taking in the fully assembled and validated dataset from the data layer. These transformations consist of:
1. Filtering the records by their dividend streak status e.g, the financial quarters which belong to firms with a streak of 3 consecutive years of dividend increases are kept, while the remainder are dropped. This is done using the flags defined in Section 3.4 (Eq. (3.1)). The filtering criteria of the backward streaks can be changed by the user, even though the standard baseline for the system is 3 years;

2. The prediction labels for classification are assigned, using the future increase flags defined in Section 3.4 (Eq. (3.2)), such that each observation in the dataset is paired to a binary value, signalling if the firm will or not increase their dividend in the following year;

3. The categorical features converted to the one-hot encoding scheme, due to incapability of XGBoost of handling purely categorical variables as is. In this context, the only categorical variable

---

**Figure 3.5:** Simplified flowchart of the data update module. The chart represents the execution thread used to update the initial dataset, with the data available in the present, posterior to the availability of CRSP/Compustat records.
which will be added as a feature is the industry sector GICS codes of each financial observation;

4. Rows containing more than 20% of NaN values are dropped from the dataset. The reason for not dropping all rows including missing values is to preserve data in the sample which may still include valuable information, while taking advantage of the missing value robustness provided by XGBoost [18].

The preprocess module also reports back to the user relevant statistics and data insights, such as the sector distributions, class balance and missing value percentage. Class balance is reported due to the strong imbalance of the data, since dividend cutters are in minority among dividend paying firms, as most firm observations of firms with a 3 year streak do maintain the tendency, as previously discussed.

The other key functionality of the preprocess module is splitting the data in different folds, a key step in the development of the predictive ML models. The splits are done accordingly to the release date of the financial report that originated the data observations, such that a 2018 financial report is considered as such if released during that year – albeit it could pertain to an earlier period. The data is first split into the following folds:

- **Training set**, used to fit an initial model on the financial data;

- **Validation set**, used to evaluate the performance of the model for different hyperparameter setups, assessing its extrapolation capabilities and guiding the process of parameter searching, which is essential to avoid overfitting;

- **Test set**, also described as evaluation set in the context of this work, comprises the data observations to be scored and evaluated by the final model.

Since the goal of the system is to use historic data of time-series nature to predict future dividend increases, the data splitting process is non-trivial, due to the data not being independent and identically distributed (i.i.d) – which causes correlation between data observations. This means that the traditional validation approaches, such as n-fold cross-validation, cannot be used. Instead, the approach chosen uses the “walk-forward” partitioning technique – such that observations up until \( t \) comprise the training data, from \( t \) to from \( t + 1 \) the validation data and \( t + 1 \) to \( t + 2 \) the test data. This essentially means that the observations from a window of time are used to train and score observations in the next time window. Specifically, the data splitting process is defined such that 6 years worth of observations are used in training, 2 years are used in validation and 1 year is used for scoring/evaluation. As adjacent observations are prone to increased inter-dependency, the system leaves a 1 year gap between the validation and testing sets in order to increase the independence between folds.

Fig. 3.6 represents the data window splitting process. The definition of the window is based on the testing period, given by \( t_{\text{evaluation}} \). Following the example, setting \( t_{\text{evaluation}} \) to 2018 uses data
Figure 3.6: Example of a single data window defined by the system, in which the model scores the financial observations from the test year 2018, using them to rank the sustainability of dividend stocks for 2019. Initial training data is used for parameter tuning, after which it is combined with validation set for retraining. Initial data split

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Training Data</th>
<th>Validation Data</th>
<th>Test Data</th>
</tr>
</thead>
</table>

Final split for model retraining

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Training Data</th>
<th>Test Data</th>
</tr>
</thead>
</table>

Observations from 2009 to 2014 for the initial training, from 2016 to 2017 for validation and scores observations from 2018. As also described by Fig. 3.6, an additional fold is generated, which aggregates the initial training and validation sets. This set is used to retrain the model after the model parameters are learned.

After the splitting step, the module also standardizes the data, removing the mean and scaling to unit variance. The preprocess module outputs the model ready subsets of data.

### 3.7 XGBoost Module

The creation, training and evaluation of the ML models is implemented by the XGBoost module. The module takes as input the training data, the evaluation data and the hyperparameters and builds the binary logistic XGBoost classifier, returning the list of scores given to each financial observation on the evaluation data.

XGBoost supports several hyperparameters which control the general behaviour of the model, the boosting process or the learning task. While the general and learning setup is usually predefined, tree booster parameters should be fine tuned in order to adapt the model to the particular problem and data. These parameters significantly change the behaviour of the algorithm, for example by controlling the sequential weights given by base learning trees during the boosting process. The set of parameters to be defined, as well as their respective role, are represented in Table 3.2.


<table>
<thead>
<tr>
<th>Parameter</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>n_estimators</td>
<td>Number of weak learners used</td>
</tr>
<tr>
<td>eta</td>
<td>Shrinkage or learning rate, used to shrink feature weights at each step, making the boosting process more conservative</td>
</tr>
<tr>
<td>min_child_weight</td>
<td>Minimum sum of instances weight in a child. Used to prevent overfitting</td>
</tr>
<tr>
<td>max_depth</td>
<td>Maximum depth of the tree. High values make the model more complex</td>
</tr>
<tr>
<td>gamma</td>
<td>Minimum loss function reduction required to make a node split</td>
</tr>
<tr>
<td>subsample</td>
<td>Subsample ratio of training observations used when growing a tree</td>
</tr>
<tr>
<td>subsample_bytree</td>
<td>Subsample ratio of training features used when growing a tree</td>
</tr>
<tr>
<td>alpha</td>
<td>L1 regularization term</td>
</tr>
<tr>
<td>lambda</td>
<td>L2 regularization weight</td>
</tr>
</tbody>
</table>

When running the model the class imbalance is evened by using the scaling parameters `scale_pos_weight`. This parameter balances the data patterns by changing the weights applied to the positive instances during training, which correspond to the minority class in the system. As suggested by Chen and Guestrin [18], this value is set to the ratio between negative and positive instances in the initial training set, during validation.

The module also implements a feature importance evaluation, which is supported by XGBoost. As described in Section 2.4.2, XGBoost uses the gradient boosting framework to iteratively build trees in order to minimize loss in the training set. The concept of feature importance is derived from identifying the features which have the largest impact during the training process, hence being the more predictive. The XGBoost library [18] supports different types of importance metrics:

- **Weight (or frequency)** – evaluates the number of times a feature is used to split a data across all trees. The weight importance score of a feature is computed by summing the number of tree splits it appears in, across all boosted trees. This metric tends to favour numerical features, which are more likely to reappear in trees;

- **Coverage** – evaluates the total or average number of observations related to this feature. This metric is computed according to the number of times an observation is decided on a leaf node based on the feature;

- **Gain** – evaluates the total or average model performance increase attained when adding a tree split based on a feature. It assesses the predictive power of each feature based on the improvement it single-handedly gives to base learners.

In the context of this work, the main metrics chosen to evaluate feature importance on the developed
models will be the average gain and weight. The combined use of these metrics is likely to do a better job of giving insights to the importance of the chosen features during the training.

3.8 Genetic Algorithm Module

Finding the best set of machine learning inner model parameters, or hyperparameters, can be a slow and difficult process. This is partially caused by the time overhead involved in evaluating a solution, which involves training a model on data and evaluating it on a validation set. In particular, due to the large number of parameters to be tuned in XGBoost, this problem is exacerbated.

Typical search algorithms used include grid search and random search [37]. In grid-search, the solution space is searched exhaustively, which causes even more overhead with increased number of parameters to tune, suffering from the curse of dimensionality. While random search is faster, it often does not return the best results. On the other hand, the Genetic Algorithm (GA), described in Section 2.5, has proven to show the best results in problems with wide parameter search spaces, both in terms of time and quality of solutions [37]. As such, it has been recently used in the optimization of neural networks and gradient boosting algorithms parameters [30, 38]. For these reasons, the GA will be implemented as the search algorithm to select the model parameters of the system.

The GA module is implemented using the Distributed Evolutionary Algorithms in Python (DEAP) library, which provides the base for GA Python prototyping, such as out of the box support for most crossover, selection and mutation methods, as well as support for custom data types when creating the individuals. The implementation flow of the GA module follows the one presented on the diagram of Fig. 3.7.

The specific chromosomes of the individuals, which correspond to the parameters defined in Section 3.7, must first be defined. The set of parameters to be optimized are encoded in a chromosome as a fixed length array of size 9, composed of floats and ints, as represented by Fig. 3.8.

In order to generate the population, each of the values is initialized from an uniform distribution within the reasonable bounds for each parameter, as described in the XGBoost documentation. The initial population was set to 100 individuals. At each iteration, the newly created offspring are evaluated in terms of fitness. The fitness function is obtained by fitting an XGBoost model using the parameters encoded in an individual, which is handled by the XGBoost module described in Section 3.7. The model is fit to the initial training data and the respective performance is evaluated on the validation set, from which the fitness values are obtained for a particular setup of parameters. The performance score reflects the extrapolation capabilities of the model under a set of parameters, diagnosing underfitting and overfitting of the model. The system implements two distinct evaluation metrics, which take in the raw model probability scores given to validation set observations and return a single score – to be
Start
Generate initial random population with N = 100
Evaluate fitness of each individual, by fitting and scoring XGBoost and assessing performance on validation data, according to metric chosen
Store best individual
Select individuals from mating pool, with tournament selection
Perform uniform crossover
Perform mutation
Replace old population by generated individuals

Yes

No

Start
Generate initial random population with N = 100
Evaluate fitness of each individual, by fitting and scoring XGBoost and assessing performance on validation data, according to metric chosen
Store best individual
Select individuals from mating pool, with tournament selection
Perform uniform crossover
Perform mutation
Replace old population by generated individuals

Stopping criteria met?

Yes

No

Start
Generate initial random population with N = 100
Evaluate fitness of each individual, by fitting and scoring XGBoost and assessing performance on validation data, according to metric chosen
Store best individual
Select individuals from mating pool, with tournament selection
Perform uniform crossover
Perform mutation
Replace old population by generated individuals

Figure 3.7: Flowchart for the GA execution, as implemented by the GA module.

Figure 3.8: Encoding of Genetic algorithm chromosomes. The 9 genes, represented by either integers or floats, contain the XGBoost hyperparameters to be optimized.

maximized. The two scoring metrics employed were the ROC AUC and the Precision-Recall AUC.

The ROC takes in the probability outputs of the classifier and represents the tradeoff between the true positive rate and false positive rate of predictions at different classification thresholds, in the shape of a curve. At each threshold, the higher the true positive rate is and the lower the false positive rate is, the better the system discriminates between classes and the larger the area under the curve will be.
One of the main advantages of using the ROC AUC in the context of this work is the fact that it diagnoses how well the model ranks predictions, meaning that a high ROC AUC score is associated with an higher probability of ranking a random positive observation higher than a negative observation.

The Precision-Recall AUC combines precision and recall in a single scoring metric. Precision is the rate of true positives out of all positive predictions made, whereas Recall represents the rate of true positives out of all the positives in the sample. The Precision-Recall curve, similarly to the ROC curve, represents the precision and recall values plotted at different binary classification thresholds. The AUC score follows the same intuition of the ROC AUC, but is considered to be a more suited method when the positive class is more relevant or in imbalanced data classification problems, since it is based on the precision and recall scores.

Both evaluation metrics are therefore useful since they take in the scoring values of the model and not the prediction itself, as well as the discrimination between classes of observations, which is the intended goal of the stock ranking system. Examples of both curves and the intuition behind the AUC score are represented in Fig. 3.9.

![ROC curve](image1)

![Precision-Recall curve](image2)

**Figure 3.9:** Examples of ROC and P-R AUC curves. Improved classifiers have larger under curve areas, which signal larger discriminative power between observations. Larger values of the TRP and FPR for the array of thresholds (conversely, the precision and recall), increase the area covered by the curve.

After the fitness is evaluated, the most fit individuals are sequentially stored – as described in the diagram of Fig. 3.7. Furthermore, the system employed the uniform crossover function and the tournament selection algorithm, with probabilities $P_c = 0.7$ and $P_m = 0.1$. These design choices were made based on a set of non-extensive trial and error tests performed, in which tournament selection and uni-
form crossover yielded showed acceptable results. The mutation step was implemented based on a custom mutation function, which is applied to each of the genes selected for mutation, by resampling the encoded value from an uniform distribution. Finally, the optimization process terminates when two of the following stopping criteria are met:

1. The maximum number of iterations, set to 50, is reached;
2. There are no score improvements for the past 5 consecutive iterations.

The module outputs the set of optimized XGBoost parameters, which can now be used to refit the model.

### 3.9 Stock Ranking Module

The stock ranking module takes in the scores given by the model to the test observations of the predictive window and ranks the dividend stocks, according to user settings.

First, the evaluation observations are coupled with the respective score given so that they can be grouped by firm and ranked. For example, if the evaluation period is set to the year 2019, the test observations include in most cases 4 quarters of each sample firm, each of the 4 scored by the model. The module then averages the score of the last $T$ test observations of every sample firm included in the evaluation set period, where $T \in \{1, 2, 3, 4\}$ is passed as an input parameter. Then, the averaged scores are listed and sorted by firm, in ascending order, with lower scores signalling lower averaged probability of dividend cuts in the following year and higher scores unsustainability of the dividend. Therefore, in the context of this work, the top ranking firms are those which were given the lower scores by the system.

After the stocks are ranked, the fraction $p$ of top ranking companies is returned by the module, where $p$ is an input parameter to the module. The overall ranking process is described in Fig. 3.10.

### 3.10 Sliding Window Module

The predictive layer modules described above implement the system features needed to score and rank the firms based on a single train-test window period. The sliding window module controls the selection of these predictive windows and also provides support for iteratively combining and updating the score rankings. This process relies on sliding the data window 1 year forward each time a model is built and the stocks are ranked, such that a different model is used for a single evaluation year, as Fig. 3.11 describes. The scores obtained during each run are sequentially updated using the moving average as the window shifts, leading to more robust rankings.

The module receives input from the user layer as to define the ranking parameter $p$, the GA objective for each run and the evaluation/test year list. $T$ is set by default to 4 on the sliding window setup, so that
Figure 3.10: Example of ranking of evaluation scores. After each of the financial observations generated by the firms is scored, the values are averaged and ordered to create the rankings for the evaluation year.

Figure 3.11: Sliding window data partitioning, with $t_{\text{evaluation}} \in [2017, 2018]$. 

$$
\text{Company X} \\
\begin{align*}
\text{Score}(X, t) &= 0.26 \\
\text{Score}(X, t - 1) &= 0.35 \\
\text{Score}(X, t - 2) &= 0.12 \\
\text{Score}(X, t - 3) &= 0.23 \\
\end{align*}
\text{Company Y} \\
\begin{align*}
\text{Score}(Y, t) &= 0.19 \\
\text{Score}(Y, t - 1) &= 0.05 \\
\text{Score}(Y, t - 2) &= 0.02 \\
\text{Score}(Y, t - 3) &= 0.10 \\
\end{align*}
\text{Company Z} \\
\begin{align*}
\text{Score}(Z, t) &= 0.56 \\
\text{Score}(Z, t - 1) &= 0.87 \\
\text{Score}(Z, t - 2) &= 0.47 \\
\text{Score}(Z, t - 3) &= 0.62 \\
\end{align*}

\begin{tabular}{|c|c|c|}
\hline
Rank & Company & Score \\
\hline
1 & Y & 0.09 \\
2 & X & 0.24 \\
3 & Z & 0.63 \\
\hline
\end{tabular}
the system evaluates, if available, all 4 quarters of that firm's financial year – as previously mentioned. The system uses the evaluation year list parameter to define the data windows, as described later in Section 3.8. Each of the windows is then used to reiterate the process described from Sections 3.8 to 3.9, generating new XGBoost models calibrated with the optimal GA parameters found for the data window and objective in use.

Since the XGBoost scaling parameter can skew the probabilities scores in each run depending on the degree of imbalance correction employed, normalization is applied by demeaning the scores in every iteration. While this correction is not necessary using one standalone window, combining the averaged scores possibly affects the overall ordering of the stocks, since the scaling used on one training-validation-test split is usually different from the next.

Each iteration of the sliding window can be summarized as follows:

1. The evaluation year $t_{\text{evaluation}}$ is retrieved, and the data training-validation-testing split is performed accordingly;

2. The GA is used to find the best XGBoost hyperparameters, according to the predictive performance of the probabilities given to validation period data;

3. XGBoost is trained with the obtained parameters, using the full training data;

4. The model is used to score evaluation period observations. The resulting probability scores given to observations are normalized and used to rank the stocks;

5. The standalone ranking scores are filtered, such that only $p$ percent of the top ranking stocks are kept. Smaller values of $p$ increase the elitism in selection, dropping the bottom $1 - p$ portion of the worst scored companies each year;

6. The resulting stocks are intersected with the rankings obtained in the previous evaluation years, such that only firms that ranked in the top $p$ percent in previous evaluation rounds are carried over;

7. The scores are then updated according to the moving average, so that the current score of a company is averaged with the past scores given to that same company. This operation is equivalent to equal weighted ensembling of the previous XGBoost classifiers, in which classification scores are changed iteratively based on new models added. Fig. 3.12 describes this process, for 2 evaluation years.

The results lead to a robust ranking of stocks based on multiple models trained year after year. Additionally, it allows to only select stocks which ranked in the top $p$ percent for all evaluation years. This dynamic shifting of rankings can be used to monitor portfolios, removing or adding stocks based on the yearly updates.
### 3.11 User Interface Module

The UI module is designed to allow the system user to set relevant system parameters and receive information throughout usage, using the *Jupyter Notebooks* application environment.

The system implements a verbose option that, when activated, logs and reports details of the operations executed. The data layer reports back to the user the firms being processed during data extraction, and in case the system is updating the data, the progress and data periods being retrieved. On the other hand, the predictive layer reports in detail the transformations applied to prepare the data for modelling, including the number of observations that match the applied criteria, the rows dropped and the column features to be used. This module also reports in detail the data window years being used, as well as the progression of the GA search of the optimal set of model parameters and relevant population statistics such as maximum and average fitness values. After one sliding window iteration is concluded, the system outputs the ordered rankings generated by the system, the single scores and moving average scores – so that the evolution of values can be monitored.

Conversely, the user module allows the user to input parameters into the system. For the data layer, the user is given the power to force an update to the data sample, by scraping newly issued data and re-generating all financial ratios. With respect to the predictive layer, the user is prompted to input the GA scoring metric, the evaluation window years used to select the data windows and the proportion $p$ of top ranking stops to return, according to the number of dividend stocks the user wants to invest in.
Results

Contents

4.1 Data Overview ......................................................... 54
4.2 Single Window Dividend Increase Stock Ranking ....................... 57
4.3 Sliding Window Dividend Increase Stock Ranking .......................... 62
4.4 Sliding Window Dividend Maintenance Stock Ranking .................... 69
4.5 Summary and Final Remarks ............................................ 71
This chapter has the goal of presenting and evaluating the results obtained from the developed system, in different scenarios. The chapter is organized as follows:

1. Data overview – summary analysis of the sample data to be used by the system, including analysis on dividend cuts and descriptive statistics on firms with ongoing dividend streaks;

2. Single window dividend increase prediction – rank and select stocks based on a single train-validation-test split, with respect to 1-year ahead dividend increase;

3. Sliding window dividend increase prediction – dynamic ranking and selection of stocks using a 3 year sliding window setup, with respect to 1-year ahead dividend increase;

4. Sliding window dividend maintenance prediction – dynamic ranking and selection of stocks using a 3 year sliding window setup, with respect to 1-year ahead dividend maintenance;

5. Summary of results and key takeaways, for each of the case studies.

### 4.1 Data Overview

In order to get a better insight on the data used by system, the filtered, modelling-ready dataset was collected, which ensured that only the sample observations belonging to companies with 3 years of consecutive dividend increase streak were under analysis.

The final data includes 12051 financial observations, from 2008 to 2020. The data is strongly imbalanced, as out of these total observations, 1152 belong to firms which break their dividend streak in the following year, while 10899 belong to firms which keep raising the dividend. The total number of companies which achieved dividend streaks during the sample period is 430.

With respect to dividend cuts, Fig. 4.1 shows that every year some of the firms with ongoing streaks dividend break the trend, with certain years presenting abnormal amounts of dividend cuts. One of such years is 2008, one of the worst for dividend cuts in the S&P500 index history, during which 80 firms of the sample dividend paying firms did not increase their dividend, with more than half cutting or ending dividend payout altogether.
Figure 4.1: Number of dividend streak breakers by year, from companies with a 3-year dividend increase streak. The number of streak interruptions which correspond to dividend reductions or omissions is also represented.

The data summary statistics are represented in table 4.1, where the count, mean and median of the features variables is measured, grouped by the target classes — representing whether or not the firms raise their dividend in the year subsequent to the respective observation, as described in Section 3.8. The univariate statistics of the features allow for \textit{a priori} insights to be extracted from the data. From the summary statistics it is observed that:

- The streak breakers have overall smaller profitability than the raiser group, in terms of the Gross Profit and EBITDA margins, as well as ROE, ROIC and risk-adjusted ROA;

- In terms of earnings quality, the Sloan ratio shows that dividend raisers have values slightly closer to zero, which signifies higher quality cash earnings instead of accruals;

- In terms of leverage, the biggest differences between groups are seen for Debt-to-EBITDA, which tends to be larger for non-increasers;

- The measures of turnover (assets, receivables and inventory turnover ratios) show that the companies that maintain their streaks tend to enjoy higher efficiency in the management of assets;

- The growth ratios support the hypothesis that non-increasers have declining sales and earnings. The growth of long-term debt also seems to be increasing for non-raisers;

- Dividend streak breakers are, on average, smaller sized firms;

- Both the Dividend Yield and DPR tend to be smaller for firms which raise their dividend. This supports the popular dividend sustainability hypothesis, which describes that abnormally high values of yield and payout rates are associated with unsustainable dividends.

Even though the summary statistics show an overall distinction between the features of the two groups, they are not able to fully capture the non-linear relationships between features and labels. In
order to find the most predictive features, other methods such as XGBoost feature importance rankings will be employed.

**Table 4.1:** Summary statistics of class observations, for firms who have an history of at least 3 years of consecutive dividend increase, from 2007 to 2020. The outliers were truncated such that only data points from the 5th to 95th percentile were used for the summary statistics.

<table>
<thead>
<tr>
<th></th>
<th>Non-Raisers</th>
<th>Raisers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Count</td>
<td>Mean</td>
</tr>
<tr>
<td>Gross Profit Margin</td>
<td>1069</td>
<td>0.401</td>
</tr>
<tr>
<td>ROE</td>
<td>1024</td>
<td>0.044</td>
</tr>
<tr>
<td>Risk adjusted ROA</td>
<td>973</td>
<td>2.860</td>
</tr>
<tr>
<td>ROIC</td>
<td>979</td>
<td>0.027</td>
</tr>
<tr>
<td>Cash Flow Margin</td>
<td>981</td>
<td>0.194</td>
</tr>
<tr>
<td>EBITDA Margin</td>
<td>803</td>
<td>0.214</td>
</tr>
<tr>
<td>Current Ratio</td>
<td>814</td>
<td>1.575</td>
</tr>
<tr>
<td>Quick Ratio</td>
<td>807</td>
<td>0.974</td>
</tr>
<tr>
<td>Cash Ratio</td>
<td>830</td>
<td>0.438</td>
</tr>
<tr>
<td>Accruals Ratio</td>
<td>810</td>
<td>0.008</td>
</tr>
<tr>
<td>Sloan Ratio</td>
<td>1026</td>
<td>0.020</td>
</tr>
<tr>
<td>Debt Ratio</td>
<td>1039</td>
<td>0.629</td>
</tr>
<tr>
<td>Debt-to-Equity Ratio</td>
<td>1023</td>
<td>2.343</td>
</tr>
<tr>
<td>LTD-to-Assets</td>
<td>1026</td>
<td>0.223</td>
</tr>
<tr>
<td>Asset Turnover Ratio</td>
<td>1107</td>
<td>0.166</td>
</tr>
<tr>
<td>Inventory Turnover Ratio</td>
<td>734</td>
<td>3.580</td>
</tr>
<tr>
<td>Receivables Turnover Ratio</td>
<td>973</td>
<td>1.975</td>
</tr>
<tr>
<td>Revenue Growth</td>
<td>946</td>
<td>0.039</td>
</tr>
<tr>
<td>Net Income Growth</td>
<td>1012</td>
<td>-0.079</td>
</tr>
<tr>
<td>Leverage Growth</td>
<td>1000</td>
<td>0.011</td>
</tr>
<tr>
<td>LTD Growth</td>
<td>977</td>
<td>0.121</td>
</tr>
<tr>
<td>Debt-to-Equity Growth</td>
<td>1008</td>
<td>0.034</td>
</tr>
<tr>
<td>Sustainable Growth Rate</td>
<td>1023</td>
<td>0.108</td>
</tr>
<tr>
<td>Dividend Growth</td>
<td>960</td>
<td>0.152</td>
</tr>
<tr>
<td>Market-to-Book Ratio</td>
<td>1052</td>
<td>2.942</td>
</tr>
<tr>
<td>Dividend Yield</td>
<td>860</td>
<td>0.026</td>
</tr>
<tr>
<td>DPR</td>
<td>961</td>
<td>0.434</td>
</tr>
<tr>
<td>Dividend-to-Assets</td>
<td>1017</td>
<td>0.023</td>
</tr>
<tr>
<td>Dividend-to-FCF</td>
<td>1001</td>
<td>0.921</td>
</tr>
</tbody>
</table>
4.2 Single Window Dividend Increase Stock Ranking

In this case study, the system was used to score dividend stocks based on a single training window. The window was set with the goal of scoring financial observations issued during 2018, while performing training and validation on the preceding years - as described by Fig. 3.6. The predictions were then evaluated for 2019, both in terms of dividend policy change and stock returns, the last fully observable year in the time of writing. The test sample included the total of 313 firms which paid increasing annualized dividends 3 years up to the end of 2018. From this set of firms, a subset of 15 did not manage to increase their dividend throughout 2019.

Two distinct models were obtained, trained using the best hyperparameters returned by each of the two GA fitness metrics, and then used to score the stocks. The results generated by the system are represented by Figures 4.2, 4.3 and 4.4. Figures 4.2(a) and 4.2(b) show respectively the top and bottom ranking stocks according to the scores using the ROC AUC fitness metric, while Fig. 4.3(b) and 4.3(a) show the top and bottom ranking stocks according to the scores using the Precision-Recall AUC fitness metric. A comparison between the results shows that the scores given by the GA with Precision-Recall AUC objective are higher than those obtained with the ROC AUC, which is consequence of the higher importance given to the positive class observations.

In order to evaluate the quality of the results, the ranking positions of the 15 firms that did not increase their dividend is assessed. The placement of these stocks is key to the performance evaluation of the
Figure 4.3: Overview of rankings obtained for $t_{\text{evaluation}} = 2018$, using the Precision-Recall AUC GA objective. The highlighted bars represent firms which break the dividend streak in 2019.

Figure 4.4: Cumulative Distribution of streak breakers in the single window ranking list, for the ROC AUC and Precision-Recall AUC fitnesses. The distribution is represented as function of the percentage $p$ of top ranking stocks returned.
25% includes 9 out of the 15 dividend streak breakers in both cases. With the ROC AUC objective, it is noticeable that in the tail end there is a larger concentration of streak breaking firms, with approximately half of them ranking in the lower 10%. In general, the results show that the models developed by the system have a strong discriminative power, since by selecting among the top 50% ranking firms we would be able to pre-emptively avoid the selection of the majority of companies which are going to end their dividend streak in the following year.

In the obtained results, there are specific cases worth analyzing, such as the firm Accenture plc (ACN), which is the only firm ranked above 20% that does not raise their dividend in the following year. Indeed, the company reduces their per-share dividend in $0.53 - which corresponds to a cut of 19% with respect to 2018. A thorough analysis showed that the firm changed their distribution schedule from bi-annual to quarterly dividend payouts, in the end of 2019. While this policy shift caused their dividend to be diluted throughout the year, which made the last dividend payout of 2019 to be smaller, it is not a distinctive signal of underperformance of the company - which was proven by the solid earnings coverage of dividends maintained and stock price growth. Additionally, at the time of writing, the current Trailing Twelve Months (TTM) dividend shows a remarkable 41.6% dividend growth with respect to 2019. The total price and dividend Return on Investment (ROI), given by

\[
\text{ROI} = \frac{\text{Current Dividend Adjusted Price} - \text{Initial Price}}{\text{Initial Price}},
\]

was computed for the top ranking stocks and compared with the S&P500 index annual returns. In order to do so, the price weighted average value of several portfolios composed of the \( n \) top ranking stocks was computed, for \( n = 10, 25, 50 \) and 100. The portfolios were generated at the end of the evaluation period, and the adjusted ROI was computed throughout the following year.

Fig. 4.5 shows the returns of the defined portfolios using the top ranking stocks, for the models obtained from both ROC AUC and Precision-Recall AUC GA fitness. On the one hand, figure 4.5(a) shows that the least diversified portfolios with the top 10 and 25 ranking firms generated returns as much as 10% above S&P500 compound returns. On the other, Fig. 4.5(b) shows that the top 10 and 25 returns underperform those generated by the top 50 and 100.
Figure 4.5: Returns of top scoring dividend stocks for $t_{\text{evaluation}} = 2018$, benchmarked against S&P500 index returns in the following year. The returns are adjusted for dividends and splits.
As a way to identify the financial determinants of dividend sustainability, the feature importance rankings were extracted from the XGBoost module, for both models trained. The average gain and weight of the 15 top ranking features is displayed in Fig. 4.6. Each of the feature bars is plotted as a function of F-Score, which is equivalent to the importance score given by XGBoost and is computed differently for each of the importance types, following the description given in Section 3.7.

By analyzing the importance plots, it can be stated that:

• Even though their total frequency in the boosted trees is low, as shown by the weight importance plots, the sectors variables seem to be helpful when integrated in the boosted trees. This is shown in particular by their high average gain, which shows that adding a split based on specific sectors greatly increases the predictions. One interesting results lies in the fact that the inclusion in the Real Estate sector is considered as a strong predictor of future dividend policy. This may be explained by the fact that a large percentage of Real Estate firms are Real Estate Investment Trusts (REIT), which are mandated by law to distribute 90% of their income to shareholders and follow overall different dividend policies from regular dividend paying firms;

• In terms of sole performance increase, the features that added the largest improvements in tree splits include the dividend yield, revenue (or sales) growth, firm size debt-to-EBITDA and current ratio. This comes as no surprise, as sales, leverage and yield are well documented drivers of dividend policy known to be followed by most investors as well as in literature, as discussed in Chapter 2.6;

• In terms of frequency of feature usage the weight importance plots show dividend yield, firm size and sales growth are also among the most occurring features in boosted trees. Measures of activity (assets, receivables and inventory turnover), dividend-to-assets, LTD-to-assets, gross profit margins and Market to Book Ratio (MBR) are also used frequently. These results favour numerical features, which is expected when using frequency based importance metrics.
4.3 Sliding Window Dividend Increase Stock Ranking

The case study discussed in this section uses the sliding window module, described in Section 3.10, to generate rankings based on multiple predictive windows of data. The module iteratively intersects and averages the previous ranking results, if available, using a moving average of the past scores to compute the rankings of year $t$, such that the scores not only evaluate the likelihood of future dividend increase but the past trend of this likelihood.

The sliding window stock ranking was executed with 3 evaluation years ($t_{\text{evaluation}} \in [2016, 2018]$) and $p$ initially set to 1, such that all stocks ranked for each evaluation period are intercepted. The
Precision-Recall AUC score was selected as the GA fitness metric for the sliding window study cases, as it yielded the best validation results in the period leading to the final evaluation year 2018. The choice of the evaluation years was done in conformity with the availability of data necessary to set up the windows, as described in Section 3.8. Since 2016 is the first year with sufficient training data for the system to generate scores, and 2018 the last year that can be fully evaluated, those were defined as the starting and ending evaluation periods. This scenario can be interpreted as an historical simulation of performance, in which the system is first executed at the end of 2016, ranking the best stocks for the following year, then repeating the process one year after and then another, readjusting the ordering of stocks along the way.

The results obtained during the execution of the sliding window are presented in Table 4.2. They show, for each iteration, the number of sample firms with ongoing streaks, the total number of streak breakers in the following year and the performance of the ranking system each year at identifying and ranking those firms. A step by step analysis of the results shows that:

1. In 2016, 314 firms were ranked, 26 breaking the dividend streak in the following year. The results show that in a single training window the system was able to score more than 90% of those firms below 50%;
2. In 2017, 286 firms were ranked, 9 breaking the dividend streak in the following year. The results show that the moving average of scores placed 7 of the streak breakers below 50%, with no streak breaking firm on the upper 25%;
3. In 2018, 272 firms were ranked, 11 breaking the dividend streak in the following year. The final 3-year moving average ranked 7 of the future streak breakers below 50%, placing a single of those firms on the top 25%.

Table 4.2: Results obtained by the rankings iteratively generated by the sliding window setup from 2016 to 2018, with \( p = 1 \). The table shows the number of firms with an ongoing streak at each evaluation year \( t_{\text{evaluation}} \), the number of those firms which break the streak in \( t_{\text{evaluation}} + 1 \) and the performance of the system in predicting them, as measured by how those firms were ranked.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>( t_{\text{evaluation}} )</th>
<th>Nr. of Firms Included</th>
<th>Streak Breakers in ( t_{\text{evaluation}} + 1 )</th>
<th>Distribution of Streak Breakers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2016</td>
<td>314</td>
<td>26</td>
<td>Top 25% 2  Top 50% 2  Low 25% 17  Low 50% 24</td>
</tr>
<tr>
<td>2</td>
<td>2017</td>
<td>286</td>
<td>9</td>
<td>Top 25% 0  Top 50% 2  Low 25% 4  Low 50% 7</td>
</tr>
<tr>
<td>3</td>
<td>2018</td>
<td>272</td>
<td>10</td>
<td>Top 25% 1  Top 50% 3  Low 25% 6  Low 50% 7</td>
</tr>
</tbody>
</table>

To compute the yearly returns of the top ranking stocks obtained with the sliding window, several price-weighted portfolios containing those stocks were set up in the end of each evaluation year, similarly to Section 4.2. The resulting plots are represented in Fig. 4.7, each representing the returns of the selected groups of stocks in the year following the update of the stock rankings. Fig. 4.7(a) shows the...
returns obtained during 2017, in which the selected dividend stocks return from 2 to 7% above S&P500, which presents an annual return of 18%. Fig. 4.7(b) shows that the year 2018 yielded overall negative returns, with the S&P500 falling over 8%. The selected stocks followed the same trend, although the top 25 group was able to close the year with ROI = 0. Finally, Fig. 4.7(c) shows the returns for 2019, in which the selected groups were able to again beat the compound S&P500 returns.

The sliding window uses the elitism parameter $p$ to drop the bottom ranking stocks for each evaluation year, before intersecting them with past rankings. While setting $p = 1$ averages all the model scores, using smaller thresholds is likely to affect the ordering of stocks returned. In order to test the impact of this parameter in the results, the system was tested with a range of increasing equally spaced $p$ thresholds. The performance of the final rankings generated by the system was evaluated for 2019, similarly to Section 4.2, by checking the number and distribution of dividend streak breakers. Fig. 4.8 represents two examples of stock rankings with lower thresholds ($p = 0.1$ and $p = 0.2$), while the overall results are displayed in Table 4.3. This table shows the number of companies included in the final rankings, as well as the number and distribution of those which break their dividend streak in 2019, for increasing values of $p$.

One logic caveat of using this setup is the fact that the ranks generated include a smaller number of companies. While using a single window keeps in sample all the test companies that increased the dividend for 3 previous consecutive years, the sliding window intersects the stocks in the top $p$ percent at each iteration, which diminishes the amount of selected firms the lower $p$ is, as observed in Table 4.3. Additionally, the number of evaluation years of the sliding window forces the past dividend streak requirement to be higher. For example, firms with a 3-year dividend streak as of 2016 must also be in the rankings as of 2018, which only allows firms with a 6-year streak in 2018 to fulfill. This phenomenon is already observed in Table 4.2, as the number of firms logically decreases in year $t_{\text{evaluation}}$ by approximately the same amount of firms that break the dividend streak from $t_{\text{evaluation}} - 1$ to $t_{\text{evaluation}}$. In addition, Tables 4.2 and 4.3 show that at most 10 streak breakers were included in the sliding window rankings in 2018, less than the original 15 studied in the single window case study, meaning that these 5 firms did not have 3-year streaks as of the end of the previous evaluation years. This confirmed that firms with shorter streaks were indeed prone to dividend streak interruptions.

The results from Table 4.3 show that, for $p \leq 0.5$, only 1 firm selected by the system breaks the dividend streak in the following year. This number increases to 2 until $p = 0.7$, while the overall number of selected stocks greatly increases. From then on, with $p > 0.7$, the number of non-increasers grows steeply, until $p = 1$. At this point, with no elitism in use, the results show that among the 10 streak breakers included, 7 rank below the 50% mark, which corresponds to the final results from Table 4.2.

Overall, the results suggest that dropping 30% to 50% of the bottom ranking firms at every iteration, such that $p \in [0.5, 0.7]$, yields a good tradeoff between diversity of stocks for selection and dividend cut
risk reduction. This threshold would allow to select up to a half of the stocks, while avoiding 80% of those which will cease increasing their payouts in the following year.

**Table 4.3:** Results of the sliding window rankings tested for a range of $p$ thresholds, with evaluation years $t_{\text{evaluation}} \in [2016, 2018]$. For each evaluation year included in the list, only the top $p$ percent firms are retained. The results are evaluated in terms of the number and distribution of selected firms which do not increase their dividend in the year $t + 1$, where $t$ is the final year evaluated by the system.

<table>
<thead>
<tr>
<th>$p$</th>
<th>Nr. of Firms Included</th>
<th>Streak Breakers in $t + 1$</th>
<th>Distribution of Streak Breakers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Top 25%</td>
</tr>
<tr>
<td>0.1</td>
<td>7</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.2</td>
<td>24</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.3</td>
<td>38</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.4</td>
<td>64</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.5</td>
<td>93</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0.6</td>
<td>113</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>0.7</td>
<td>148</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>0.8</td>
<td>184</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>0.9</td>
<td>227</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>272</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

The stock rankings produced by the sliding window module were also benchmarked against the standard single window setup presented in Section 4.2, by assessing the cumulative distribution of firms that break the dividend streak among the $n$ best. This comparison allows to study if it is preferable to select the $n$ top ranking stocks using a single training window or combining multiple predictive windows, for different values of $n$. In the case of the sliding widow, even though smaller $p$ thresholds limit the number of stocks returned, the cumulative distributions of two setups for $p < 1$ were tested.

From the results depicted in figure 4.9:

- Using the sliding window with $p = 0.6$, for $n = 100$, a single non-increaser is selected by the system. This a slight improvement over a single model, which selects two non-increasers for $n = 100$.

- Using the sliding window with $p = 0.8$, for $n < 120$, only 1 non-increaser is selected by the system, which is also a slight improvement over the single window. For $120 < n < 150$ the results worsen, since 3 of these firms are selected by the sliding window. For $n > 150$, the results show that picking the $n$ best stocks for $p = 0.8$ or from the single window returns similar results.

- The sliding window with $p = 1$ yields the larger number of ranked stocks for this setup. The results are similar to those obtained with $p = 0.8$ for $n < 180$, but short after there is a spike in the number of streak breakers selected. This spike shows that, if we were to select a large number of stocks such as $n > 200$, a single training window would provide the best results.

On the one hand, the sliding window combination achieves some improvements by intersecting the rankings and increasing the dividend streak requirements, which decreases the number of non-
increasers included in the rankings. On the other hand, the results show that, when selecting up to 200 stocks, the moving average of model scores does not provide significant improvements over using a single model to rank the stocks. This supports that, for the evaluation period tested, a single window of training is able to achieve very similar positive results in identifying the unsustainable dividend stocks.

Additionally, when comparing the final results obtained using the sliding window (Fig. 4.7(c)) with the single training window returns (Fig. 4.5), it is also shown that the stocks selected by the latter still outperform those selected by the former, which return at most 33%.

Since the models trained and evaluated one year before dividend streak interruptions are able to discriminate between dividend sustainers and cutters at least as well as the combination of the model scores trained in the past 3 years, an argument can be made that recent financial distress events had the most impact in the decision to stop increasing dividends in 2019. This is often the case, as sudden changes in sales or increasing leverage often force the companies to shift their policies to quickly respond to the needs to preserve cash.
(a) Sliding Window 2017 ROI.

(b) Sliding Window 2018 ROI.

(c) Sliding Window 2019 ROI.

**Figure 4.7:** Returns of top ranking dividend stock, benchmarked against S&P500 index returns in the year following evaluation, using the sliding window setup with $p = 1$. The returns are adjusted for dividends and splits.
Figure 4.8: Final normalized rankings generated using the sliding window setup, with \( t_{\text{window}} \in [2016,2018] \) and low threshold values. They represent the dividend stocks classified as the most sustainable, by the end of 2018.

(a) Sliding window with \( p = 0.1 \).

(b) Sliding window with \( p = 0.2 \).

Figure 4.9: Cumulative distribution of 2019 streak breakers between the \( n \) best stocks selected by the single window rankings and the sliding window moving averaged rankings, for a subset of \( p \) values. For clarity, only 3 of the sliding window setups were displayed, due to the similarity between the rankings shown and those obtained for other setups.
4.4 Sliding Window Dividend Maintenance Stock Ranking

Until this point, the concept of dividend streak used was based on consecutive dividend increases, similar to the S&P Dividend Aristocrats index methodology. To further test the capabilities of the system, the definition of dividend streak for the sample observations was modified to the maintenance of the current dividend. This imposes that every observation included in the data has instead a 3-year streak of dividend maintenance - meaning that the dividend did not decrease in the 3 consecutive years leading up to the year of the data observation. Additionally, each observation was also labeled in terms of future dividend maintenance, e.g. a data pattern is only labeled as positive if it belongs to a firm which will decrease their dividend the following year. Applying this modification, the system will instead be trained on learning to identify strictly the dividend cuts.

Similarly to the previous section, the sliding window setup is employed with 3 evaluation years, such that \( t_{\text{evaluation}} \in [2016, 2018] \), using the Precision-Recall AUC GA fitness scoring. In order to first test the baseline performance of the system, \( p \) was initially set to 1, similarly to the previous case study. Table 4.4 shows results obtained during execution of the sliding window. The first difference worth referencing is the greater number of firms included in the rankings, which is a direct consequence of allowing firms which do not increase their dividend nor cut it to be included. Conversely, the number of streak breakers is considerably lower, consequence of not considering the companies which do not increase their dividends as streak breakers.

A step by step analysis of the table further shows that:

1. In 2016, 382 firms were ranked, 19 breaking the dividend streak in the following year. The results showed that a single training window scores 11 of those firms under 50%, with 6 of them on the bottom 25%;

2. In 2017, 359 firms were ranked, only 2 breaking the dividend streak in the following year, 1 of which ranking under 50%;

3. In 2018, 390 firms were ranked, 8 breaking the dividend streak in the following year. The system ranked 6 of those cutters below 50%, with 4 ranking under 25%.

The results generated by the model show a significant discriminative power, as more than half of the dividend cutters land on the bottom 50%. In particular, looking at the final rankings generated after the last evaluation year \( t_{\text{evaluation}} = 2018 \), the 4 dividend cutters classified in the bottom 25% include the only firm which would end up ceasing payouts in the following year (GME), as well as the two larger cutters of 2019 (NLSN and PBI), which cut their dividend in 20% and 73%, respectively. This shows the system was able to detect beforehand the worst dividend traps, scoring them among the unsafest.
Table 4.4: Results obtained by the rankings iteratively generated by the sliding window setup from 2016 to 2018, with $p = 1$. The table shows the number of firms with an ongoing streak of dividend maintenance at year $t_{\text{evaluation}}$, the number of those firms which cut the dividend in $t_{\text{evaluation}} + 1$ and the performance of the system in predicting them, as measured by how those firms were ranked.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Year</th>
<th>Nr. of Firms Included</th>
<th>Streak Breakers in $t_{\text{evaluation}} + 1$</th>
<th>Distribution of Streak Breakers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2016</td>
<td>382</td>
<td>19</td>
<td>3 8 6 11</td>
</tr>
<tr>
<td>2</td>
<td>2017</td>
<td>359</td>
<td>2</td>
<td>1 1 0 1</td>
</tr>
<tr>
<td>3</td>
<td>2018</td>
<td>348</td>
<td>8</td>
<td>1 2 4 6</td>
</tr>
</tbody>
</table>

Similarly to the previous section, the price weighted returns of the top ranking firms were computed for the periods following each evaluation year. The results, represented in Fig. 4.10, show that the stocks selected by the system slightly outperform the S&P500 index in the first two years, with the best performing portfolios being those containing the top 10 ranking stocks. These portfolios return in 2017, 26% while in 2018 it takes -5% losses, which are still inferior to the overall losses of the S&P500 in the same year. In 2019, the results show that the selected dividend stocks largely outperform both the index and the returns obtained in Sections 4.2 and 4.3 during the same period, with the top 25 and 50 groups generating annual returns above 40%.

The effect of $p$ was also tested for a range of values. Fig. 4.11 represents two examples of stock rankings with lower thresholds ($p = 0.1$ and $p = 0.2$), while the overall results are displayed in Table 4.5. This table shows the number of companies included in the final rankings, as well as the number and distribution of those which cut their dividend following the last evaluation year (2019), for increasing values of $p$.

For $p < 0.5$, the system selects at most approximately 100 stocks, which include a single dividend cutting firm. This particular company is again ACN, which despite the apparent cut is still an overperforming dividend stock with strong fundamentals, hence the placing given by the model $^1$. For $0.5 \leq p \leq 0.8$, the number of cutters increases by 1 while the overall number of firms also doubles. With $p > 0.8$ the number of dividend cutters spikes, doubling for every 0.1 increase of $p$.

Overall, the results have shown very positive stock selection outcomes, in particular for $p \leq 0.8$. This threshold offered a good tradeoff between number and diversity of selected firms for investment and risk of experiencing a dividend cut in the following year. For comparison, this means that the system was able to select up to 226 out of the total 348 stocks of firms with ongoing dividend maintenance streaks, excluding 75% of the firms that would end up cutting their dividend in the following year.

---

$^1$Refer back to Section 4.2 for a detailed description on this case.
Table 4.5: Results for the sliding window setup, executed for a range of $p$ values, with evaluation window $t \in [2016, 2018]$. During each evaluation year included in the window, only the top $p$ percent firms are retained. The results are evaluated in terms of the number and distribution of selected firms which cut their dividend in the year $t + 1$, where $t$ is the final simulation window year.

<table>
<thead>
<tr>
<th>$p$</th>
<th>Nr. of Firms Included</th>
<th>Streak Breakers in $t + 1$</th>
<th>Distribution of Streak Breakers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Top 25%</td>
</tr>
<tr>
<td>0.1</td>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0.2</td>
<td>21</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0.3</td>
<td>47</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0.4</td>
<td>79</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0.5</td>
<td>104</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0.6</td>
<td>137</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>0.7</td>
<td>178</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>0.8</td>
<td>226</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>0.9</td>
<td>279</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>348</td>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>

4.5 Summary and Final Remarks

In this section, the standout results obtained are highlighted and summarized, for each of the three case studies:

1. **Single Training Window Ranking, with respect to dividend increase probability** – Two models were trained using different GA fitness metrics, and used to score the probabilities that the sample dividend stocks break their dividend, based on 2018 evaluation data. In terms of top ranking firms, the best results showed that, using the Precision-Recall AUC, only 3 out of the 15 dividend streak breakers rank among the best 50%. In terms of returns, the best results were obtained using the ROC AUC objective, with the selected stocks generating as much as 40% of total returns – 10% more than the S&P500;

2. **Sliding Window Ranking, with respect to dividend increase probability** – Using only the Precision-Recall AUC, 3 XGBoost models were trained for the evaluation years of 2016, 2017 and 2018. The final combined scores showed that for the top half ranked stocks, 3 streak breakers were selected by the system, similarly to the first case study. In terms of returns, the best ROI obtained was 33% – which suggests that the combined use of XGBoost models did not improve returns. In order to further test the sliding window, elitism and filtering of stocks was introduced for each evaluation year. The comparative results between the sliding window, with different degrees of elitism, and single window did not show significant differences – depending on the specific number of stocks to be selected, advantage is given to one or another setup. Overall, no evidence is found that using multiple ensembled models improves the results;

3. **Sliding Window Ranking, with respect to dividend maintenance probability** – This case study,
although not directly comparable to the previous, employed a similar methodology to dividend maintaining firms, predicting instead solely dividend cuts. In this case, the results show that the best 100 stocks returned by the system a single one cuts the dividend in the following year, while the other 7 firms that cut or omit their dividend payouts are avoided. In terms of total returns, for 2019, the best stocks selected in this case study largely outperform those obtained in the previous case studies – generating a maximum ROI of 44%.
Figure 4.10: Returns of top ranking dividend stocks in terms of dividend maintenance, benchmarked against S&P500 index returns in the year following evaluation, using the sliding window setup with $p = 1$. The returns are adjusted for dividends and splits.
Figure 4.11: Final normalized rankings generated using the sliding window setup, with $t_{\text{window}} \in [2016, 2018]$ and low threshold values. They represent the dividend stocks classified as the less likely to cut their dividend, by the end of 2018.
Conclusion

Contents

5.1 Future Work ................................................................. 77
The results obtained from the developed system lead to the conclusion that XGBoost can achieve outstanding results in the field of dividend policy prediction. The downsides of XGBoost complexity and wide parameter space were successfully surpassed with the application of GA optimization, which allowed to find multiple sets of optimal parameters for different predictive windows – which ultimately facilitated the construction of multiple refined models. These XGBoost models were used to score several evaluation years, in which dividend stocks were ranked in terms of dividend sustainability.

First, using a single model to rank stocks with respect to sustainability of future dividend increase, the results obtained were very positive for both of the two GA objective functions tested. While in terms of returns the ROC AUC yielded the best results, the Precision-Recall AUC metric returned a smaller number of streak breakers for a larger percentage of top ranking stocks, placing 80% of these firms below 50%.

The results of this case study were used to investigate which data features were the most essential for making predictions. Strong evidence is found that dividend yield, revenue growth and firm size are among the most effective predictors. Other relevant features which influence future dividend sustainability include industry sector, leverage, turnover and liquidity measures.

Furthermore, the sliding window setup combined the rankings generated by each XGBoost model iteratively for 2016, 2017 and 2018. The intersection of the top $p$ ranking firms for each evaluation year made the system selections more elitist, reducing the overall number of firms ranked and filtering some of the firms with stale dividends - which brought by itself some improvement. The final stock rankings were evaluated by assessing performance of the scored stocks in terms of dividends and overall returns, throughout 2019. Overall, selecting up to 200 stocks, the results were very similar between the setups tested, with small differences in terms of streak breakers selected, which gave in some cases advantages to the combined use of XGBoost models. In some other cases, when selecting a large number of stocks, it was preferable to use a single model. The fact that a single training window suffices to provide safe stock selections supports the argument that the most recent scores are the most relevant, as more signals of financial distress can be observed in the year preceding dividend cuts. This suggests future approaches where it might be more valuable to add more weight to recent scores.

Finally, the sliding window setup was also adapted for dividend maintenance streaks, predicting only dividend cuts. The results showed that after running all evaluation years, most firms which end up cutting their dividend in the following year are not selected by the system. In particular, by setting the threshold $p = 0.8$, 65% of the total maintaining firms with a 3-year history of stable dividends would be selected, while more than 75% of the cutting firms would be excluded.

In general, the approach taken of combining binary multiple classification scores to rank stocks showed robust results and the developed system proved to be a reliable tool to select safe dividend stocks, fulfilling all of the initial requirements.
5.1 Future Work

In the future, the system could be improved by expanding some of its features or adding novel ones. These potential enhancements include:

- Extending the study to quantify the future dividend increase (or cut) amount or percentage change. In this context, one could apply for example the XGBoost regressor and rank the stocks with regard to their future growth potential;

- Testing of other weighting methods for the sliding window other than model score average. Weighting each scored evaluation year differently is likely to impact the overall performance of the stock rankings;

- Explore different sets of financial features, or potentially create new features around the most relevant identified by this work. Extensive feature engineering is likely to further improve the results;

- Test different settings of GA configurations, exploring other parameter settings more exhaustively (i.e., population size, crossover and mutation rates). Even though some selection and crossover functions were comparatively tested in the context of this work in order to select the settings of the GA, due to the time complexity of the algorithm and the resources available, other setups of the algorithm were not explored.
Bibliography


