Evaluating Score-Investing Methodologies: A Systematic Review of Tweenvest's Algorithm for Long-Term Stock Investing Using Descriptive Analytics and Predictive Modeling.

Trabajo Fin de Grado Integrado
Grado en Administración y Dirección de Empresas
Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Author: Carlos Eduardo Domínguez Martínez

Tutor: Joaquín Martínez Minaya (GADE)

Tutor: Alberto Albiol Colmener (GITST)

Extern Co-Tutor: José Tatay Sangüesa (Tweenvest)

Course: 2024-2025

Valencia, June 2025

Escuela Técnica Superior de Ingeniería de Telecomunicación
Universitat Politècnica de València
Edificio 4D. Camino de Vera, s/n, 46022 Valencia
Tel. +34 96 387 71 90, ext. 77190
www.etsit.upv.es

Abstract

This study investigates the effectiveness of a factor-investing methodology developed by Tweenvest, leveraging a proprietary algorithm grounded in fundamental financial analysis. The algorithm scores companies across four key factors: Quality, Growth, Value, and Dividends.

The research aims to evaluate the profitability of investment strategies based on these scores over multiple profitability horizons from 1 month to five years. A comprehensive dataset was constructed, integrating factor scores with additional variables such as sector and geographic region, standardized for currency and timeframes.

Statistical analyses will explore relationships between these factors and returns, identifying optimal investment periods. Subsequently, predictive modeling—including econometric regressions, time series models, and neural networks—will be applied to assess the translation of these factors into market performance.

The study incorporates an interdisciplinary approach, combining financial theory and econometrics with advanced programming, data engineering, and artificial intelligence. This integration bridges Business Management and Telecommunications Engineering, offering insights into the practical application of Tweenvest's scoring algorithm and contributing to the advancement of financial technology analytics.

To my parents.

Contents

List of Figures

List of Tables

Part I

Main Report

Chapter 1

Introduction to Fundamental Analysis

1.1 Definition

Fundamental Analysis is a methodology used to evaluate the intrinsic value of a company, asset, or market by analyzing various economic, financial, and qualitative and quantitative factors. Unlike technical analysis, which focuses on price movements and chart patterns, fundamental analysis seeks to determine an asset's "true value" to identify investment opportunities that may be undervalued or overvalued in the market.

This intrinsic value is defined by numerous experts and renowned investors, including Benjamin Graham, Warren Buffett, and Pat Dorsey [II]; and many Stock's Index such as IBEX-35 or MSCI, as a company's ability to adapt to an ever-changing environment while adding value to the market and establishing barriers to entry for competitors—commonly known as economic moats.

Measuring these moats is challenging due to the difficulty of quantifying certain variables, such as brand strength and market influence. However, over the long term, these intangible factors translate into tangible financial data, reflected in a company's balance sheet, income statement, and cash flow statement; that when exposed to the public market share creates a need to buy or sell the stocks, altering the companies profitability. So from now on, this economic moats will be called *alpha*, following the common literature.

1.2 Tweenvest's Scores

Following this approach, Tweenvest developed its four main scores based on key focus areas that investors have to consider before making any decision:

- Profitability
- Financial Health
- Predictability
- Consistent Growth
- Entrance Moment
- Dividends Payed

1.2.1 Quality

This Tweenvest's score is approached in a similar way that many successful investors would, distinguishing on three main categories: **profitability**, **financial health**, and **predictability**. Each of them includes inside of them multiple financial ratios that take account of different relevant data within the company's reports.

To understand the complexity of this score, we need to look at each category separately. Starting with **profitability** we can separate:

Profitability Margins

These are essential for understanding how a company is managing its costs and generating profits from its revenues.

- **Net margin**: Represents net profit as a percentage of total sales, indicates a company's efficiency in generating profits after accounting for all expenses, taxes, and costs. A high net margin suggests that the company has good cost control and strong pricing power in the market.
- **Operating margin**: Measures operating profit (EBIT) as a percentage of total sales, providing a clear view of the profitability of a company's core operations, excluding interest and taxes. This metric is fundamental for evaluating a company's operational efficiency.
- **EBITDA margin**: Removes the effects of capital structure and accounting policies, offering a clear view of the company's pure operational profitability.
- **Gross margin**: Focuses on revenues after deducting the cost of goods sold, is a key measure of production efficiency and a company's ability to manage its direct costs.

Performance Ratios

These are used to measure the overall performance of the company.

- **ROA (Return on Assets)**: Measures how efficiently a company converts its assets into profits. This is especially important in capital-intensive sectors, where efficient asset management can make a significant difference in profitability.
- **ROE (Return on Equity)**: Focuses on the profits generated per dollar of equity invested by shareholders. This ratio is crucial for evaluating a company's overall profitability from the shareholders' perspective. It is an especially valuable metric for investors seeking to maximize their returns on equity investment.
- **ROIC (Return on Invested Capital)**: Focuses on the return generated by all the funds invested in the company, including both shareholders' equity and debt. It is a comprehensive measure of a company's ability to generate value from all its sources of financing.
- **ROCE (Return on Capital Employed)**: Measures the funds used to finance operations, regardless of the source. This ratio is useful for comparing the efficiency of companies with different capital structures, as it focuses on total capital employed rather than just equity.

Also, to not only look at actives Tweenvest uses the cash generated to calculate: **CROIC** (Cash Return on Invested Capital), **CROCE** (Cash Return on Capital Employed), **OCF/Sales**, and **FCF/Sales**. And lastly it takes account also the Owner's income to calculate: **Owner's Income/Sales**, **Owner's CROIC** and **Owner's CROCE**.

Continuing with the financial health, we need to analyze the company's debt in multiple aspects:

Leverage Ratios

These ratios assess how much a company relies on debt to finance its assets and operations. They are essential for evaluating financial risk and long-term solvency.

- **Financial Leverage** (Total Assets / Equity): Measures the proportion of a company's assets that are financed by shareholder equity. A higher ratio suggests the company is using more debt relative to equity, indicating greater financial risk but also potential return amplification through leverage.
- **Total Debt/Assets**: Indicates what portion of the company's assets is financed through debt. A lower ratio implies a more conservative capital structure, while a higher one may indicate increased risk if the company becomes over-leveraged.
- **Total Debt/Capital**: Measures the share of total capital (debt + equity) that comes from debt. This ratio is useful for understanding how dependent the company is on borrowed funds compared to its overall capital base.
- **Total Debt/Equity**: Compares the company's total debt to its shareholder equity. It provides insight into the balance between debt and equity financing. A high ratio may signal financial risk, but also the potential for higher returns if debt is managed well.

Debt Coverage Ratios

These metrics evaluate a company's ability to cover its debt using its earnings or cash flow. They reflect the sustainability of a company's debt in relation to its operational performance.

- **Net Debt/EBIT**: Shows how many years it would take for a company to repay its net debt using EBIT (Earnings Before Interest and Taxes).
- **Net Debt/EBITDA**: Similar to the above, but adds back depreciation and amortization. This gives a more cash-focused view of a company's ability to handle its debt load, and is especially useful for comparing companies in capital-intensive industries.
- **Net Debt/FCF**: Evaluates how many years of free cash flow would be needed to pay off net debt. Since FCF includes investment needs, this ratio gives a more conservative view of debt sustainability.
- **Net Debt/Owner's Income**: Compares net debt to the income available to equity holders (after all operating and investing costs).

Interest Coverage Ratios

These ratios measure how easily a company can meet its interest payments on outstanding debt — critical for assessing short-term debt service capability.

- **EBIT/Interest**: Indicates how many times a company can cover its interest expenses with its operating income.
- **EBITDA/Interest**: Similar to the above, but adds back depreciation and amortization. This gives a clearer picture of available cash earnings before fixed financial obligations, ideal for heavily asset-based businesses.

- **FCF/Interest**: Since FCF considers investment needs, this is a stringent test of how much real, discretionary cash is available for debt servicing.
- **Owner Earnings/Interest**: Evaluates a company's ability to meet interest payments based on the earnings effectively attributable to shareholders. It accounts for operational cash flow minus necessary capital expenditures.

Liquidity Ratios

These ratios measure a company's ability to meet short-term obligations with its short-term assets. They are essential for evaluating near-term financial health and risk of insolvency.

- **Current Ratio** (Current Assets / Current Liabilities): Shows whether a company has enough assets to cover its short-term liabilities. A value above 1 is generally considered healthy, though excessively high values may imply inefficiency.
- **Quick Ratio** ((Current Assets - Inventory) / Current Liabilities): A more stringent version of the current ratio that excludes inventory, which may not be easily liquidated. It's useful in assessing true short-term liquidity.
- **Cash Ratio** (Cash and Equivalents / Current Liabilities): The most conservative liquidity metric, focusing only on cash and equivalents. It shows the immediate solvency of a company in a worst-case scenario.
- **OCF Ratio** (Operating Cash Flow / Current Liabilities): Assesses how well the company's operational cash flows can cover its current obligations. This offers a realistic view of liquidity since it's based on actual cash generation rather than accounting figures.

Predictability

And finally, for the quality score we need to look at the company's predictability. This is achieved by trying to fit values related to the company's success —such as Sales— to a exponential curve, which is supported by large financial literature.

After calculating all of this ratios, Tweenvest compares them to the sectors' median and interpolates it to create a single score for each ratio and then aggregates them all using personalized weights to create the final Quality Score, which is then showed to the clients:

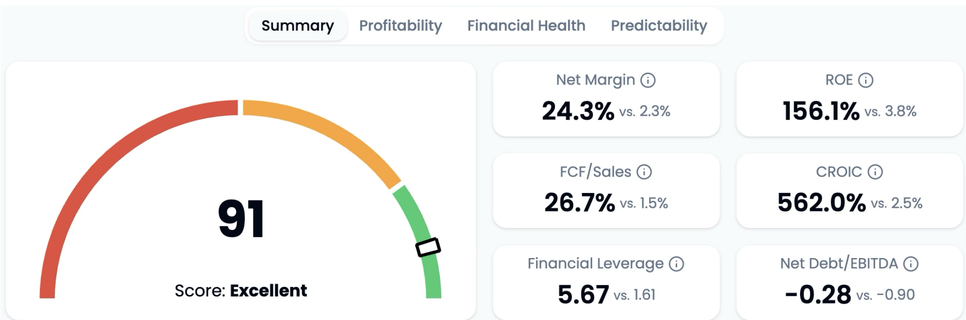

Figure 1.1: Tweenvest's Quality Score

1.2.2 Growth

The Growth Score evaluates a company's historical growth across multiple key metrics, comparing them to industry standards. This comprehensive approach ensures a balanced assessment of growth across different aspects of the business.

Revenue & Profitability Growth

- **Sales**: Measures growth in core revenue streams
- **EBITDA**: Captures growth in operational cash-generating ability before non-cash and financing impacts
- **Operating Income**: Reflects growth in profit from core business operations
- **Net Income**: Includes all income sources, showing overall profitability growth

Cash Flow Growth

- **Operating Cash Flow**: Measures growth in cash generation from operations
- **Simple FCF**: A straightforward proxy for available cash after essential investments
- **Levered/Unlevered FCF**: Provide detailed views of free cash flow with and without debt impact
- **Owner Earnings**: Useful for volatile capex cases, emphasizing cash available to shareholders

Capital Base Expansion

- **Total Assets**: Indicates expansion in overall asset base
- **Equity**: Reflects growth in shareholders' claim on the business
- **Tangible Book Value**: Highlights growth in physical net assets, excluding intangibles
- **Invested Capital**: Captures total capital being put to productive use
- **Capital Employed**: A broader measure of capital supporting business operations

Per-Share Value Growth

- **Diluted EPS**: Tracks per-share earnings growth, accounting for dilution effects
- **Diluted Shares**: Included to track share count changes, ensuring EPS growth isn't artificially inflated by buybacks or dilution
- **Ordinary DPS**: Tracks the growth of shareholder payouts, a proxy for confidence in future earnings

To compute the Growth Score, Tweenvest calculates 10-year, 5-year, and 3-year averages and then interpolates the growth rate to industry standards. This approach reinforces the long-term investment philosophy, giving lasting growing companies a better score.

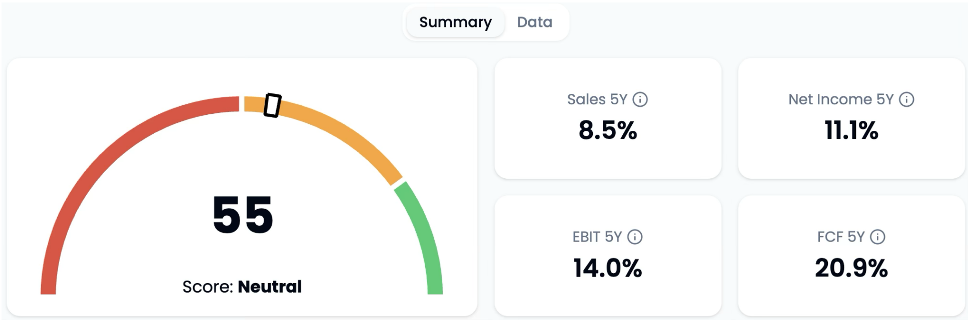

Figure 1.2: Tweenvest's Growth Score

1.2.3 Valuation

The Valuation Score measures how attractively a company is priced relative to its fundamentals such as earnings, cash flow, sales, and dividends. This is critical for investors following value investing principles, where the goal is to buy quality companies for less than their intrinsic worth. The algorithm evaluates a series of valuation multiples—both price-based and enterprise value-based—and dividend yield.

Price-Based Multiples

- **P/E** (Market Cap / Adjusted TTM Earnings): Measures how much investors are willing to pay per dollar of earnings
- **P/S** (Market Cap / TTM Revenue): Useful when earnings are volatile; shows valuation relative to sales
- **P/CF** (Market Cap / TTM Operating Cash Flow): Reflects valuation relative to cash-generating ability
- **P/B** (Market Cap / Tangible Equity): Especially relevant for asset-heavy sectors like banks or industrials

Enterprise Value-Based Multiples

- **EV/Sales** (Enterprise Value / TTM Revenue)
- **EV/EBITDA** (Enterprise Value / TTM EBITDA)
- **EV/EBIT** (Enterprise Value / TTM EBIT)
- **EV/FCF** (Enterprise Value / TTM Free Cash Flow)

Yield-Based Valuation

- **Dividend Yield (%)** (Dividend per Share / Price per Share)

Each of these ratios is compared to multiple historical statistics and sectoral benchmarks to create individual scores and then average them.

Note:

- Market Cap = Share Price × Total Outstanding Shares

- Enterprise Value = Market Capitalization + Total Debt - Cash + Marketable Securities

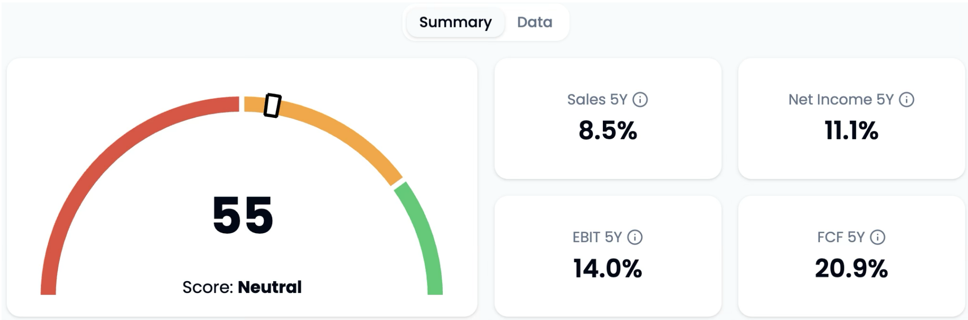

Figure 1.3: Tweenvest's Value Score

1.2.4 Dividend

When analyzing Tweenvest most common investors profile, we see a high tendency to dividend investment.

The Dividend Score measures the attractiveness, reliability, and growth potential of a company's dividend payments. It helps investors assess whether the dividend is both rewarding today and sustainable for tomorrow—a key concern for many users of the platform that are income-focused and long-term investors using mainly dividends as their principal concern. And it is built from three primary components:

Safety

- **Payout Ratio EPS** (DPS / Diluted EPS): Shows if dividends are covered by accounting earnings
- **Payout Ratio FCF** (DPS / Free Cash Flow per Share): Shows if dividends are funded by real cash generation
- **Payout Ratio Owner Earnings** (DPS / Owner Earnings per Share): A conservative test of sustainability (excludes CAPEX)

Growth

Measures how consistently and strongly the dividend has grown over time.

- **Ordinary DPS CAGR**: 3-Year, 5-Year, 10-Year growth rates

Yield

This evaluates the attractiveness of the dividend today relative to the company's historical averages and sector benchmarks.

- **Dividend Yield** (DPS / Price per Share): Represents how much income an investor receives annually from dividends

These components are individually scored weighted and interpolated against industry benchmarks to form a composite score. Finally, the algorithm adjusts this score based on how many years the dividend has been maintained or increased, rewarding consistency.

Figure 1.4: Tweenvest's Dividend Score

1.3 Problems with the current approach

In the context of this thesis, we have to explain the need to make a systematic review of the scores to check wether the simplifications and hypotheses assumed by the financial consensus truly show a company's ability to create *alpha* over time or not.

1.3.1 Unquantified Variables

Since the current model only includes variables found in a company's financial statements, there is a significant amount of relevant information being left out, for example:

- The **perceived differentiation of a product** is one such element that may not be properly captured by traditional financial analysis. A trusted brand or strong reputation can allow a company to charge premium prices and build customer loyalty, generating extraordinary long-term profits. For instance, brands like Tiffany or Rolex have high perceived value that justifies elevated prices—something that cannot be easily quantified using standard financial metrics such as profit margins or return on capital.

- Additionally, strategies such as **cost leadership** and offering products at lower prices can provide a significant competitive advantage that is not always directly reflected in financial data.

- Companies may also establish **barriers to entry** and **high switching costs** for customers, making it more difficult for them to move to competitors. These strategies may involve investments in technology, patents, or simply building long-term customer and employees relationships.

To properly measure these variables, a more exhaustive analysis of each company would be required, pulling from multiple secondary information sources such as news articles, conference transcripts, customer blogs, competitive product reviews, and more. This task is far more difficult to automate via code, as it would require multimodal AI techniques—thus, it will fall outside the scope of this work.

1.3.2 Emergent Effects in Complex Systems

In complex systems like financial markets, network effects emerge, making most relations between variables not be linear or stationary, adding noise to actual reliable data behind it:

- **Herd behavior** is a network effect where investors tend to follow the actions of others rather than rely on their own analysis. This can amplify market movements, both upward and downward. Herd behavior can lead to speculative bubbles and abrupt corrections when collective expectations shift.

- **Feedback loops** are another network effect where market participants' actions reinforce existing market behavior. For example, rising asset prices may attract more investors, which in turn drives prices even higher. This type of positive feedback can cause price escalation that becomes detached from underlying fundamentals. Conversely, negative feedback can occur during a sell-off, where falling prices trigger more selling, further accelerating the decline.

- **Macro-level influences** are another crucial factor where broader structural changes—such as political decisions, economic policies, technological shifts, or industry-wide trends—can significantly impact market behavior. These larger forces often operate beyond the scope of individual company analysis and can create ripple effects throughout the entire market ecosystem.

1.3.3 Linearity and Stationarity

Non-linearity

The relationships between financial variables are often non-linear, meaning that changes in one variable may not result in proportional changes in another. For example:

- **Economies of scale** can create non-linear relationships between production volume and costs. As a company grows, it may experience decreasing marginal costs due to better resource utilization, bulk purchasing discounts, or spreading fixed costs over larger output.

- **Market saturation** can lead to diminishing returns on marketing spend or R&D investments. Initial investments might yield significant returns, but as the market becomes saturated, additional spending may produce smaller incremental benefits.

- **Competitive dynamics** can create threshold effects where small changes in market share or pricing can trigger significant shifts in competitive position or profitability.

Trends

Traditionally, standard growth ratios have been used, based on the assumption that in competitive markets, when a new business model or opportunity emerges with above-average margins, entrepreneurs quickly move in to capitalize on it—eventually saturating the opportunity and driving margins back down to average levels over time. But what happens when there is a significant shift in trend?

Markets are "fluctuating entities", so static metrics can become problematic when underlying trends change. A clear example is the rise of artificial intelligence and the surge in stock prices of companies involved in the production and development of the necessary technologies.

Chapter 2

Objectives

The main objective of this thesis is to check if the scores actually represent an objective and accurate view of the company's ability to generate *alpha* for different time frames, and propose changes to the current algorithm if needed. To do this, we need to follow multiple steps:

1. **System Architecture Enhancement**: Modify Tweenvest's database architecture to enable historical score storage and retrieval and create the datasets for the analysis.
2. **Data Curation**: Clean and preprocess the collected data to ensure consistency and reliability, handling missing values and outliers appropriately.
3. **Exploratory Analysis**: Process and analyze the data to check if the distributions and correlations between variables, looking for early on patterns to later compare and use for the modeling.
4. **Predictive Modeling**: Develop a multiple predictive models to check what is the best way to use the scores for multiple investment strategies.
5. **Validation & Benchmarking**: Back-test the algorithms to check their performance and compare them with S&P 500 profitabilities.

Chapter 3

Methodology & Theoretical Framework

3.1 Architecture and Database

3.1.1 Code practices

Since the code that needs to be changed is used daily in a production environment by Tweenvest, it is important to follow some practices to ensure the code is easy to understand, maintain, and to avoid introducing new bugs.

- **Understanding the code**: Before starting to work on the code, it is important to understand the codebase and the purpose of the code. For this, it is recommended to use some tools like flux diagrams, code comments, and documentation. As it will be shown later on.
- **Testing**: Every function and class should have unit tests that cover all possible scenarios for not introducing new bugs.
- **Logging**: After each feature is implemented, it is important to analyze the logs to check if the feature is working as expected, and to see if there are any possible optimizations to be made in the queries to improve the performance.
- **Code review**: Whenever all of this is done, the code needs to be reviewed and approved by the team to be merged into the main branch.

3.1.2 Data creation

3.1.3 Data storage

3.2 Preprocessing

Once we have our final dataset, we need to inspect it and make sure it is consistent and ready to be used for the modeling.

3.3 Outlier Detection

Anomalies are data patterns that have different data characteristics from normal instances. The ability to detect anomalies has significant relevance, since anomalies often provide critical and actionable information in many different contexts.

For example, anomalies in credit card transactions could signify fraudulent use of credit cards. An unusual computer network traffic pattern could stand for an unauthorised access.

3.3.1 Inter Quartile Range

Given a univariate dataset, the Interquartile Rule identifies outliers based on the interquartile range (IQR). The steps are as follows:

1. Compute the first quartile (Q_1), which is the 25th percentile of the data.
2. Compute the third quartile (Q_3), which is the 75th percentile of the data.
3. Calculate the interquartile range:
$$\mathrm{IQR} = Q_3 - Q_1$$
4. Define the lower and upper bounds for non-outlier values following the 1.5 rule:
$$\text{Lower bound} = Q_1 - 1.5 \times \mathrm{IQR}$$
$$\text{Upper bound} = Q_3 + 1.5 \times \mathrm{IQR}$$
5. Any data point x is considered an outlier if:
$$x < Q_1 - 1.5 \times \mathrm{IQR} \quad \text{or} \quad x > Q_3 + 1.5 \times \mathrm{IQR}$$

As we can see this is a very simple and intuitive method, but it has some limitations:

Advantages	**Limitations**
Simple to compute and interpret.	Not suitable for multivariate data with correlated features.
Non-parametric.	May misclassify values in skewed distributions as outliers.
Robust to extreme values, since it relies on percentiles rather than the mean.	Fixed multiplier is arbitrary and may need tuning for each dataset.

Table 3.1: Advantages and Limitations of the Interquartile Rule

3.3.2 Single Vector Machine

Support Vector Machines (SVMs) represent a robust class of supervised learning algorithms that can be effectively adapted for anomaly detection tasks. This algorithm establishes itself on the premise that the majority of real-world data is inherently normal. Where the goal is to define a boundary encapsulating the normal instances in the feature space, thereby creating a region of familiarity.

To capture the bases of the scikit-learn implementation developers [3] used in this work, the principle idea is to find a sphere, of minimum volume, containing all the training samples. This sphere, described by its center c and its radius r, is obtained by solving the constrained optimization problem: [6]

$$\min_{r,c} r^2 \quad \text{subject to} \quad \|\Phi(x_i) - c\|^2 \leq r^2 \quad \text{for} \quad i = 1, 2, \ldots, n \tag{3.1}$$

This boundary is strategically positioned to maximize the margin around the normal data points, allowing for a clear delineation between what is considered ordinary and what may be deemed unusual. This emphasis on margin maximization is akin to creating a safety buffer around the normal instances, fortifying the model against the influence of potential outliers or anomalies.

In summary, this method has the following characteristics:

Pros	Cons
Effective for high-dimensional data and complex boundaries.	Sensitive to the choice of kernel and hyperparameters (e.g., ν, γ).
Works well when the training set contains mostly or only normal instances.	Computationally intensive, especially on large datasets.
Can capture nonlinear patterns using kernels (e.g., RBF kernel).	Difficult to interpret or explain the decision function.
No need for labeled data from the outlier class.	–
Solid theoretical foundation and widely supported in machine learning libraries.	–

Table 3.2: Pros and Cons of One-Class SVM for Outlier Detection

3.3.3 Isolation Forest

As we are seeing in this small fraction of outlier detection methodologies, most of the existing anomaly detection approaches are based on the premise of normal distributions, then identify anomalies as those that do not conform to the normal profile.
As it is explained in the original paper Liu, Ting, and Zhou [4], the Isolation Forest algorithm constructs multiple isolation binarytrees (iTrees) from the dataset, where anomalies are identified as data points that exhibit shorter average path lengths across these trees. The algorithm's configuration involves three key parameters:

- The number of trees to generate
- The size of the subsample
- The maximum tree height during the assessment phase

Lets consider a dataset $X = \{x_1, \ldots, x_n\}$ containing n points in d-dimensional space, and a subset $X' \subset X$. An Isolation Tree (iTree) is constructed as follows:

- Each node T in the tree is either:
  - An external node (leaf) with no children, or
  - An internal node with exactly two children (T_l and T_r) and a test condition
- The test condition at each internal node consists of:
  - A randomly selected attribute q
  - A split value p
  - A test $q < p$ that determines whether a point goes to T_l or T_r

The tree construction process recursively partitions X' by randomly selecting attributes and split values until either:

- The node contains only one instance, or
- All instances at the node have identical values

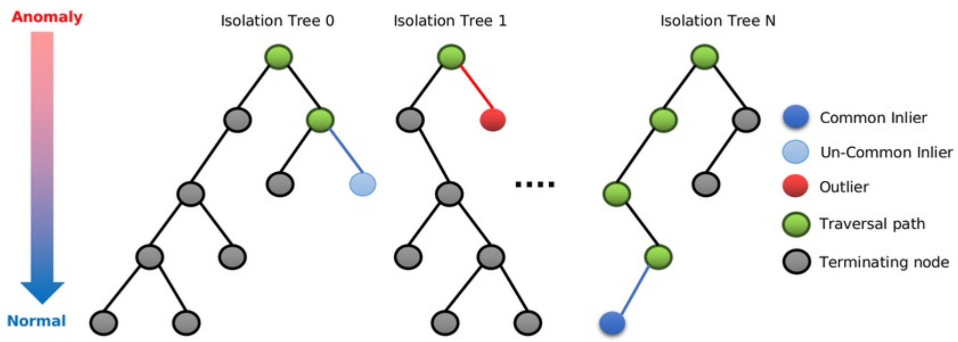

Figure 3.1: Visualization of how Isolation Forest works.

As we can see in Figure 3.1, once the iTree is fully constructed, each point $x_i \in X$ is isolated at a leaf node. The path length $h(x_i)$ of a point x_i is defined as the number of edges traversed from the root to its leaf node. Points with shorter path lengths are considered more likely to be anomalies, as they require fewer splits to be isolated from the rest of the data.

This method works very good in most cases, but it has also its own limitations

Pros	Cons
Efficient and scalable to large datasets due to its tree-based structure.	Less effective for detecting local outliers in densely clustered regions.
Non-parametric: does not assume any data distribution.	Performance can vary with the choice of parameters like number of estimators and subsample size.
Naturally handles high-dimensional data.	Not suitable for detecting subtle anomalies that closely resemble normal data.
Requires little preprocessing (no need for feature scaling).	Interpretation of results is less intuitive compared to statistical methods.
Robust to irrelevant features due to random sub-sampling.	May struggle with data that has complex, non-isolated outlier structures.

Table 3.3: Pros and Cons of the Isolation Forest Method for Outlier Detection

3.3.4 Local Outlier Factor

Since we are looking at outliers from the perspective of correlations between the different response variables to see which companies don't behave naturally. We also checked on the Local Outlier Factor (LOF) for being based on hyper-plane densities of data.

The algorithm measures the local deviation of the density of a given sample with respect to its neighbors. It is local in that the anomaly score depends on how isolated the object is with respect to the surrounding neighborhood. More precisely, locality is given by k-nearest neighbors, whose distance is used to estimate the local density. By comparing the local density of a sample to the local densities of its neighbors, one can identify samples that have a substantially lower density than their neighbors. These are considered outliers. Breunig et al. [2]

Pros	Cons
Can detect outliers relative to their local neighborhood, allowing it to identify context-specific anomalies.	LOF scores are relative and lack a universal interpretation threshold for outliers.
Effective in datasets with varying densities — unlike global methods, it can recognize outliers near dense clusters.	Sensitivity to parameter selection (e.g., number of neighbors) can affect performance and consistency.
Works well across domains, such as network intrusion detection or classification tasks.	Interpretation of LOF scores can vary significantly between datasets or even within a dataset.
Applicable in any domain where a dissimilarity measure is defined, not limited to vector spaces.	Not inherently scalable to very large datasets without approximation or optimization.
Easily generalizable and adaptable for use in spatial data, temporal data, and network structures.	Difficult to determine a consistent threshold for what constitutes an outlier.

Table 3.4: Pros and Cons of the Local Outlier Factor (LOF) Method

3.3.5 Multimodal Outlier Detection

There is a big controversy in the field about the best method to detect the outliers, since they can vary depending on the methodology used.

After reading some aggregative methods such as Abro, Taşci, and Uğur [1], I decided to create a multi-modal method myself, that combines the results of the different methods to get a more robust result.

This method consists in only deleting the common outliers between the different methods, assuring that the data excluded is only formed by *true outliers*.

3.4 Data Transformation

To ensure a robust and well-performing model, it is often necessary to preprocess the data by transforming it into a format more suitable for learning algorithms—while preserving the ability to revert it back to its original scale if needed. In many practical cases, rather than analyzing the precise distribution of each variable, we simply standardize the data by centering each feature (subtracting the mean) and scaling it (dividing by the standard deviation), provided the feature is not constant.

This step is particularly important because many components of machine learning models—such as the RBF kernel in Support Vector Machines or the regularization terms (L1 and L2) in linear models—implicitly assume that features are centered around zero and have comparable variances. Without this adjustment, features with significantly larger variances could dominate the optimization process, leading the model to underweight or ignore more informative but lower-variance features.

3.4.1 Standard Scaler

When training the neural networks, it is really important to standardize the data. We can easily use the StandardScaler from the scikit-learn library that uses the transformation of a feature x_i is calculated as:

$$z_i = \frac{x_i - \mu}{\sigma} \tag{3.2}$$

where μ is the mean of the feature and σ is its standard deviation. This transformation results in a distribution with a mean of 0 and a standard deviation of 1.

3.4.2 Gaussian Transformation

In numerous modeling applications, having normally distributed features is advantageous. Power transformations represent a set of parametric, monotonic functions that are an extension of the Box-Cox transformation. They are designed to convert data from various distributions into approximately Gaussian distributions, thereby reducing variance fluctuations and decreasing distribution asymmetry. These family of transformations are also reversible, so we can easily transform the data back to its original space when needed.

We finally decided to use the Yeo-Johnson transformation because it allows for negative values and it is reversible.

$$x_i^{(\lambda)} = \begin{cases} \frac{[(x_i+1)^\lambda - 1]}{\lambda}, & \text{if } x_i \geq 0, \lambda \neq 0 \\ \ln(x_i + 1), & \text{if } x_i \geq 0, \lambda = 0 \\ -\frac{[(-x_i+1)^{2-\lambda} - 1]}{2-\lambda}, & \text{if } x_i < 0, \lambda \neq 2 \\ -\ln(-x_i + 1), & \text{if } x_i < 0, \lambda = 2 \end{cases} \tag{3.3}$$

Where λ is a power parameter that helps minimize the skewness of the data. And since we have already deleted extreme outliers, the final distribution shouldn't be too distorted.

Here are some exammples of the transformations for different distributions:

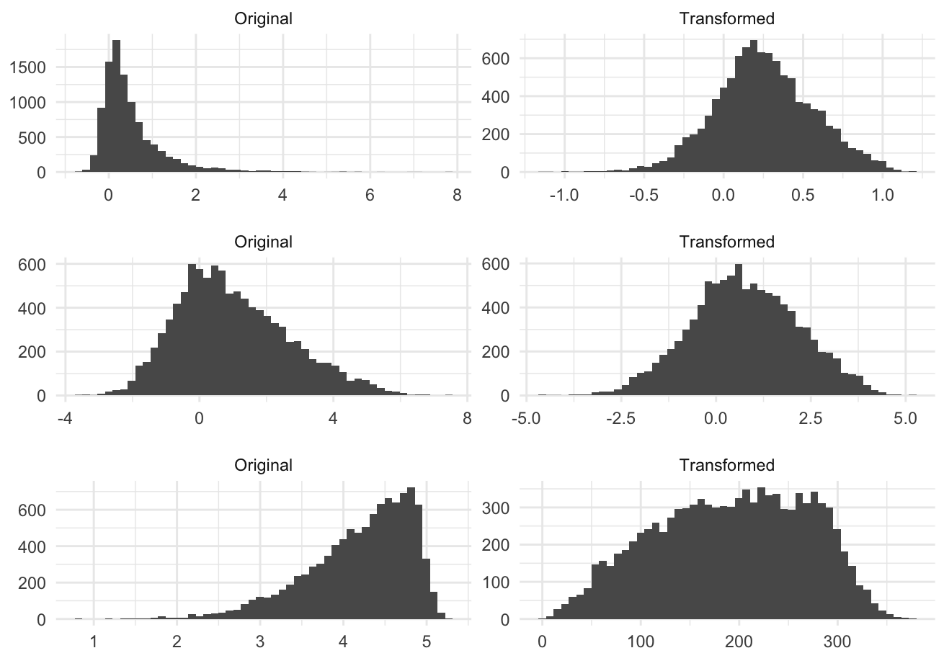

Figure 3.2: Yeo-Johnson Transformations examples

3.5 Predictive Models

The main objective of this thesis is to check if the scores have any relation with the future profits of the companies. For doing this in the most generalized way, instead of creating a portfolio following some subjective criteria on the scores, markets and other variables, we will use predictive modeling to check for the actual relationships between the response variables and the predictive ones.

As a first approach, we will be creating a set of models that separate the scores and the different returns.

Response Variables

We will use different profits that from know one will be representad as P_i for each time horizon: $i \in \{1 \text{ month}, 3 \text{ months}, 6 \text{ months}, 1 \text{ year}, 2 \text{ years}, 5 \text{ years}\}$

Explanatory Variables

For being able to train the models, we have different variable categories:

- **Main variables**, these are numerical variables from **Tweenvest's Scores**, that are in the range of 0 to 100: S_j where $j \in \{$ Quality; Value; Dividend; Growth$\}$
- **Dummy variables**, these are boolean variables that allow us to separate the companies by:
  - **Industry**: I_k where $k \in \{$Financials; Healthcare; etc.$\}$
  - **Region**: R_l where $l \in \{$Europe; North America; etc.$\}$
- **Market Variables**, these are numeric variables that represent the companies *"size"* using the: **Market Cap** and **Volume**.

3.5.1 Regression Models

To create the econometrical models, we want to be able to understand the actual relationship between the response variables and the explanatory ones. For this we started with the creation of multiple regression models.

Linear Regression

As for the first models, we created a series of linear regression models for each factor score that includes the interactions between the dummy variables and the score used in each model.

$$\hat{P}_i = \beta_0 + \beta_1 \cdot S_j + \sum_{n=2}^{N} \beta_n \cdot D_n + S_j \cdot \sum_{m=N+1}^{M} \beta_m \cdot D_m \tag{3.4}$$

where:

- P_i is the estimated profit for each time horizon i
- S_j is the score for each factor j
- $D_{n,m}$ are the dummy variables for the industry and region, where $D_{n,m} := I_k \cup R_l$
- β are the coefficients that weight each variable.

For fitting the models, we will also apply a backward selection to remove the variables that are not statistically significant, using the **p-value** as the criterion.

Generalized Additive Models

Because the relationships between the variables are not linear, we also used the Generalized Additive Models (GAM) to check if they are able to capture the non-linear relationships between the variables.

According to Servén and Brummitt [5], GAMs are smooth semi-parametric models that can capture non-linear relationships between variables. They take the form:

$$g(\mathbb{E}[y|X]) = \beta_0 + f_1(X_1) + f_2(X_2, X_3) + \cdots + f_M(X_N) \tag{3.5}$$

where:

- $X^T = [X_1, X_2, \ldots, X_N]$ are the independent variables
- y is the dependent variable.
- $g()$ is the link function that relates the predictor variables to the expected value of the dependent variable.
- $f_i()$ are feature functions built using penalized B-splines, which automatically model non-linear relationships without requiring manual transformation of variables.

In our case, we will use pyGAMs LinearGAM since it gives a Normal error distribution, and an identity link.

3.5.2 Time Series

In the stock market, the prices of the stocks are not independent of each other, they are correlated in the time series and they also have memory on past behavior of the company. This is what is known as **Momentum**, so companies that have a good past performance are more likely to have a good future performance, and vice versa.

For this reason, we contemplated two possibilities:

ARIMA Models

To take account the profit tendency, we implemented a ARIMA model to improve the regression models performance by using the residues of the already fitted model:

$$P_i = \hat{P}_i + \varepsilon_i \quad \longleftrightarrow \quad \varepsilon_i \sim ARIMA_i(p, d, q) \tag{3.6}$$

The ARIMA model is defined as:

$$\phi_i(B_i)(1 - B_i)^{d_i} \varepsilon_i^t = \theta_i(B_i)\eta_i^t \tag{3.7}$$

where for each i profit:

- ε_i^t is the residual at time t.
- η_i^t is a white noise error term (innovation).
- B_i is the backshift operator, such that $B\varepsilon_i^t = \varepsilon_i^{t-1}$.
- $\phi_i(B) = 1 - \phi_{i1}B - \phi_{i2}B^2 - \cdots - \phi_{ip}B^p$ is the autoregressive (AR) polynomial.
- $\theta_i(B) = 1 + \theta_{i1}B + \theta_{i2}B^2 + \cdots + \theta_{iq}B^q$ is the moving average (MA) polynomial.
- d_i is the order of differencing.

Windowed Models

And for capturing possible memory of the companies behavior in different aspects, we also will implement a windowed model that will use the last n scores statistical properties to predict future profits.

So for each score j, we will calculate the average and the standard deviation of the scores over the last w periods. Let S_j^t be the score at time t, then for a window of size w:

$$\bar{S}_j^w = \frac{1}{w} \sum_{i=t-w+1}^{t} S_j^i \tag{3.8}$$

where:

- $\bar{S}_j^w$ is the average score over window w for score j
- w can be 3 or 6 months, 1 or 2 years.
- t is the current time point
- j represents the different scores (Quality, Growth, Value, Dividends)

We will also calculate the standard deviation of the scores over the same window:

$$\sigma_j^w = \sqrt{\frac{1}{w} \sum_{i=t-w+1}^{t} (S_j^i - \bar{S}_j^w)^2} \tag{3.9}$$

These statistical measures will be used as additional features in our predictive models to capture the temporal behavior of each score.

3.5.3 Neural Networks

Finally, for trying to capture the non-linear relationships between all the available variables, including the windowed statistical properties, we will use the Neural Networks.

Chapter 4

Development and Results

Chapter 5

Discussion

Chapter 6

Conclusion

Bibliography

[1] A. A. Abro, E. Taşci, and A. Uğur. "A Stacking-based Ensemble Learning Method for Outlier Detection". In: *Balkan Journal of Electrical & Computer Engineering* 8.2 (Apr. 2020). URL: https://dergipark.org.tr/en/download/article-file/1106223.

[2] Markus M. Breunig et al. "LOF: identifying density-based local outliers". In: *ACM sigmod record*. 2000. URL: https://dl.acm.org/doi/pdf/10.1145/335191.335388.

[3] scikit-learn developers. *OneClassSVM*. scikit-learn Documentation. URL: https://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html. 2025.

[4] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation-based anomaly detection". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 6.1 (2012), p. 3. URL: https://www.lamda.nju.edu.cn/publication/tkdd11.pdf.

[5] Daniel Servén and Charlie Brummitt. *Generalized Additive Models*. pyGAM Documentation. URL: https://pygam.readthedocs.io/en/latest/notebooks/tour_of_pygam.html. 2018.

[6] Noumir Zineb, Paul Honeine, and Cedue Richard. "On simple one-class classification methods". In: *IEEE International Symposium on Information Theory Proceedings* (2012). URL: https://www.researchgate.net/publication/261248728_On_simple_one-class_classification_methods.

Part II

Appendices

Appendix G

Additional Listings

Example appendix content.