UCLouvain

LOUVAIN
School of Management

Louvain School of Management

Comparing Earnings Prediction Models to Analysts' Consensus Forecasts: A Machine Learning Approach

Author: Gauthier Thieren
Supervisor: Corentin Vande Kerckhove
Academic year 2022.-2023.
Dissertation for the master of
Master [120]: Business Engineering
Daytime schedule

Abstract

Value investing is an investment approach that involves selecting undervalued assets based on fundamental analysis, with the expectation that their true intrinsic value will be recognized by the market over time, leading to abnormal long-term returns. We apply machine learning techniques, specifically histogram-based gradient boosting regression trees, to predict one-year net income growth using a trailing three-year window of historical company fundamentals. The premise is that an accurate earning prediction leads to the selection of better-performing stocks, enabling the development of a long-holding, systematic value investing strategy. We assess the predictive performance of the approach with that of analysts' consensus forecasts alone and a similar approach combining analysts' consensus forecasts and company fundamentals as model input data. With respective R^2 values of 0.045 and 0.232, results do not show that making use of fundamental data alone can offer comparable results to analysts' forecasts. Moreover, while the combined approach shows marginal improvements over analysts' consensus forecasts, the potential of combining consensus forecasts with historical fundamentals remains underutilized with the chosen configuration. These findings suggest future research should make use of more sophisticated models incorporating historical fundamentals alongside analysts' consensus.

I would like to express my sincere gratitude to Professor Corentin Vande Kerckhove for his patience, advices and for providing me with valuable opportunities throughout the course of this thesis.

I would also like to thank my close friends for challenging and helping to improve the ideas developed in this thesis.

Table of Content

List of Figures

List of Tables

1. Introduction

1.1. Theoretical background

1.1.1. Equity Markets and the Efficient Market Hypothesis

Equity Markets (or Stock Markets) refer to exchanges in which shares of publicly held companies are bought and sold. These shares constitute a fractional ownership of individual companies. Prices associated with these shares are determined by the equilibrium between supply and demand. In simple terms, when the demand for a specific company stock surpasses the available supply, the usual outcome is an increase in the per-share price. Conversely, if the supply exceeds the demand, the stock price typically experiences a decline (Egan, 2023).

While engaging in equity markets, investors will typically seek to select and combine a set of stocks into a portfolio, with the goal to maximize returns—i.e., outperform a pre-defined benchmark—and minimize risk. This requires estimating the value of the considered stocks. To this end, two approaches are usually combined: *technical* analysis, which involves analyzing historical stock prices to forecast future prices; and *fundamental* analysis, which involves evaluating a company's financial information to identify stocks with potential. By comparing the estimated value to the current stock price, investors can then determine whether a particular company is undervalued.

In his review paper *Efficient Capital Markets*, Eugene F. Fama (1970) argued that markets featured a high level of efficiency in incorporating information regarding both individual stocks and the overall stock market. The valuation of a company—and subsequently its per-share price—is such that it integrates all the information publicly available without delay. Consequently, both *technical* analysis and *fundamental* analysis would not enable an investor to outperform the returns achievable by holding a randomly selected portfolio of stocks with similar risk (Malkiel, 2003).

However, emphasis has since then been put on the psychological and behavioral factors influencing the determination of this demand-supply price equilibrium. The scientific community has proposed several ways in which future stock prices can be predicted, by analyzing past stock price patterns and/or company fundamental data (Cavalcante et al., 2016). The partial predictability of these prices means that an investor can outperform benchmark risk-adjusted returns.

1.1.2. Fundamental Analysis and Value Investing

Several factors should thus help predict this price equilibrium. However, the set of these drivers depends on the considered time scale (Baker, 2022). In the short run (i.e., intra-day transactions), the dynamics of order execution and the activities of high-frequency traders largely explain price movements. Medium run price changes (i.e., over several days) are mostly explained by the news cycle. By contrast, long-term price movements are exclusively explained by the financial performance of companies, i.e., their *fundamentals*—such as balance sheet, income statements or cash flow statement items.

Benjamin Graham (1965) famously suggested that while the stock market may function as a voting mechanism in the short run, it operates as a weighing mechanism in the long run. This means that, despite their short-term variability, share prices are expected to eventually converge towards the company's *intrinsic* value, which is determined by the cumulative discounted cash flows generated by the company. One investment strategy called *value investing* capitalizes on this concept by advocating for the identification of companies with prices inferior to the corresponding *intrinsic* value, and by suggesting that the best way to estimate this *intrinsic* value is through a *fundamental* analysis.

Most of the research has been focused on *technical* analysis, using stock prices or related indicators as inputs to stock price forecasting models (Bustos & Pomares-Quimbaya, 2020). Resulting predictions are being used on a short-term basis as indicators for when to buy or sell a stock. However, research has also indicated that trading strategies relying on *technical* analysis yield limited results (Park & Irwin, 2007).

Conversely, *fundamental* analysis aiming at forecasting future stock prices is less present in the literature. The reason is that it is hard to build models that explain stock prices fluctuations using fundamental information, given the high signal-to-noise ratio that characterizes the relationship between fundamentals and price. However, relationships among fundamental data offer a larger signal-to-noise ratio (Alberg & Lipton, 2018). This entails that a model that aims at predicting company fundamentals instead of prices should exhibit a greater prediction performance.

However, forecasting fundamentals instead of prices means no straightforward selection of better performing stocks can be achieved. To cope with this, using these forecasted fundamentals, a systematic *value investing* strategy can be developed, and estimations regarding the associated future return can be derived. An option would be to sort

companies based on a given *factor* (e.g., P/E, EBITDA/EV, or the equity book-to-market ratio) that puts into relation forecasted future financial performance (‘*is it a good company?*’) and current valuation (‘*is it a good price?*’). Another option would be to rank companies based on some predicted fundamental surprise (typically revenue or earnings)—i.e., the difference between the fundamental value expected by analysts and the predicted fundamental value. This enables the identification of companies for which the analyst community is likely to have over- or under-estimated fundamentals, and to benefit from the subsequent arbitrage opportunity caused by the mispriced stock (see 1.1.3 for greater explanations). Regardless of the chosen option, the strategy would then essentially consist in constructing portfolios having a long position in stocks which rank highest, and a short position in stocks which rank lowest (see Figure a).

Figure a: Overview of the Four Considered Modelling Pipelines.

1	**Technical data** (price, …)	→	**Price prediction**		→		**Ranking**	→	**Portfolio Simulation** (return backtest)
2	**Technical data** (price, …)	→	**Earnings prediction**	→	**Value Investing Strategy**	→	**Ranking**	→	**Portfolio Simulation** (return backtest)
3	**Fundamental data** (earnings, …)	→	**Price prediction**		→		**Ranking**	→	**Portfolio Simulation** (return backtest)
4	**Fundamental data** (earnings, …)	→	**Earnings prediction**	→	**Value Investing Strategy**	→	**Ranking**	→	**Portfolio Simulation** (return backtest)

Figure a illustrates the four considered modelling pipelines. Pipeline 1 has been dismissed given its already high prevalence in literature and its short-term orientation. Pipelines 2 and 3 have not been selected given the high signal-to-noise ratio that characterizes the relationship between fundamentals and prices. Pipeline 4 has been selected, but we decided to focus on only the first two stages, as later stages only translate *statistical* performances into *financial* performances (read 1.2.1 for greater explanations).

1.1.3. Analysts’ Consensus Estimates

Sell-side analysts play a crucial role as expert financial intermediaries in the equity market (Wang et al., 2022). Their responsibilities include gathering and analyzing both public and private information to assess the present performance of companies and generate earnings forecasts based on this evaluation. These earnings forecasts help evaluate a company's stock *intrinsic* value and offer investment guidance to investors.

They also serve as a significant benchmark when studying finance and accounting issues. Indeed, as Abarbanell et al. (1995) put it, *“empirical forecast measures are assumed to proxy for investor beliefs, which are unobservable”*. This means that analysts' consensus

estimates [1] are considered to represent the market's expectations for company earnings. According to the rational expectations theory, current stock prices should thus reflect consensus estimates at various time horizons for the respective companies. This entails that if actual earnings are in line with consensus forecasts for a given period, there should be no substantial movement in the stock price when results are published. However, actual earnings significantly above (below) corresponding consensus forecasts—meaning a positive (negative) earning surprise is observed—should lead to a stock price increase (decrease) when results are published. This occurs because the market had so far not fully incorporated this information.

Yet, research has shown that these analysts' estimates *"differ significantly from actual reported earnings"* (see Dreman & Berry, 1995). As a result, surprise values can be substantial, and—if better earnings forecasts than analysts' are achieved—the corresponding arbitrage opportunities as well.

1.2. Research Question

1.2.1. Development

Previous studies examining analysts' earnings forecasts suggest that analysts fail to consistently and accurately integrate fundamental information in their estimations (Abarbanell & Bushee, 1997; Wieland, 2011). Consequently, there is potential for improving the development of earnings prediction models by incorporating company fundamentals as input.

Research has also shown that active trading strategies with a long holding horizon outperform strategies with a short holding horizon. For example, Lan et al. (2023) provide empirical evidence that *"long-horizon funds exhibit positive future long-term alphas by holding stocks with superior long-term fundamentals".* These findings suggest that adopting a long holding, systematic investing strategy, based on a fundamental analysis, holds promise.

[1] To mitigate the influence of individual idiosyncratic errors, individual analysts' forecasts are often combined into a consolidated measure known as a consensus estimate. For this study, the consensus estimate is defined as the average forecast derived from the group of analysts who have provided estimates.

In this study, we investigate the extent to which performances of prediction models for 1-year earnings growth can compare to corresponding analysts' consensus forecasts. To this end, we compare the performances of machine learning models in two scenarios against a baseline, and assess how well these three outputs compare to actual 1-year earnings growth:

- **Baseline:** Analysts' consensus forecasts of earnings as output.
- **Scenario 1:** Earnings prediction output from models where only fundamental data from previous periods is taken as input.
- **Scenario 2:** Earnings prediction output from models where combined analysts' consensus forecasts and fundamental data from previous periods are taken as inputs.

The goal of the first scenario is to assess how well machine learning techniques can provide a comparable result to analysts. The second scenario seeks to examine the extent to which *forward-looking* yet *biased* human-based input (analysts' consensus forecasts) and *unbiased* but *backward-looking* quantitative input (fundamental information) can complement each another.

As described in 1.1.2, an improved prediction of earnings thanks to these models enables the execution of a systematic *value investing* strategy that necessarily outperforms an identical strategy relying solely on analysts' consensus forecasts [2]. Therefore, this study does not involve the formulation of a specific *value investing* strategy or the implementation of a *return backtest*—i.e., a trading simulation. Part of the reason is that the development of an optimal investment heuristic and of an accurate portfolio simulator requires additional considerations—for the former, selecting investing decision rules *via* an oracle; for the latter, modelling transaction costs, slippage, etc. This would only translate *statistical* performance (e.g., MSE) into *financial* performance (e.g., CAR, Sharpe ratio), which is unnecessary for addressing the research question at hand. In practical terms, this means that this paper will focus on the first two stages shown in Figure a.

[2] This holds as long as the chosen investing strategy is sensible and identical for both predicted future earnings and analysts' consensus forecasts.

1.2.2. Related Work

As mentioned in 1.1.2, most of the machine learning research in stock market applications has been applied to stock prices forecasting (be it through *technical* or *fundamental* analyses), as opposed to future fundamentals predictions. Still, one notable study by Alberg & Lipton (2018) show that *value investing* strategies can be improved by selecting stocks using factors calculated on future fundamentals (via oracle), instead of using factors calculated on already published fundamentals. From there, they posit that an accurate prediction of these future fundamentals should lead to an improved portfolio performance. After training deep neural networks to forecast future fundamentals based on a 5-years window of previous fundamentals, they used predicted fundamentals as a proxy for *actual* future fundamentals and demonstrate a significant improvement in portfolio returns. However, they do not compare how well their prediction compare to analysts' consensus forecasts, and, in an extended version of their paper (Chauhan et al., 2020), mention that further investigation should be carried out on whether *"forecasts can be improved by the consensus forecast"*.

When it comes to comparing future earnings predictions and analysts' consensus forecasts, a paper by Wahlen & Wieland (2011) investigates *"whether share prices and analysts' consensus recommendations fully reflect available financial statement information"*. They test if financial statement information can be used to predict future earnings changes. Subsequently, they examine whether these predictions can be used to implement a trading strategy that outperforms analysts' consensus recommendations (i.e., *strong buy, buy, hold, underperform*, or *sell*). Another study by CHEN et al. (2022) aims at predicting the direction of one-year-ahead earnings changes using machine learning. Results show greater predictive power in machine learning models compared to analyst forecasts, and, as a result, greater annual returns. However, both studies do not examine the effect on prediction performances of the combined use of financial statements information and consensus information.

1.3. Methodology

Before dealing with the specificities of data handling and modelling (see parts 2 and 0), we must first cover the salient points of the methodology and present the reasons that underpin these choices.

1.3.1. Time Series Frequency

Exported company fundamental data takes the form of *panel* data (also known as *longitudinal* or *cross-sectional time series* data). This is a type of data structure that combines multidimensional information from multiple entities—in our case publicly traded companies—observed repeatedly across different time periods (see the structure of Compustat Fundamental data in Table 9). Panel data allows for analyzing both cross-sectional interactions and time-specific patterns. One of the main characteristics of panel (and time series) data is the time interval between measurements, i.e., how frequently company data is refreshed.

Technical analysis makes use of data that is available with a high temporal granularity (daily frequency, if not more), and enables the development of models with a high-frequency of prediction outputs. Accordingly, a high frequency of portfolio updates can be set.

Conversely, *fundamental analysis* makes use of reported information that arrives quarterly. The frequency of prediction outputs is constrained by this time interval of measurements, as more frequent predictions would essentially be identical if the input data did not update. Thus, predictions for a given company are assumed to be made every quarter. This means that, for a given company, one earning forecast is made every quarter for a future quarter—i.e., there is a one-to-one correspondence (or bijection) between quarters during which the prediction is carried out and quarters for which earnings are forecasted. [3]

[3] Note that this does not imply that the portfolio derived from the chosen trading strategy is only rebalanced on a quarterly basis. One reason is the intermittent release of reported information from different companies, which spans across a quarter. Another reason is that forecasted future financial performance needs to be regularly compared with the company's valuation—which experiences daily fluctuations—to make an investing decision.

1.3.2. Portfolio Holding Horizon and Model Prediction Horizon

Now that this one-to-one correspondence has been established, another crucial decision must be made regarding the interval between these two dates—i.e., the prediction horizon. This choice should be guided by the intended purpose of the predictions in practical application. In the context of this study, where the goal is to enhance long-term value investing strategies (see 1.2.1), it is essential to align the chosen prediction horizon with the holding horizon.

In machine learning applications, training models on a diverse set of data is key to reduce overfitting and improve generalization performances. This applies to time series characteristics—patterns, trends, relationships; in our case, the underlying behavior of the stock market—, given that these evolve over time. This means that greater chronological coverage is beneficial to training.

However, this span of available quality company fundamental data is limited. And longer prediction horizons inherently reduce the number of independent time periods available for analysis. Therefore, considering this limited span, an excessively long prediction time-horizon further diminishes the diversity of time-related information.

This means that it becomes necessary to select an intermediate prediction horizon—namely a one-year timeframe—which strikes a balance between aligning it with the desired portfolio holding horizon and maximizing data diversity.

In practice, it is important to note that, (i) due to the selection of the last available consensus forecasts for a given analyst forecast horizon and a given quarter end date, and (ii) due to the delay between the quarter end data and the results publication date, the model prediction horizon usually ranges between 8 and 9 months. More detailed information can be found in appendix D.4, where Figure p provides a visual representation of the issue at hand.

1.3.3. In-Sample and Out-Of-Sample Data Sets

As for any predictive models, splitting the available set of data into train & test sets is necessary to assess performances and generalization capabilities. The training set is used to train or "teach" the model on historical data, enabling it to learn the underlying patterns and relationships. Once the model is trained, it is evaluated on the testing set, which consists of unseen data that the model has not been exposed to during training.

Usually, a specific splitting ratio (e.g., 70:30) is selected, and available observations are randomly allocated to the train & test sets. However, unlike cross-sectional data, time series data has a temporal dependency—i.e., there is an inherent relationship between data points based on the order and timing of their occurrence. Thus, random shuffling cannot be applied to create independent train and test sets. To address this, the train-test split is typically performed using a chronological approach (see Figure b). The dataset is divided into two parts based on a specific point in time, where all data before that point constitutes the training set (*in-sample* period), and all data after that point forms the testing set (*out-of-sample* period). This ensures that the model is trained on past observations and evaluated on future or unseen data, replicating a real-world scenario where predictions are made based on historical information. This approach prevents the model from accessing future information during training, avoiding look-ahead bias and providing a more accurate assessment of its predictive performance on unseen future data.

Figure b: Time-Series Train and Test Split.

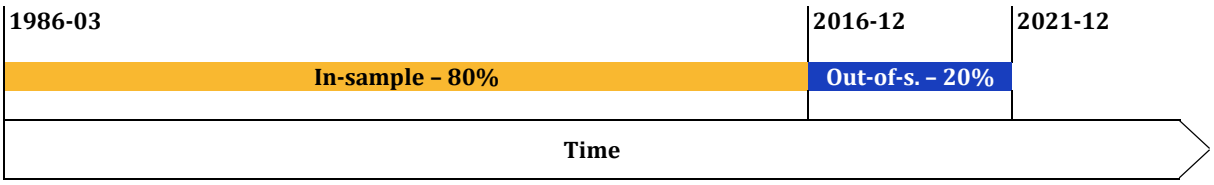


Figure b illustrates the chronological split between train and test sets with time-series data. The in-sample period (train data) is marked in yellow. In blue, we show the out-of-sample period, which is used to calculate performance measures on a real-world, historical, scenario.

In our case, the *in-sample* period covers 80% of the available dataset and spans from 1986-Q1 to 2016-Q4. Accordingly, the *out-of-sample* period spans from 2017-Q1 to 2021-Q4.

1.3.4. Walk-Forward Validation and Hyperparameters Tuning

When training predictive models, a key determinant of the resulting performances is the choice made for the value of hyperparameters. These parameters are set before the model training process and cannot be learned from the data. They control various aspects of the model's behavior, such as the learning rate, regularization strength, etc. Tuning these hyperparameters involves systematically exploring different combinations of values and evaluating the model's performance using a predefined metric (e.g., MSE). This is crucial

for optimizing model performance and ensuring that the model generalizes well to new, unseen data.

One widely used technique to select the right values for hyperparameters is cross-validation. It involves further splitting the training data into multiple subsets, training the model on a portion of the data, and evaluating its performance on the remaining subset. This process is repeated iteratively, with each subset serving as the validation set. By systematically testing different hyperparameter combinations and assessing the model's performance using cross-validation, it becomes possible to identify the optimal hyperparameters that result in the best average performance across multiple folds.

However, as for the train-test split, the temporal nature of the data implies random shuffling cannot be applied to create independent subsets. Instead, a walk-forward validation method with an expanding window is implemented (see Figure c), preventing any look-ahead bias.

Figure c: Expanding Window Walk-Forward Validation for Time Series Data.

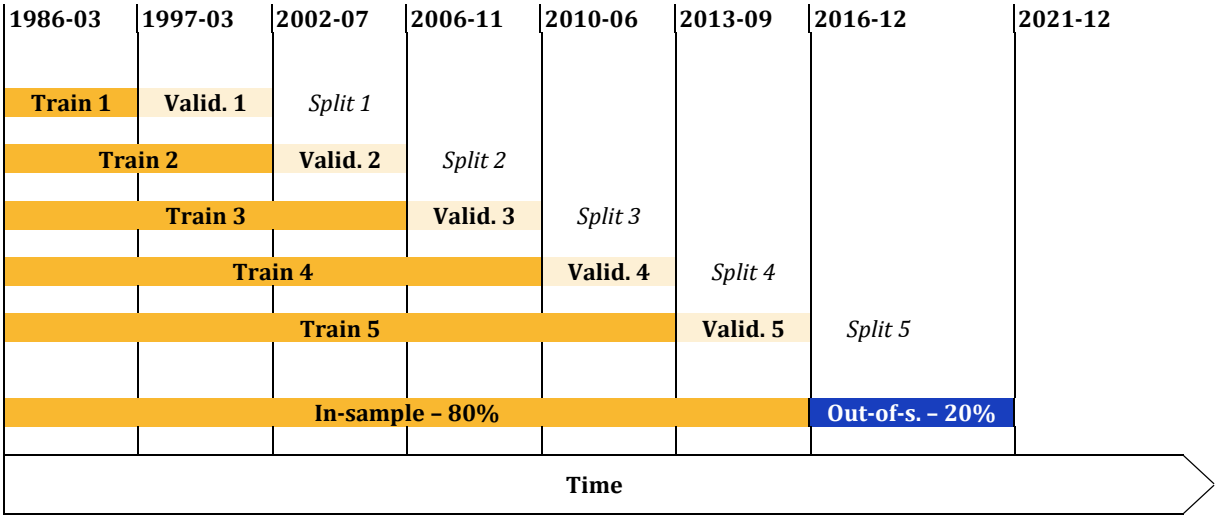


Figure c illustrates how the in-sample period gets further divided into folds when performing a walk forward validation with time-series data. This phase of validation is used to tune models hyperparameters.

2. Data Preparation

2.1. Data Fetching

2.1.1. Universe

In this study, we focus on all stocks—active and inactive—publicly traded on the NYSE, NASDAQ or AMEX exchanges, that featured a minimum listing period of 16 quarters. We exclude any companies with an inflation-adjusted market capitalization below $100 million, that are not registered in the US, and that are active in the financial sector. Finally, we ensure selected companies are both availability in the Compustat and IBES databases. The rationale behind these selection criteria is presented in appendix D.1.

At this stage, the list contains 8,379 companies, representing 620,408 company-quarter combinations, spanning between 1973 and 2023. Further processing narrows this list to a final set of 6,964 companies, representing 215,296 company-quarter combinations, spanning between 1986 and 2021.

2.1.2. Source

All the data has been retrieved from two different databases available online on the Wharton Research Data Services (WRDS) platform. The first is the *Compustat* database, from which company fundamentals and market information were retrieved. The second is the *IBES* database, from which company consensuses data was obtained.

More details regarding databases sources and data retrieval procedure can be found in appendix A.

2.2. Feature engineering

2.2.1. Target Variable

Several potential target variables were initially considered, in part because forecasting several earning items (Sales, EBIT, etc.) concurrently ("multi-task learning") has the advantage of reducing overfitting. In the end, we settled on a single earning measure, namely *net income*.

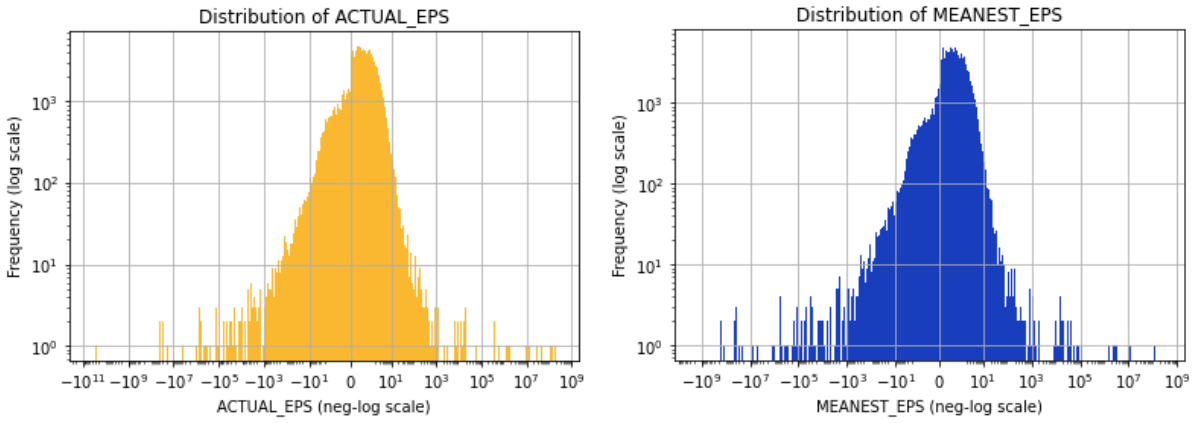
The reason is that we want to maximize the use of the available coverage of historical data. Compustat data is available from the 60's onwards, while IBES consensus forecasts (Sales, EBIT, etc.) are only made available in the early 2000's—except for EPS forecasts. IBES started publishing consensuses for EPS (*Earning Per Share*, i.e., the net income divided by

the weighted average number of common shares outstanding, adjusted for splits) in 1976. Choosing *net income* maximizes data availability—though at the expense of predictability given that *net income* incorporates many elements, compared to other income statement metrics such as Revenue, or EBIT.

Compustat and IBES databases are not necessarily compatible (refer to the discussion about the limits of this study in 5.2.2 for more exhaustive explanations). Thankfully, IBES provides the actual value of EPS besides the analysts' consensus forecast. Thus, in order to maximize the relevance of the comparison between the analysts' consensus EPS forecast and the actual EPS, both data series were retrieved from IBES—instead of retrieving the former from IBES and the latter from Compustat.

With this approach, we obtain a *per-share* measurement. Two companies with identical net income may have a different number of shares and hence a different EPS figure. This leads to great disparity in EPS actuals and EPS forecasts (see Figure d).

Figure d: Distribution of EPS *Actuals* and *Forecasts*



A *non-per-share* measure (i.e., *net income*) is thus necessary for the model to make comparison across companies. IBES does not disclose the weighted average number of shares outstanding, meaning a *net income* figure cannot be reliably deducted from the corresponding EPS figure [4].

[4] A considered option was to use a fairly unprecise number of shares outstanding to compute the consensus net income—namely, the Compustat variable *'Common Shares Used to Calculate Earnings Per Share'* (CSHPRQ). A test for Microsoft (MSFT) over the period 2002-2023 yielded a difference of 1.27% compared to the IBES actual net income. This option was dismissed in favor of a computation of the relative difference.

However, for reasons developed in 2.2.3, we compute the relative difference for all variables (EPS included) compared to the corresponding value in previous period. This enables comparison between companies, given that EPS figures retrieved from IBES (*actuals* and consensus *forecasts*) are adjusted for splits.

2.2.2. Most Recent Quarter & Trailing Twelve Months items

Basic accounting often distinguishes between two types of financial quantities, namely *stock* and *flow*, which have different aspects in respect to *time* (Fisher, 1896). The former relates to a financial quantity measured at a *point* in time, while the latter is applicable to a financial quantity measured over a *stretch* of time. Although this distinction may appear trivial, it nevertheless holds significant implications for the data preparation process.

In the present case, company data from three types of financial statements is retrieved—namely: the *Balance Sheet* (BS), the *Income Statement* (IS), and the *Cash-Flow* statement (CF). Items in the balance sheet quantify *stocks*, whereas items in the income and cash flow statements measure *flows*.

Features derived from BS items are *Most Recent Quarter* (MRQ), meaning data for a given quarter corresponds to observed items at the end of the quarter (see equation (1)). This is akin to annual financial statements which provide this picture usually on Dec. 31st. In our case, the financial picture is representative of the company state at every quarter end date—implying the interim FQ4 balance sheet is the same as the fiscal year balance sheet.

$$X_{t,q}^{MRQ} = X_{t,q} \tag{1}$$

Conversely, IS & CF items for a given quarter are not equivalent to a yearly picture—i.e., FQ4 items are not identical to annual items. The reason is that these items only account for the financial flow of ¼ of the yearly figure. This is not a problem *per se*, but to smooth out seasonal fluctuations and to allow an easier comparison of records for the model across different quarters, we decided to use a cumulative *Trailing Twelve Months* (TTM) figure instead. This means that, for each company-quarter combination and for each IS & CF items, we sum-up the values of the past four consecutive quarters (see equation (2))[5].

[5] A more appropriate abbreviation could have been Trailing Four Quarters (TFQ), given that we work with quarterly time steps—i.e., predictions are made on a quarterly basis—but the current mnemonic was chosen given its greater prevalence in the literature.

$$X_{t,q}^{TTM} = X_{t,q} + X_{t,q-1} + X_{t,q-2} + X_{t,q-3} \quad (2)$$

2.2.3. Relative Change of Features

A common need in supervised learning algorithms applied on financial time series data is the requirement of stationary features (Granger & Newbold, 1974) . A time series is said to be "*stationary when its statistical properties are invariant by change of the origin of time*" (López De Prado, 2018, p. 75). The reason is that any model makes accurate inferences only if the data taken as input is not specific to the given company, as this could result in overfitting.

However, financial data is notoriously not stationary: financial data of a particular company at a specific time step t is correlated with its financial data from preceding periods, such as $t - 1$. For instance, when predicting sales for the upcoming quarter using a simple strategy that assumes the same value as the previous year, a high R2 score is achieved (Flovik, 2018). Remarkably, this remains true even if the sales time-series is modeled as a random walk process, which is entirely stochastic in nature.

Data transformation is thus needed, otherwise a ML algorithm would assign a wrong prediction to an unseen observation (Walasek & Gajda, 2021). A common method to remove non-stationarity consists in differencing (Tsay, 2005). For example, a first-order differentiation can be obtained by subtracting from each observation its predecessor (see equation (3)). In that case, the above-mentioned model would consist in predicting the actual change in sales from one year to another, which offers a significantly more rigorous test of the model's predictive powers.

$$\Delta X_{t,q} = X_{t,q} - X_{t-1,q} \quad (3)$$

Still, given the high discrepancy in the magnitude of financial items across the company universe, the significance of a similar actual change in an item might not be the same for two different companies. For instance, Apple's 12 months revenue as of the end of CQ2'19 increased by $3,996M, while Occidental Petroleum (OXY) increased its revenue by a comparable $4,065M figure over the same period. However, Apple's total 12 months revenue as of the end of CQ2'18 was $255,038M, while Occidental Petroleum's equivalent was $14,337M. In relative terms, the increase in revenues is thus much more significant for Occidental Petroleum than it is for Apple.

As opposed to this actual difference, the computation of a relative difference between two time-steps (see equation (4)) enables a more meaningful comparison of companies of various sizes. Moreover, given that the effect of company size is erased, new observations can more easily be mapped to a set of examples from the training set, facilitating inference.

$$\Delta_{\%}X_{t,q}=\frac{X_{t,q}-X_{t-1,q}}{X_{t-1,q}} \tag{4}$$

For the aforementioned example, the relative difference values for Apple and Occidental Petroleum are 1.57% and 28.35%, respectively. This transformed feature offers a much more meaningful and representative picture of this $4B increase in revenues.

However, equation (4) fails at explaining the relative change of a negative number. Some financial statement items can take negative values (e.g., net income). A positive actual difference $X_{t,q} - X_{t-1,q}$ leads to a negative relative difference $\Delta_{\%}X_{t,q}$ if the previous period considered item $X_{t-1,q}$ takes a negative value.

The selected approach has thus been to use the absolute value of $X_{t-1,q}$ as the denominator (see equation (5)), so that the relative change formula works correctly for all non-zero values of $X_{t-1,q}$.

$$\Delta_{\%}X_{t,q}=\frac{X_{t,q}-X_{t-1,q}}{|X_{t-1,q}|} \tag{5}$$

This arrangement still leaves the issue of a zero value of $X_{t-1,q}$, which leads to the relative change taking either a positive or a negative infinity value. Moreover, the presence of $X_{t-1,q}$ values close to zero is likely to generate outliers. Both issues are discussed in 2.3.

2.2.4. Lag Features

Predictive models cannot access future information during training, in order to avoid look-ahead bias. Thus, only data available at the point in time when the prediction is made should be used as training data. However, models cannot automatically access data from previous time steps.

To address this issue, lagged variables are created, which take the values of the corresponding variables shifted by a predefined number of periods. In our case, every explanatory variable is shifted by 1, 2 and 3 years, meaning a 3-year window of past fundamentals serves as input variables. Through this procedure, we essentially triple the number of explanatory variables.

2.3. Data Cleaning

2.3.1. Infinite Values

The computation of the relative change of features (as detailed in 2.2.3) leads to the appearance of infinite values, when a zero value of the denominator $X_{t-1,q}$ is observed. No specific guidance regarding this particular situation could be found, so we decided to set these values to *NA* (i.e., as missing), given that the number of concerned observations is relatively small. This implies that these missing values are processed in an identical way to any other missing values, as detailed in 2.3.3.

2.3.2. Outliers

A similar issue occurs when denominator values are close—but not equal—to zero. Consequently, resulting variables can take large values, leading to distorted distributions. Since most prediction models are based on minimizing errors, these outliers can have a disproportionate influence, pulling the model's predictions towards them and causing poor generalization to new, unseen data. Dealing with these outliers is thus essential to ensure more robust and accurate predictions.

This process should be carried out carefully, in order not to distort prediction performances. More specifically, the treatment of outliers must take into account (i) the nature of variables (*target* or *explanatory*) as well as (ii) the period in which observations belong (*in-sample* or *out-of-sample* period) (see Figure e). Special care has to be put on observations for target variables belonging to the test set (*out-of-sample* period), as this sample constitutes the basis upon which performance metrics are computed.

In our case, we decided to apply a *winsorization* to explanatory features—regardless of the set to which observations belong. With this approach, outliers in the data are replaced with the nearest non-outlying values, from a specified threshold. In our case, the bottom 5% of the smallest values and the top 5% of the largest values are replaced with the values at the 5th and 95th percentiles, respectively. This imputation is done on a quarterly basis to avoid introducing look-ahead biases, meaning quarterly cross-sections of observations are considered independently (around 300 observations per quarter) when computing percentiles.

Figure e: Overview of Outliers Handling.

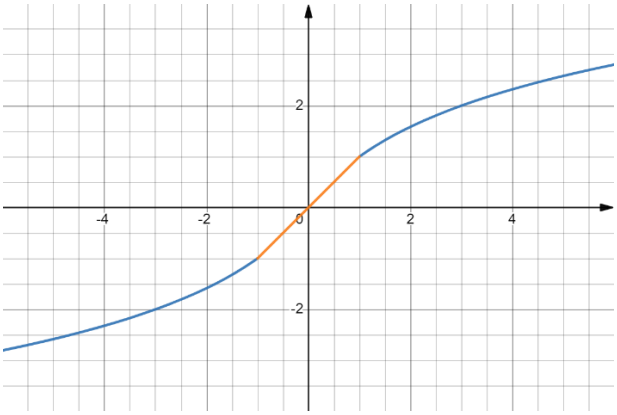
	Explanatory Features	Target Feature
Train Set – In-sample period	Cross-section winsorization	Non-linear bijective (invertible) transformation
Test Set – Out-of-sample period	Cross-section winsorization	Non-linear bijective (invertible) transformation

Although it is generally advised to not transform target features from the test set—as mentioned previously, this can lead to overoptimistic regression results—, we decided to apply a non-linear positive monotonic transformation to all target data. The reason is that the presence of very large outliers renders modelling especially hard, and the development of more intricate solutions is outside of the scope of this thesis. The alternative would have been to drop these observations altogether, which would have created even greater biases.

The chosen transformation is a variation of a neglog transformation defined by Whittaker et al. (2005), where the basic idea is to create a two-tail logarithm function, defined on the set of real numbers $\mathbb{R}$ instead of the set of positive real numbers $\mathbb{R}_{>0}$ only (see equation (6)). We decide to use a base 2 logarithm as $\log_2(|x|+1) = 1$ when $x = 1$, so we ensure that the function is continuous on the entire set of real numbers $\mathbb{R}$—as shown in Figure f.

$$\mathrm{nl}(x) = \begin{cases} -\log_2(|x|+1), & x < -1, \\ x, & -1 \leq x \leq 1, \\ \log_2(|x|+1), & x > 1. \end{cases} \tag{6}$$

Figure f: Visualization of the Neglog Transformation Applied to the Target Feature.



This choice presents several advantages compared to other transformations, namely:

- Data leakage issues are avoided, as no learnable parameters is being used—contrary to a winsorization where quantiles are derived from the data.
- Most values are left unchanged, as most values lie between the [-1,1] range (see Figure g)—i.e., EPS values rarely increase or decrease by more than 100% on a yearly basis.
- As the defined function is invertible, an inverse transformation can be carried out to get the non-log estimate.
- Given that the function is strictly positive monotonic, the relative order of values is preserved. This is important as the order of numbers encodes the inherent relationships in the data. Preserving this relative order enables the model to make sense of the underlying patterns in the data.

Figure g: Effect on the Target Variable Distribution of the Neglog Transformation.

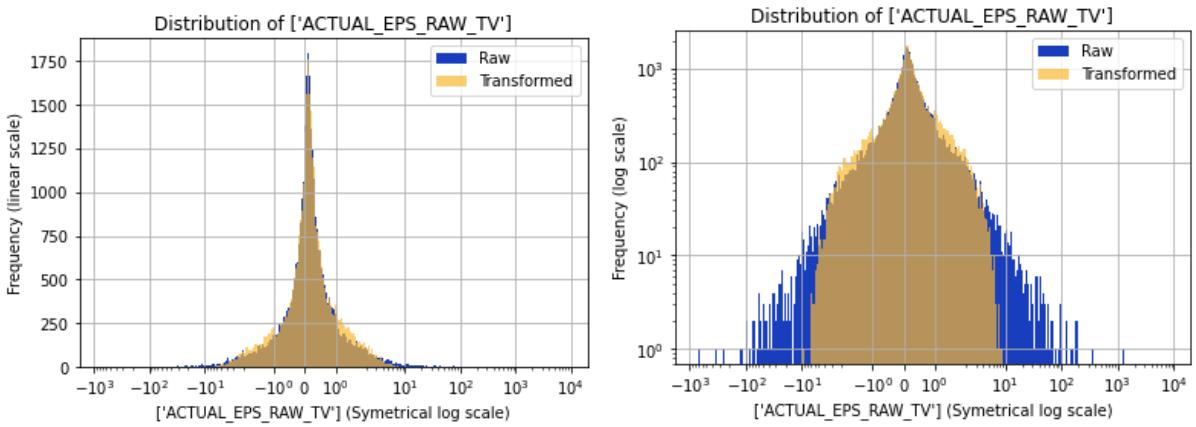


Figure g shows the effect that the transformation detailed in equation (6) has had on the target variable distribution. Untransformed data is shown in blue, transformed data in yellow. The left panel shows the frequency on a linear scale (accurate density function), while the right panel shows the frequency on a logarithmic scale, to better highlight the presence of outliers. Note that the x-axis is for both panels set using a *symmetrical log* scale, which is a base 10 log scale that allows for allows positive and negative values. The left panel demonstrates that the bulk of the target feature data (i.e., between -1 and +1, or -100% and +100%) has been left unaffected. The right panel demonstrates that this transformation caps extremum values in the [-10, 10] range, or between -1,000% and +1,000%.

2.3.3. Missing Values

Before treating missing values inherent to the data source, the first step consisted in dealing with missing values induced by structural modification of the data. It consisted of removing records for which lagged data was non-obtainable (i.e., the first three years of

results published for a given company) or misaligned (missing quarter). This respectively affected 134,064 and 3,349 records. The effect of this process can be seen in Figure h.

Figure h: Visualization of the Effect on the Chronological Distribution of Data of the Removal of Rows with Misaligned or Missing Lagged Dates.

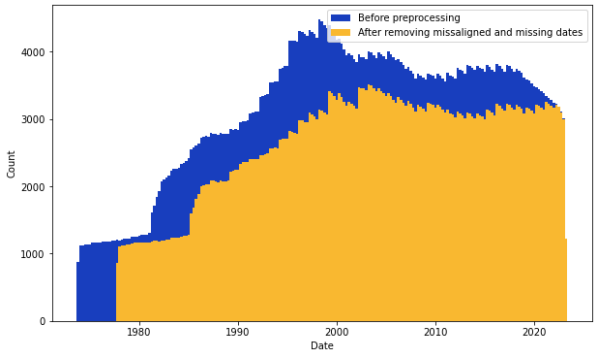


Figure h shows that removing rows for which lagged variables are not available shifts the temporal distribution of available data points rightwards—as the first three years of reported information do not have a sufficient backlog of financial information to create lag variables.

We are now left with missing values that are not caused by a structural decision and for which no straightforward treatment can be determined.

We first decided to drop records with missing EPS forecasts (*explanatory* variable) and actuals (*target* variable) altogether (see). This explains the greatest part of the reduction in the number of records carried out in the processing phase (see Figure j). As the goal is to compare the usefulness of consensuses forecasts, these two variables cannot be missing as this would create a bias in the model. This has a detrimental effect, as all records with quarter results prior to 1986 are removed (no IBES quarterly EPS were available before that date). Still, fundamental data prior to this date is still kept as input variable in lagged variables.

Figure i: Effect on the Proportion of Missing Values Caused by the Removal of Rows with Missing EPS Forecasts and Actuals.

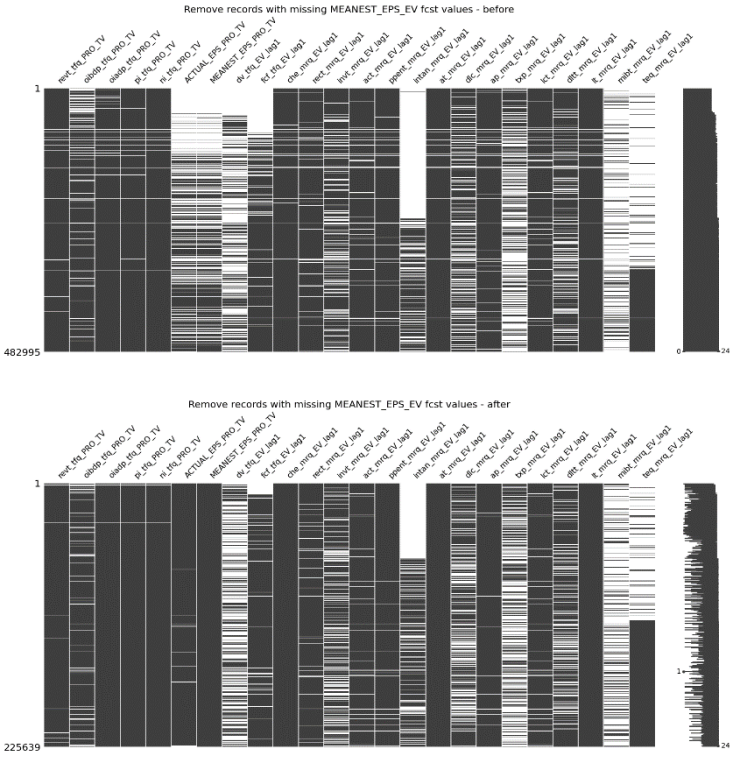


Figure i shows the evolution in the proportion of missing values for variables, following the removal of rows with missing EPS forecasts and actuals. As records are chronologically sorted, the y-axis in both panels represents the time axis; hence, for instance, the actual EPS variable (6th column) is characterized by a greater proportion of missing values as we go back in time. This explains why the rows with quarter results prior to 1986 are removed (see Figure j), as EPS forecasts values are missing prior to that date.

Figure j: Visualization of the Effect on the Chronological Distribution of Data of the Removal of Rows with Missing EPS Forecasts and Actuals.

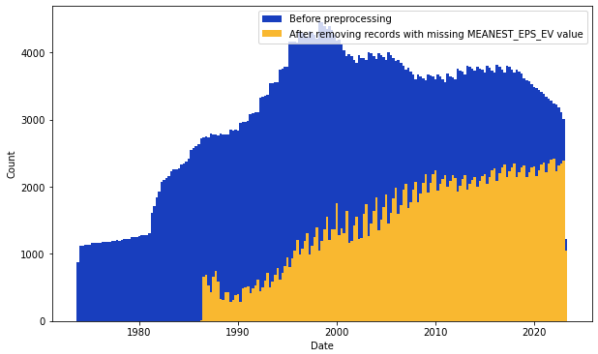


Figure j highlights that most of the diminution in the number of records during the pre-processing phase is due to the removal of rows with missing EPS forecasts and actuals—see Figure h for comparison.

Secondly, we decided to drop variables with too many missing values—namely *Intangible Assets* (intan), *Redeemable Noncontrolling Interest* (mibt), and *Total Stockholders' Equity* (teq). As it can be seen in Figure i, these variables feature a high percentage of missing

values (respectively; 45%, 74% & 46%), and, most importantly, this proportion of missing values is time dependent. This implies that a quarter-by-quarter imputation of missing values is rendered extremely difficult, as no data for imputation is available for some quarters.

Finally, we deal with the rest of missing values. Here, two approaches are combined. If, for a given variable, less than 50% of observations are missing on the quarterly cross-section, we impute with the median value of the cross-section. Otherwise, we impute with zero—i.e., we assume that no year-on-year growth was observed for the variable.

Figure k: Effect on the Proportion of Missing Values Caused by the Two-Step Imputation of Missing Values.

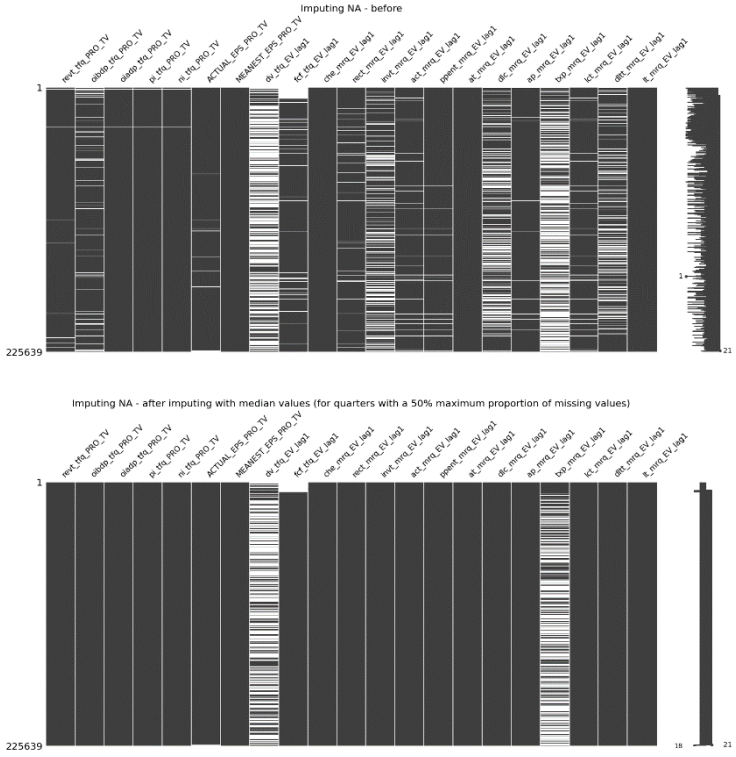
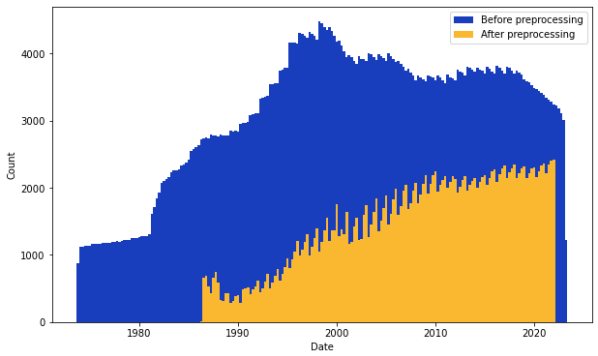


Figure k shows the evolution in the proportion of missing values for variables, following the imputation of missing values. The top panel shows the initial state of missing values, while the lower panel shows the state of missing values after the imputation using the median of the quarterly cross-section (if below the 50% threshold), but before the imputation using a zero-value (if above the 50% threshold). Note that the median imputation significantly reduces the proportion of missing values for most variables, with the exception of the variables *Dividends* (dv) and *Income Taxes Payable* (txp), for which the quarterly proportion of missing values was consistently above 50%. Most missing values for these two variables are thus imputed with zero-values.

2.3.4. Cutoff date

Finally, we decided to remove data past a certain date. The reason is that financial information from recent years might not be exhaustive yet, with some companies not having fully published their financial results. For this reason, we set a cutoff end date on Dec. 31st 2021, meaning the last financial information to be used in the modelling phase corresponds to the fourth calendar quarter of 2021. This cutoff can be visualized on Figure l.

Figure l: Visualization of the Effect on the Chronological Distribution of Data of the Introduction of a Cutoff Date.



3. Earnings Prediction Modelling

Now that that the dataset has been pre-processed, we can describe how the modelling phase is carried out. For reminder, the goal is to get the most accurate prediction of year-on-year net income—or equivalently, earning per share—growth. To do so, we will tune and fit models on the *training* dataset (*in-sample* period) and measure the associated prediction performances on the test dataset (*out-of-sample* period).

3.1. Regression Metric

In machine learning tasks, the selection of a regression metric is crucial to measure how well a model's predictions match the actual target values. By acting as a loss function, this regression metric enables the learning process to improve the ability to make accurate predictions. It does so at several stages of the pipeline:

- **The hyperparameter tuning phase**, where the *ideal set of model hyperparameters* is selected in such a way that a model fitted on the training set and scored on the (unseen) validation set optimizes the chosen loss function.
- **The training phase**, where the *ideal set of model parameters* is selected in such a way that a model fitted on the training set and scored the same (known) set optimizes the chosen loss function.
- **The testing phase**, where the *ideal model* is selected in such a way that models fitted on the training set and scored on the (unseen) test set optimize the chosen loss function. In our case, as we do not compare different machine learning models, the goal is to compare how regression metrics compare across the different scenarios considered for our research question—refer to 1.2.1.

We decided to stick to a standard approach for regression tasks; i.e., we used the mean squared error (MSE) as regression loss metric for all three phases (and the coefficient of determination R^2 for the testing phase).

3.2. Histogram-based Gradient Boosting Regression Tree

Our research question does not involve exhaustively finding the best regression performance among multiple models. Instead, we aim to compare the performance of two scenarios against a baseline—refer to 1.2.1. For this reason, we decided to focus on a single model, namely Histogram-based Gradient Boosting Regression Trees. This is an

advanced variant of Gradient Boosting Regression Trees (like the ubiquitous XGBoost algorithm), but with a focus on computational performance and scalability. Notably, LightGBM, initially developed by a Microsoft team, exemplifies this gradient-boosting framework [6].

Several reasons explain why this algorithm was chosen:

- It makes use of decisions trees, which are proven to be flexible learners adaptable to a wide range of tasks.
- It combines the predictions of multiple of these weak learners, which provides a more accurate and robust final prediction.
- The required computational power is low compared to standard gradient boosting algorithm, especially for big datasets where the number of samples is larger than tens of thousands of samples. For that reason, according to the Scikit-Learn documentation, these histogram-based estimators can be *"orders of magnitude faster"*. This advantage becomes even more apparent when compared to other algorithms, such as recurrent neural networks.

3.3. Hyperparameters optimization

3.3.1. Selected Hyperparameters

Each model has its specific set of hyperparameters to optimize. In the case of a Histogram-based Gradient Boosting model, we decided to on a set of five hyperparameters to optimize (retrieved from the Scikit-Learn documentation), namely:

- **The learning rate**: it determines the step size at which each tree's contribution is added to the overall model. A smaller learning rate makes the model converge slowly but might lead to better generalization.
- **The maximum number of iterations of the boosting**: this hyperparameter sets the maximum number of trees that will be built during the training process. Increasing the number of trees can improve the model's performance, but it may also lead to overfitting if not controlled properly.

[6] Since we make use of the default scikit-learn library, the estimator used is not *strictly* the LightGBM algorithm, but is heavily inspired by its framework.

- **The minimum number of samples per leaf**: it specifies the minimum number of samples required in a leaf (terminal node) of each tree. This parameter helps prevent overfitting by controlling the complexity of individual trees.
- **The maximum depth of each tree**: this hyperparameter limits the depth of the decision trees. A shallow tree prevents overfitting and is computationally efficient, while a deeper tree can capture more complex patterns in the data.
- **The L2 regularization parameter**: this parameter controls the amount of regularization applied to the model. It penalizes large coefficients in the model, helping to prevent overfitting and improve generalization.

These parameters were selected for their high sensitivity to the regression performance of the validation set. In other words, they have a strong influence on finding the right equilibrium between *underfitting* and *overfitting*.

3.3.2. Successive Halving Randomized Search

Hyperparameters tuning is known for being a time-consuming task. The standard approach is to follow an exhaustive grid search procedure, where all combinations (i.e., the cross-product) of hyperparameters within a specified space are tested thoroughly. As this search space grows, the number of tested combinations increases exponentially. For instance, optimizing three hyperparameters with five potential values each requires testing 125 models (5^3), and with ten potential values, it demands testing 1,000 models (10^3).

Additionally, for each hyperparameter combination, the model needs to be trained five times due to the five splits on the training set for obtaining the best cross-validation score (see 1.3.4). This effectively results in a further fivefold increase in the number of training sessions, reaching 625 and 5,000 training instances for the respective cases.

To reduce the number of hyperparameter combinations being tested, a randomized hyperparameter optimization was instead carried out. Instead of testing all possible combinations within the predefined space, we performed random sampling of hyperparameter combinations for a prespecified number of iterations. This ensures that regardless of the number of parameters and potential values, the total number of training instances remains capped. For example, if we limit the sampling iterations to 50, the

search space expansion won't lead to an increase in the tested models. Consequently, the total number of training instances is capped at 250 for both cases mentioned earlier.

Of course, this randomized procedure may result in a sub-optimal model compared to an exhaustive test of all combinations. However, we argue that this choice allows for a broader search space, increasing the likelihood of finding a more optimal combination. In essence, we prioritize seeking an *approximate* but *global* optimum over an *exact* but *local* optimum [7].

Still, to improve the precision of this global optimum, on top of this randomized search, we search the hyperparameter space using successive halving. This is an iterative selection process where all candidates (the hyperparameter combinations) are evaluated with a small amount of number of training samples at the first iteration. Only some of these candidates (in our case, the best performing third) are selected for the next iteration, which will be allocated more training samples. This process continues until all available training samples have been used, and the best-performing hyperparameter combination is then selected. Although more training iterations eventually take place (in our case, with a halving parameter of 3, training instances get multiplied by 1.5), the bulk of these training iterations is performed on a small subset of the training data, which overall accelerates the tuning process significantly.

The use of successive halving allows us to quickly eliminate poor-performing combinations and allocate greater processing power to the best-performing candidates only. As a result, a greater number of hyperparameters combination can be screened, but only a narrowed-down list of these combinations will be thoroughly tested. This enables us to explore a broad range of potential global optima in the first stage and then fine-tune this exploration in the second stage. Ultimately, by efficiently narrowing down the list of combinations to thoroughly test, we strike a balance between *exploration* and *precision* of the global optimum.

[7] If this issue was framed as a bias-variance tradeoff, we could say we favor minimizing bias over minimizing variance.

In our case, we settled on a set of 500 hyperparameters candidates, randomly sampled from the search space specified in 3.3.3. Figure m provide an illustration of the process of the iterative screening of candidates.

Figure m: Illustration of the Successive Halving Search Process for Hyperparameters.

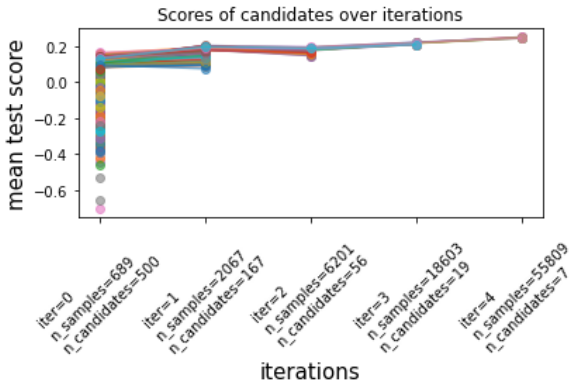


Figure m illustrates how hyperparameters combination candidates get successively selected, with progressively less candidates but more of training samples allocated at each iteration. Hyperparameters are fitted for a Histogram-based Gradient Boosting model in the case of scenario 2—i.e., where combined analysts' consensus forecasts and fundamental data from previous periods are taken as inputs. The y-axis score is the coefficient of determination R^2, computed as the average score across validation sets.

3.3.3. Hyperparameters Search Space and Final Values

For simplicity purposes, we use the same predefined search space for hyperparameters in both scenarios of our research question. Whether both analysts' consensus forecasts and fundamental data from previous periods are used as inputs or only the latter, the random sampling of hyperparameter combinations occurs within this identical search space.

The search space of these hyperparameter combinations as well as the optimized final combination for both scenarios is presented in Table 1. Note that either a distribution over possible values (sampled according to this specified distribution) or a list of discrete choices (sampled uniformly) is specified. Scikit-learn recommends specifying a distribution when possible, to take full advantage of the randomization.

Table 1: Hyperparameters Grid Search Space and Final Values.

Hyperparameters	Search space	Final value	
		Scenario 1 [1]	Scenario 2 [2]
learning_rate [3]	{min=0.001, max=1} Continuous log-uniform distribution	0.0084	0.0516
max_iter [4]	{min=15, max= 2000} Discrete log-uniform distribution	33	96
min_samples_leaf [5]	{min=5, max= 50} Discrete uniform distribution	39	33
max_depth [6]	[None, 2, 5, 10, 20, 50, 100, 500, 1000] Discrete distribution	100	2
l2_regularization [7]	[1, 0.5, 0.1, 0.01, 0.001, 0.0001, 0] Discrete distribution	0.01	0.01

1. Only fundamental data from previous periods is taken as input; 2. Combined analysts' consensus forecasts and fundamental data from previous periods are taken as inputs; 3. Learning rate; 4. Maximum number of iterations of the boosting; 5. Minimum number of samples per leaf; 6. Maximum depth of each tree; 7. L2 regularization parameter.

4. Fundamental Backtest

Now that the optimal values of hyperparameters for the two Gradient Boosting models (see Table 1) have been found, these two models can be fitted on the entire *training* dataset and scored on (still unseen) *test* data. The *training* data spans from 1986-Q1 to 2016-Q4, representing the *in-sample* period; the *test* data spans from 2017-Q1 to 2021-Q4, representing the *out-of-sample* period. This section aims at analyzing the performances of these two models over the *out-of-sample* period. These scores are then compared to the equivalent baseline scenario performance, where we directly score analysts' consensus forecasts.

4.1. Results on the Out-Of-Sample Period

Regression performance results have been made available in Table 2.

Table 2: In-Sample and Out-Of-Sample Scores of the Three Scenarios.

Regression metric	In-sample scores					Out-of-sample scores				
	Baseline Scenario	Scenario 1		Scenario 2		Baseline Scenario	Scenario 1		Scenario 2	
		HGBDT	OLS	HGBDT	OLS		HGBDT	OLS	HGBDT	OLS
MSE	0.7352	0.7625	0.8905	0.5800	0.7017	0.7984	0.9874	1.0285	0.7932	0.8111
R^2	0.2016	0.1721	0.0331	0.3702	0.2381	0.2274	0.0445	0.0047	0.2324	0.2151

Baseline Scenario: score of analysts' consensus forecasts; *Scenario 1:* score of the model where only fundamental data from previous periods is taken as input; *Scenario 2*: score of the model where combined analysts' consensus forecasts and fundamental data from previous periods are taken as inputs; *HGBDT:* Histogram-based Gradient Boosting Regression Trees model; *OLS:* Ordinary Least Square model; *MSE:* Mean Square Error; R^2*:* coefficient of determination.

We will first comment differences in performance (i) between scenario 1 & 2, then (ii) between the baseline scenario and scenario 2, and (iii) between the baseline scenario and scenario 1.

As expected, the out-of-sample performance of gradient-boosting models observed in scenario 2 is significantly better than the one observed in scenario 1, with respective R^2 values of 0.2324 and 0.0445. This shows that consensus forecasts are a powerful determinant of earnings, as the inclusion of this single variable leads to a fivefold improvement in the R^2 score. However, the magnitude of the difference is even greater with corresponding OLS models (respective values of 0.2151 and 0.0047), suggesting that the gradient-boosting model in scenario 1 still does a good job at exploiting the predictive power available in past fundamentals information.

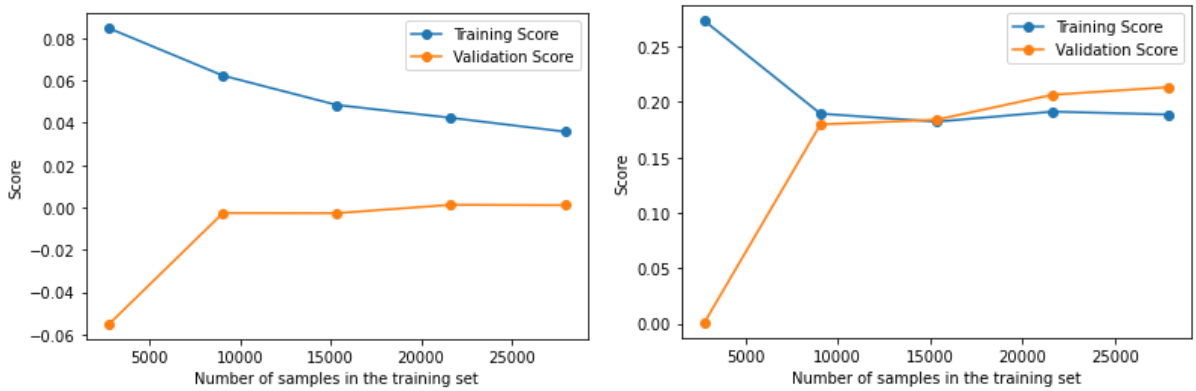
The difference in performance between the baseline scenario and scenario 2 is not significant (respective R^2 values of 0.2324 and 0.2274), indicating that adding past fundamental information to analysts' consensus forecasts only provides marginal improvements compared to analysts' consensus forecasts only. As this will be discussed in 4.2, this suggests that the potential of combining consensus forecasts with historical fundamentals is not fully exploited by the current gradient boosting model configuration.

Finally, the significantly better out-of-sample performance of the baseline scenario configuration over scenario 1 highlights that this study does not prove that machine learning techniques (in our case, histogram-based gradient boosting regression trees model) making use of fundamental data can provide a comparable result to analysts.

4.2. Learning Curves

Now that results have been obtained, we still want to assess how well models are learning from the data and if they are overfitting or underfitting. One way to proceed is to plot learning curves, which show how a model's performance changes as the amount of training data increases (see Figure n).

Figure n: Learning Curve for Both Models.



The left panel shows the learning curve in the case of scenario 1, where only fundamental data from previous periods is taken as input; the right panel shows the learning curve in the case of scenario 2, where combined analysts' consensus forecasts and fundamental data from previous periods are taken as inputs. In both panels, the y-axis score is the coefficient of determination R^2. The left panel hints a slight overfit (high variance), supporting a simplification of the model; in contrast, the right panel shows a clear underfit (high bias), motivating an increase in model complexity.

On the left panel, the learning curve for scenario 1 (when only fundamental data from previous periods is taken as input) indicates a pretty good balance between overfitting or underfitting, as training and validation curves tend to converge. Still, the gap between the scores is still pretty large, hinting a high variance. This means that additional data could

reduce this gap, but also that simplifying the model with fewer or less complex features could be beneficial.

On the left panel, the learning curve for scenario 2 (when combined analysts' consensus forecasts and fundamental data from previous periods are taken as inputs) converge very quickly, with validation scores even surpassing training scores. This indicates that the model probably underfits, that adding more examples to our model is not going to improve its performance, and that model complexity should probably be increased. Thus, great potential of improvement exists in this scenario if a more intricate model was used. This suggests that analysts' consensus forecasts show potential when combined with historical fundamentals, but that this potential is not fully exploited by the current gradient boosting model configuration. This idea is reinforced by the observation that the improvement in out-of-sample scores between HGBDT and OLS models is minimal (respective R^2 values of 0.2324 and 0.2151; see Table 2).

5. Conclusion

5.1. Study Approach and Results

The goal of this thesis was to assess the applicability of machine learning techniques in predicting earnings growth within the framework of the systematic management of an equity portfolio. The core premise driving this thesis is that an accurate earning prediction leads to the selection of better-performing stocks, thus enabling the development of a long holding, systematic value investing strategy.

To this end, we focused on predicting one-year earnings growth using historical company fundamental data. To ensure smoothness and comparability across different quarters, we used *cumulative trailing twelve-month* values for income statement and cash-flow statement items—meaning we sum-up the values of the past four consecutive quarters. Balance sheet items were represented by their *most recent quarter* values. Considering the importance of stationarity, we computed the relative difference (one-year growth) for all variables compared to their value in the previous period.

This arrangement generated infinite values and outliers. We treated the infinite values as missing values and processed the outliers using a combination of cross-section winsorization and non-linear bijective transformation. Any records with missing consensus forecast values were dropped, and remaining missing values were imputed using the median value of the cross-section or zero.

This preprocessing left a set of 215,296 records (or 6,964 companies) useable for the modelling phase, which were split according to an 80%-20% ratio. The corresponding in-sample and out-of-sample period were respectively from 1986-Q1 to 2016-Q4 and 2017-Q1 to 2021-Q4. We chose histogram-based gradient boosting regression trees for modeling, as these algorithms offer robust predictions with relatively low computational requirements. The model hyperparameters were optimized on validation sets using a successive halving randomized search process.

In the evaluation phase, we compared the out-of-sample predictive performance of this approach with that of analysts' consensus forecasts alone and a similar approach combining analysts' consensus forecasts and company fundamentals as model input data. Three takeaways emerged from our study:

- The results do not establish that machine learning techniques, specifically histogram-based gradient boosting regression trees, making use of fundamental data alone can offer a comparable result to analysts' forecasts. Subsequently, an investing strategy solely relying on these earnings predictions is not expected to outperform a strategy relying on analysts' forecasts.
- Combining historical fundamental information to analysts' consensus forecasts in machine learning techniques only provides marginal improvements compared to (unprocessed) analysts' consensus forecasts. However, this does not imply that an investment strategy relying on these earnings predictions would yield negligible benefits, as in finance even small differences in prediction performance can significantly impact an investor's equity.
- The potential of combining consensus forecasts with historical fundamentals is not fully exploited by the chosen gradient boosting model configuration, suggesting that more intricate models could lead to further improvements.

5.2. Limitations and Potential Research Improvements

Upon concluding this thesis, several limitations have emerged. These limitations present valuable opportunities for future research, and it is worth discussing their significance in relation to the obtained results.

5.2.1. Biases in the Company Universe

The first limitation has to do with sampling biases in the company universe.

The initial sampling process may have induced a survivorship bias, given the criteria that were specified (see D.1), notably the minimum consecutive listing period of 16 quarters, and the minimum inflation-adjusted market capitalization of $100 million. Given the greater size of sampled companies, the prediction of earnings might have been made easier.

A similar issue occurs with the selection of companies for which earnings forecasts were available (see 2.3.3). It is conceivable that missing earnings forecasts are correlated with greater forecast errors (if forecasts are made for companies with more easily predictable earnings). This would imply that this filtering artificially inflated the score of analysts' consensus forecasts.

5.2.2. Databases Compatibility

Another limitation has to do with the compatibility of the two databases being used. Although we used the two databases for different variables, structural differences can still have an impact.

We used the IBES database to retrieve consensuses *forecasts* of earnings, but IBES provides *actual* fundamentals as well—see A.1. However, these retrieved *actuals* from Compustat and IBES are not necessarily identical. Two reasons relevant to our case explain in part this incompatibility, which apply to IBES consensuses *forecasts* as well (IBES ensures conformity between its reported *actuals* and *forecasts*):

- **Point-in-time vs. Restated Values:** Both databases commonly use *restated* data, but Compustat offers *point-in-time* data as well—which we used. The former refers to data that has been adjusted after its initial release, while the latter refers to first-published data, meaning no correction get made after publishments. For instance, Refinitiv warns that brokers may be removed from the IBES database upon their request and consensus data may be corrected in the days and weeks following their release [8] (Wharton Research Data Services, n.d.). Not only does this affect the comparability between equivalent fundamental items, but the use of restated data implies that there is a risk of incorporating "future" information into our backtest analysis. This introduces look-ahead bias, which can distort the results and create an artificial appearance of improved performance (Amen, 2020).
- **Differences in Accounting Reporting Standards:** Compustat and IBES reported earnings differ due to differences in reporting standards. While the former is based on financial statements using GAAP reporting (standardized), the latter uses "street" earnings. That is, IBES earnings exclude various expenses required by GAAP [9].

However, Livnat & Mendenhall (2006) argue that *"these factors* [restatements & reporting standards] *do not explain a significant portion of the differences* […] *in reported earnings between the two sources"*. Essentially, this means that most of the observed differences

[8] Figures older than 6 months are generally stable, but exceptions arise.

[9] Some expenses omitted from street earnings can be found in Compustat's variable "Special Items" and can be backed out (among others), but this has not been further investigated. See pp. 178-179 Livnat & Mendenhall (2006) for more details.

between the two databases are neither due to a policy of restating earnings, nor to a difference in reporting standards. Although differences between databases reporting constitute a limit of this research, this statement summarizes why no further effort has been put into ensuring greater compatibility.

5.2.3. More Diverse Set of Information Types

In this study, only fundamental data is considered as input data. However, as explained in 1.1.2, technical indicators and sentiment data proved to provide significant improvement to prediction. Research shows that a great potential exists in combining features from these different domains (Matuozzo et al., 2022).

5.2.4. Alternative Target Variable

This study has focused on predicting net income growth. However, predicting net income is challenging due to its dependence on various accounting elements, unlike other income statement metrics such as revenue or EBIT. Moreover, revenue-related metrics have gained increasing importance, given that net income metrics are easily manipulable by executives—due to companies having greater control over costs than revenue (Gopalan et al., 2017). This makes revenues more likely to be correlated with company valuation.

Therefore, we recommend future research in this area to consider alternative company performance metrics, such as revenue or EBIT, even if it comes at the expense of lower data availability. This approach is suggested for both practical reasons, as these metrics might be easier to predict, and financial reasons, as they could be more closely related to company valuation.

5.2.5. Alternative Regression Metrics

One of the main limitations of the adopted approach is its high sensitivity to outliers, due to the computation of the relative change of features. This motivated the modification of the target variable (see 2.3.2), for both the train and test datasets, contrary to standard practice. Modification on the test set can artificially inflate the computed performance metrics.

Further investigation into alternative regression metrics robust to outliers could be carried out, to avoid making this decision. One idea could be to use the median absolute deviation (MedAE) as the main objective function for modelling.

5.2.6. Classification and Regression Tasks

It is possible that framing the prediction problem in terms of regression was too challenging—especially given the available processing power and current experience in machine learning applications—and that framing this problem in terms of classification would have been savvier. Researchers have indeed pointed out several difficulties with examining the extent of earnings change (regression), compared to predicting the direction of earnings changes (classification)—see CHEN et al. (2022). In addition to being more actionable, predicting the direction of earnings changes is also economically meaningful as many portfolios are constructed using the forecasted direction of earnings changes (Wahlen & Wieland, 2011). In our case, a classification task would also have rendered any modification of the target variable useless.

However, adopting this approach would have limited the model's ability to detect the magnitude of fluctuations in earnings. An alternative idea proposed by Matuozzo et al. (2022) is to develop "*models learning jointly regression and classification tasks*". This approach would enable both the combined prediction of (i) the direction of earnings change and (ii) the magnitude of this change. By doing so, it would address the individual shortcomings of regression and classification tasks, namely situations with accurate point estimate but incorrect directions, or accurate direction classification but with a dismissible magnitude of change.

5.2.7. Single Model Robustness

In this study, only one category of model was tuned, fitted, and scored—namely histogram-based gradient boosting regression trees. Even though the model was chosen given its high flexibility, results have shown that investigating other models could lead to further improvements in prediction performance and reinforce the robustness of conclusions (see 4.2).

5.2.8. Single Scenario Robustness

Finally, when assessing the performance of models on data from the out-of-sample period, we have effectively examined only one specific scenario, which is the historical sequence of data points. Our results are dependent upon this specific order in which the data was observed, leading to potential bias. We must recognize that this scenario does not guarantee that the proposed methods perform similarly for different out-of-sample periods.

To address this limitation, López De Prado (2018) introduces the combinatorial purged cross-validation method, which generates multiple diverse paths for evaluation. This approach allows for a more robust assessment of backtesting performance (be it *fundamental* backtesting in our case, or *return* backtesting), as it considers different data sequences and provides a more reliable estimation of the models' future effectiveness.

Bibliography

Abarbanell, J. S., & Bushee, B. J. (1997). Fundamental Analysis, Future Earnings, and Stock Prices. *Journal of Accounting Research*, *35*(1), 1–24.

Abarbanell, J. S., Lanen, W. N., & Verrecchia, R. E. (1995). Analysts' forecasts as proxies for investor beliefs in empirical research. *Journal of Accounting and Economics*, *20*(1), 31–60. https://doi.org/10.1016/0165-4101(94)00392-I

Alberg, J., & Lipton, Z. C. (2018). Improving Factor-Based Quantitative Investing by Forecasting Company Fundamentals. *ArXiv:1711.04837 [Cs, Stat]*. http://arxiv.org/abs/1711.04837

Amen, S. (2020, August 7). *Why is point-in-time data crucial in backtesting?* Refinitiv Perspectives. https://www.refinitiv.com/perspectives/future-of-investing-trading/why-is-point-in-time-data-crucial-in-backtesting/

Baker, B. (2022, October 28). *Which Factors Determine Stock Prices?* Bankrate. https://www.bankrate.com/investing/what-makes-a-stock-go-up-in-price/

Bustos, O., & Pomares-Quimbaya, A. (2020). Stock market movement forecast: A Systematic review. *Expert Systems with Applications*, *156*, 113464. https://doi.org/10.1016/j.eswa.2020.113464

Cavalcante, R. C., Brasileiro, R. C., Souza, V. L. F., Nobrega, J. P., & Oliveira, A. L. I. (2016). Computational Intelligence and Financial Markets: A Survey and Future Directions. *Expert Systems with Applications*, *55*, 194–211. https://doi.org/10.1016/j.eswa.2016.02.006

Chauhan, L., Alberg, J., & Lipton, Z. C. (2020). *Uncertainty-Aware Lookahead Factor Models for Quantitative Investing* (arXiv:2007.04082). arXiv. https://doi.org/10.48550/arXiv.2007.04082

CHEN, X., CHO, Y. H. (TONY), DOU, Y., & LEV, B. (2022). Predicting Future Earnings Changes Using Machine Learning and Detailed Financial Data. *Journal of Accounting Research*, *60*(2), 467–515. https://doi.org/10.1111/1475-679X.12429

Dai, R. (2020, May 11). *I/B/E/S @WRDS 101—Introduction and Research Guide*. https://view.officeapps.live.com/op/view.aspx?src=https%3A%2F%2Fwrds-www.wharton.upenn.edu%2Fdocuments%2F1372%2FIBES_RUI.pptx%23%3A~%3Atext%3DForecast%2520Period%2520Indicator%2520%2528FPI%2529%253A%2520a%2520code%2520to%2520identify%2CFlag%2520%2528PDF%2529%253A%2520%25C2%25A0share%2520base%2520selected%2520for%2520an%2520estimate&wdOrigin=BROWSELINK

Dreman, D. N., & Berry, M. A. (1995). Analyst Forecasting Errors and Their Implications for Security Analysis. *Financial Analysts Journal*, *51*(3), 30–41. https://doi.org/10.2469/faj.v51.n3.1903

Egan, J. (2023, March 25). How Are Stock Prices Determined: The Factors that Affect Share Prices of Listed Companies. *Time*. https://time.com/personal-finance/article/how-are-stock-prices-determined/

Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of Finance*, *25*(2), 383–417. https://doi.org/10.2307/2325486

Fama, E. F., & French, K. R. (1992). The Cross-Section of Expected Stock Returns. *The Journal of Finance*, *47*(2), 427–465. https://doi.org/10.2307/2329112

Fisher, I. (1896). What is Capital? *The Economic Journal*, *6*(24), 509–534. JSTOR. https://doi.org/10.2307/2957184

Flovik, V. (2018, June 7). *How (not) to use Machine Learning for time series forecasting: Avoiding the pitfalls*. Medium. https://towardsdatascience.com/how-not-to-use-machine-learning-for-time-series-forecasting-avoiding-the-pitfalls-19f9d7adf424

Gopalan, R., Horn, J., & Milbourn, T. (2017). Comp targets that work. In *Harvard Business Review* (Vol. 95, Issue 5, pp. 102–107). HARVARD BUSINESS SCHOOL PUBLISHING CORPORATION 300 NORTH BEACON STREET ….

Graham, B. (1965). *The Intelligent Investor*. Harper & Row.

Granger, C. W., & Newbold, P. (1974). Spurious regressions in econometrics. *Journal of Econometrics*, *2*(2), 111–120.

Lan, C., Moneta, F., & Wermers, R. (2023). Holding Horizon: A New Measure of Active Investment Management. *Journal of Financial and Quantitative Analysis*, 1–80. https://doi.org/10.1017/S0022109023000303

Livnat, J., & Mendenhall, R. R. (2006). Comparing the Post–Earnings Announcement Drift for Surprises Calculated from Analyst and Time Series Forecasts. *Journal of Accounting Research*, *44*(1), 177–205. https://doi.org/10.1111/j.1475-679X.2006.00196.x

López De Prado, M. (2018). *Advances in financial machine learning*. John Wiley & Sons.

Malkiel, B. G. (2003). The Efficient Market Hypothesis and Its Critics. *Journal of Economic Perspectives*, *17*(1), 59–82. https://doi.org/10.1257/089533003321164958

Matuozzo, A., Yoo, P., Provetti, A., & Kim, M. (2022, September 16). *Machine learning methods for Equity Time Series forecasting: A compendium* [Conference]. 31st ACM International Conference on Information and Knowledge Management, Atlanta, U.S. https://amlts.github.io/amlts2022/

Park, C.-H., & Irwin, S. H. (2007). What do we know about the profitability of technical analysis? *Journal of Economic Surveys*, *21*(4), 786–826.

Reed, P. (2017, June 27). Getting a linking table from Compustat via WRDS Cloud using SSH and SAS. *Business Research Plus*. https://bizlib247.wordpress.com/2017/06/27/getting-a-linking-table-from-compustat-via-wrds-cloud-using-ssh-and-sas/

Refinitiv. (n.d.-a). *I/B/E/S Estimates*. Retrieved August 7, 2023, from https://www.refinitiv.com/en/financial-data/company-data/ibes-estimates

Refinitiv. (n.d.-b). *S&P Compustat Database*. Retrieved August 7, 2023, from https://www.refinitiv.com/en/financial-data/company-data/fundamentals-data/standardized-fundamentals/sp-compustat-database

Tsay, R. S. (2005). *Analysis of financial time series*. John wiley & sons.

Wahlen, J. M., & Wieland, M. M. (2011). Can financial statement analysis beat consensus analysts' recommendations? *Review of Accounting Studies*, *16*(1), 89–115. https://doi.org/10.1007/s11142-010-9124-5

Walasek, R., & Gajda, J. (2021). Fractional differentiation and its use in machine learning. *International Journal of Advances in Engineering Sciences and Applied Mathematics*, *13*(2), 270–277. https://doi.org/10.1007/s12572-021-00299-5

Wang, Y., Liu, Z., & Wang, X. (2022). The supply of analysts and earnings forecasts. *International Review of Financial Analysis*, *84*, 102404. https://doi.org/10.1016/j.irfa.2022.102404

Whittaker, J., Whitehead, C., & Somers, M. (2005). The Neglog Transformation and Quantile Regression for the Analysis of a Large Credit Scoring Database. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *54*(5), 863–878.

Wieland, M. M. (2011). Identifying Consensus Analysts' Earnings Forecasts that Correctly and Incorrectly Predict an Earnings Increase. *Journal of Business Finance & Accounting*, *38*(5–6), 574–600. https://doi.org/10.1111/j.1468-5957.2011.02236.x

Appendices

A. Data Fetching Procedure

A.1. Source

All the data has been retrieved from two different databases available online on the Wharton Research Data Services (WRDS) platform, namely:

- **the Compustat database**—"*a comprehensive market and corporate financial database published by Standard and Poor's, covering thousands of companies worldwide, with info dating as far back as 1950"* (Refinitiv, n.d.-b)—from which company fundamentals (Income statement, Cash Flow statement, Balance Sheet) and market information (splits, prices & returns) were retrieved. More specifically, *Compustat North America* and *Compustat Snapshot* databases were used, which cover companies in the United States and Canada. The former offers restated data, while the latter provides first-published data, which is crucial for backtesting analyses.
- **the IBES database** (Institutional Brokers' Estimate System, often abbreviated as I/B/E/S)—a compilation of *"different estimates made by stock analysts on the future earnings for publicly traded companies"* (Refinitiv, n.d.-a) provided by Refinitiv / Thomson Reuters—from which company consensuses data was obtained. Fundamental data is also offered (known as '*Actuals*' in the IBES jargon, as opposed to '*Estimates*'), but is much scarer than equivalent data sourced from Compustat.

A.2. Retrieving Identification Data & Filtering of the Company Universe

To reduce the volume of data to be handled, the initial step involved exporting only essential identification and market information (such as Name, Ticker, Stock Exchange Code, GIC group, and Market Capitalization; see B.1) from all American companies available in the *Compustat North America* universe between 1962-01 and 2022-12. This resulted in a list of 36,114 companies.

Since *Compustat* would eventually be used to obtain company fundamentals, this list of companies was directly sourced from the *Fundamentals Quarterly* database, which provides company fundamental information on a quarterly basis.

We applied several filtering criteria to this set, narrowing it down to 8,379 companies, as detailed in D.1. The goal was not only to ensure that the selected company fundamentals had sufficient predictive power but could also be matched with IBES information.

A.3. Retrieving a Compustat-IBES Linking Table

A linking table must be obtained from Compustat, in order to associate companies from the IBES universe to the Compustat universe, as the databases use two different sets of unique identifiers ('*gvkey*' for Compustat, '*ibtic*' for IBES).

Contrary to the rest of the data, this table cannot be retrieved from the *WRDS* website directly. Instead, we must proceed via *WRDS* Cloud using SSH and SAS. This can be achieved through the procedure detailed here (Reed, 2017).

A.4. Retrieving Fundamental Data

Following the selection of the sample population of companies, fundamentals for each company-quarter combination were retrieved from *Compustat Snapshot* through the *WRDS* website, more specifically from the *Fundamentals Quarterly* database. Having priorly filtered companies enabled us to specify the list of desired companies, instead of searching in the entire *Compustat* universe. This list consisted of 8,379 companies, representing 620,918 company-quarter combinations.

The selected query variables are specified in B.3. Note that some data comes from the *Compustat North America* database, which consists of restated data. The reason is that some information were not available (e.g., quarter share close price) or inconsistent (e.g., company names and ticker changing overtime) in the *Compustat Snapshot* database. However, none of these variables served as input data in the final model.

A.5. Retrieving Market Data

Using the same sample population of companies, market information was also retrieved from Compustat through *WRDS*, this time from the *Security Monthly* database. This list consisted of 8,379 companies, representing 2,016,115 company-month combinations.

The selected query variables are specified in B.4.

A.6. Retrieving Consensuses Data

The procedure is essentially similar to the ones described for the Compustat databases, except that data was retrieved from IBES' *Summary History - Summary Statistics* database.

Therefore, the list of provided companies consisted in IBES-specific company codes (‘*ibtic*’), instead of the Compustat equivalent (‘*gvkey*’). This list consisted of 7,887 companies, representing 14,718,136 company-estimate-day combinations.

Here, the data structure differs significantly from the Compustat data structure. Appendix D.2 provides an overview of the steps carried out to ensure compatibility.

The time coverage offered by IBES data is much scarer, with first data points dating back to 1983-06. This implies that merging the two databases will result in the reduction of the time coverage offered by Compustat.

The specified forecast horizons were between 1 and 4 quarters. The reason is that these four data points will eventually be summed to provide a cumulative trailing twelve month figure (as detailed in D.4).

The selected query variables are specified in B.5.

B. Lists of variables fetched from databases

B.1. Compustat Company Sampling

Table 3: Lists of Compustat Variables Retrieved during the Company Sampling Process

Variable	Mnemonic	Category	Eventual use
Compustat unique identifier	gvkey	ID	ID
Quarter end date	datadate	ID	ID
Fiscal year	fyearq	ID	Discarded
Fiscal quarter number	fqtr	ID	Discarded
Industry Format	indfmt	Form Parameters	Discarded
Consolidation Level	consol	Form Parameters	Discarded
Population Source	popsrc	Form Parameters	Discarded
Data Format	datafmt	Form Parameters	Discarded
Official ticker	tic	ID	Discarded
Company name	conm	ID	Discarded
Currency	curcdq	Form Parameters	Discarded
Calendar Data Year and Qtr.	datacqtr	ID	Universe filtering
Fiscal Data Year and Qtr.	datafqtr	ID	Discarded
Common Shares Outstanding	cshoq	Market Information	Universe filtering
Stock exchange code	exchg	ID	Universe filtering
Company Status	costat	Form Parameters	Universe filtering
Quarter Price Close	prccq	Market Information	Universe filtering
GIC Groups	ggroup	ID	Universe filtering

ID: Identification.

B.2. Compustat-IBES linking table

Table 4: Lists of Variables Retrieved in the Compustat-IBES Linking Table

Variable	Mnemonic	Category	Eventual use
Official ticker	tic	ID	Discarded
Compustat unique identifier	gvkey	ID	Linking Compustat-IBES
Security Issue ID	iid	ID	Discarded
Alternative Comp. unique id.	cusip	ID	Discarded
Security Inactivation Code	dlrsni	ID	Discarded
Security Description	dsci	ID	Discarded
Earnings Participation Flag	epf	ID	Discarded
Stock exchange code	exchg	ID	Discarded
Stock Ex. Country Code	excntry	ID	Discarded
IBES unique identifier	ibtic	ID	Linking Compustat-IBES
Int. Securities Id. Number	isin	ID	Discarded
Security Status Marker	secstat	ID	Discarded
SEDOL unique identifier	sedol	ID	Discarded
Issue Type	tpci	ID	Discarded
Security Inactivation Date	dldtei	ID	Discarded

ID: Identification.

B.3. Compustat Fundamental data

Table 5: Lists of Compustat Fundamental Variables

Variable	Mnemonic	Source	Category	Eventual use
Compustat unique identifier	gvkey	SS; NA	ID	ID
Company name	conm	SS; NA	ID	ID
Official ticker	tic	SS; NA	ID	ID
Fiscal Year-end Month	fyr	SS	ID	ID
Calendar quarter	cqtr	SS	ID	ID
Calendar year	cyearq	SS	ID	ID
Fiscal quarter	fqtr	SS	ID	ID
Fiscal year	fyearq	SS	ID	ID
Quarter end date	datadate	SS; NA	ID	ID
Report Date of Earnings	rdq	SS	ID	ID
Revenue	revtq	SS	Fundamental-IS	Train-Test
EBITDA	oibdpq	SS	Fundamental-IS	Train-Test
EBIT	oiadpq	SS	Fundamental-IS	Train-Test
Pre-tax Income	piq	SS	Fundamental-IS	Train-Test
Net Income	niq	SS	Fundamental-IS	Train-Test
Dividends	dvy	SS	Fundamental-CF	Train-Test
Operating Cash Flow	oancfy	SS	Fundamental-CF	Compute Free-Cash-Flow
Capital Expenditures	capxy	SS	Fundamental-CF	Compute Free-Cash-Flow
Cash and cash equivalents	cheq	SS	Fundamental-BS	Train-Test
Receivables	rectq	SS	Fundamental-BS	Train-Test
Inventories	invtq	SS	Fundamental-BS	Train-Test
Total Current Assets	actq	SS	Fundamental-BS	Train-Test
Property, Plant & Equipment	ppentq	SS	Fundamental-BS	Train-Test
Intangible Assets	intanq	SS	Fundamental-BS	Discarded
Total Assets	atq	SS	Fundamental-BS	Train-Test
Short-term Debt	dlcq	SS	Fundamental-BS	Train-Test
Accounts Payables	apq	SS	Fundamental-BS	Train-Test
Income Taxes Payable	txpq	SS	Fundamental-BS	Train-Test
Total Current Liabilities	lctq	SS	Fundamental-BS	Train-Test
Long-term Debt	dlttq	SS	Fundamental-BS	Train-Test
Total Liabilities	ltq	SS	Fundamental-BS	Train-Test
Redeemable Noncontr. Inter.	mibtq	SS	Fundamental-BS	Discarded
Total Stockholders' Equity	teqq	SS	Fundamental-BS	Discarded
Common Shares Outstanding	cshoq	SS	Market Information	Discarded
Common Shares Used to Calculate Earnings Per Share	cshprq	SS	Market Information	Discarded
Quarter Close Price	prccq	NA	Market Information	Discarded
Total Market Value	mkvaltq	NA	Market Information	Discarded

NA: Compustat North America (Restated); *SS*: Compustat Snapshot (Point-in-time); *IS*: Income Statement; *CF*: Cash-Flow; *BS*: Balance Sheet; *ID*: Identification

B.4. Compustat Market data

Table 6: Lists of Compustat Market Variables

Variable	Mnemonic	Category	Eventual use
Compustat unique identifier	gvkey	ID	ID
Security Issue ID	iid	ID	ID
Month End Date	datadate	ID	ID
Official ticker	tic	ID	ID
Company name	conm	ID	ID
Price Adjustment Factor	ajexm	Market Information	Discarded
Month Close Price	prccm	Market Information	Discarded
Monthly Total Return	trt1m	Market Information	Discarded
Shares Outstanding (qtr.)	cshoq	Market Information	Discarded
Shares Outstanding (month)	cshom	Market Information	Discarded
Calendar year	cyear	Market Information	Discarded
Issue Type	tpci	ID	ID

ID: Identification. The reason why no market information ended up being used is that this study initially envisioned the realization of a return backtest (later staged pictured in Figure a)—explaining the need for market data.

B.5. IBES consensuses data

Table 7: Lists of IBES Consensuses Forecasts and Actual Variables

Variable	Mnemonic	Category	Eventual use
IBES unique identifier	TICKER	ID	ID
Official ticker	OFTIC	ID	ID
Consensus prediction date	STATPERS	ID	ID
Q1 horizon results announcement date	ANNDATSACT_Q1	ID	Discarded
Q2 horizon results …	ANNDATSACT_Q2	ID	Discarded
Q3 horizon results …	ANNDATSACT_Q3	ID	Discarded
Q4 horizon results …	ANNDATSACT_Q4	ID	ID
Q1 horizon period end date	FPEDATS_Q1	ID	Discarded
Q2 horizon period end date	FPEDATS_Q2	ID	Discarded
Q3 horizon period end date	FPEDATS_Q3	ID	Discarded
Q4 horizon period end date	FPEDATS_Q4	ID	ID
Q1 actual EPS	ACTUAL_Q1_EPS	Actual	Add up 4 quarters (TTM)
Q2 actual EPS	ACTUAL_Q2_EPS	Actual	Add up 4 quarters (TTM)
Q3 actual EPS	ACTUAL_Q3_EPS	Actual	Add up 4 quarters (TTM)
Q4 actual EPS	ACTUAL_Q4_EPS	Actual	Add up 4 quarters (TTM)
Q1 estimated EBIT	MEANEST_Q1_ EBI	Forecast	Discarded
Q1 estimated EBITDA	MEANEST_Q1_ EBT	Forecast	Discarded
Q1 estimated EPS	MEANEST_Q1_ EPS	Forecast	Add up 4 quarters (TTM)
Q1 estimated Net Income	MEANEST_Q1_ NET	Forecast	Discarded
Q1 estimated Pre-tax Profit	MEANEST_Q1_ PRE	Forecast	Discarded
Q1 estimated Revenue	MEANEST_Q1_ SAL	Forecast	Discarded
Q2 estimated EBIT	MEANEST_Q2_ EBI	Forecast	Discarded
Q2 estimated EBITDA	MEANEST_Q2_ EBT	Forecast	Discarded
Q2 estimated EPS	MEANEST_Q2_ EPS	Forecast	Add up 4 quarters (TTM)
Q2 estimated Net Income	MEANEST_Q2_ NET	Forecast	Discarded
Q2 estimated Pre-tax Profit	MEANEST_Q2_ PRE	Forecast	Discarded
Q2 estimated Revenue	MEANEST_Q2_ SAL	Forecast	Discarded
Q3 estimated EBIT	MEANEST_Q3_ EBI	Forecast	Discarded
Q3 estimated EBITDA	MEANEST_Q3_ EBT	Forecast	Discarded
Q3 estimated EPS	MEANEST_Q3_ EPS	Forecast	Add up 4 quarters (TTM)
Q3 estimated Net Income	MEANEST_Q3_ NET	Forecast	Discarded
Q3 estimated Pre-tax Profit	MEANEST_Q3_ PRE	Forecast	Discarded
Q3 estimated Revenue	MEANEST_Q3_ SAL	Forecast	Discarded
Q4 estimated EBIT	MEANEST_Q4_ EBI	Forecast	Discarded
Q4 estimated EBITDA	MEANEST_Q4_ EBT	Forecast	Discarded
Q4 estimated EPS	MEANEST_Q4_ EPS	Forecast	Add up 4 quarters (TTM)
Q4 estimated Net Income	MEANEST_Q4_ NET	Forecast	Discarded
Q4 estimated Pre-tax Profit	MEANEST_Q4_ PRE	Forecast	Discarded
Q4 estimated Revenue	MEANEST_Q4_ SAL	Forecast	Discarded
Most recent period end date	INT0DATS	ID	ID

ID: Identification; *TTM*: trailing twelve month. The reason why most forecast information was discarded is that we envisioned implementing multi-task learning, with several target variables. In this case, several income statement forecast variables would have been necessary.

C. List of Features used in Modeling

Table 8: Lists of Features used in Modeling

Variable	Mnemonic	Category	Source	Preprocessing				Modelling	
				MRQ	TTM	3-y. lag	Other	TV	EV
Revenue	revt	Fundamental-IS	Compustat SS		×	×			×
EBITDA	oibdp	Fundamental-IS	Compustat SS		×	×			×
EBIT	oiadp	Fundamental-IS	Compustat SS		×	×			×
Pre-tax Income	pi	Fundamental-IS	Compustat SS		×	×			×
Net Income	ni	Fundamental-IS	Compustat SS		×	×			×
Dividends	dvy	Fundamental-CF	Compustat SS		×	×			×
Free Cash Flow	fcf	Fundamental-CF	Compustat SS		×	×	Derived from Operating Cash Flow and Capital Expenditures		×
Cash and cash equivalents	che	Fundamental-BS	Compustat SS	×		×			×
Receivables	rect	Fundamental-BS	Compustat SS	×		×			×
Inventories	invt	Fundamental-BS	Compustat SS	×		×			×
Total Current Assets	act	Fundamental-BS	Compustat SS	×		×			×
Property, Plant & Equipment	ppent	Fundamental-BS	Compustat SS	×		×			×
Total Assets	at	Fundamental-BS	Compustat SS	×		×			×
Short-term Debt	dlc	Fundamental-BS	Compustat SS	×		×			×
Accounts Payables	ap	Fundamental-BS	Compustat SS	×		×			×
Income Taxes Payable	txp	Fundamental-BS	Compustat SS	×		×			×
Total Current Liabilities	lct	Fundamental-BS	Compustat SS	×		×			×
Long-term Debt	dltt	Fundamental-BS	Compustat SS	×		×			×
Total Liabilities	lt	Fundamental-BS	Compustat SS	×		×			×
Actual EPS	ACTUAL_EPS	Forecast	IBES		×	×		×	×
Estimated EPS	MEANEST_EPS	Forecast	IBES		×				(Scenario 2 only)

MRQ: Most Recent Quarter; *TTM*: Trailing Twelve Months; *TV*: Target Variable; *EV*: Explanatory Variable; *Compustat SS*: Compustat Snapshot (Point-in-time); *3-y. lag*: three-year lag of past fundamentals.

D. Description of Python Scripts

Five different python scripts were used to carry out the analysis, namely:

- *01_CompanySampling.py;*
- *02_IBES_preprocessing.py;*
- *03_COMP_preprocessing.py;*
- *04_Merged_preprocessing.py; and*
- *05_Modeling.py*

Their purpose is detailed here below.

D.1. Company Sampling

This piece of code aims at using retrieved company identification data (as described in A.2) and subsequently filtering the universe.

The following filters were applied:

- **Minimum consecutive listing period of 16 quarters.** Since the goal is to use lagged fundamental data as input variables, it is necessary to ensure that a sufficient number of consecutive data points for a given company is available. 16 quarters implies that data for a window of 4 consecutive years is available (the first 3 years for explanatory lag variables, the last year for target variables).
- **Exclusion of financial companies, pursuant to a standard approach in research**. Researchers often argue that the concept of balance sheet leverage holds distinct implications for financial companies compared to operating companies (Fama & French, 1992). While the validity of this decision may be debated in our case, we exclude financial companies for the sake of consistency with previous research.
- **Only active (and inactive) companies which are (have been) traded on NYSE, American SE (now NYSE American) or NASDAQ.** The reason is that these exchanges are among the largest and most liquid stock exchanges globally. This means these exchanges offer a wide data availability, represent a broad cross-section of various industry sectors, and importantly, are considered research benchmarks.
- **Minimum inflation-adjusted market capitalization of $100 million**, adjusted in December 2020 terms. A company is excluded from the sample only if it never reached

such market value throughout its full listing period. The reason is that these stocks are more representative of the overall market, offer greater liquidity and data availability, and attract institutional investor interest, ensuring statistical robustness in the analysis. However, we acknowledge that this selection criteria introduces potential biases, such as survivorship bias, which could be interesting topics for future investigation but are beyond the scope of this research.

- **Dual availability of company information in the Compustat and IBES databases**. Here, we ensure that only records on which an inner join can be performed (see D.4) are kept. Note that this does not ensure that each specific company-quarter combination is available in both databases, simply that this company possesses some records in both databases. As mentioned previously, it's important to note that this criterion may introduce a selection bias, as certain categories of companies may be more likely to be tracked by analysts. Nonetheless, this requirement is essential for addressing the specific research question at hand.

D.2. IBES Preprocessing

This piece of code deals with IBES Consensuses data specific preprocessing. Two main actions are being carried out: a reshaping of IBES data structure to fit with the Compustat architecture, and a filtering of the redundant forecast data points.

The primary key structure differs between IBES consensuses data and Compustat data. Compustat fundamental data is organized based on the combination of company and quarters, whereas IBES consensuses data is organized based on the combination of company, prediction date, forecast horizon, and the estimated measure (see Table 9).

Table 9: Raw Databases Structure Comparison

Compustat Fundamental data					
Variable	**Company identifier**	**Quarter end date**	**Company ticker**	**Revenue**	**…**
Mnemonic	*gvkey*	*datadate*	*tic*	*revtq*	…
Instance	⋮	⋮	⋮	⋮	⋱
	1690	2022-09-30	AAPL	90,146	…
	1690	2022-12-31	AAPL	117,154	…
	⋮	⋮	⋮	⋮	⋱
	6066	2022-12-31	IBM	16,690	…
	⋮	⋮	⋮	⋮	⋱

IBES Consensuses data								
Variable	**Company identifier**	**Prediction date**	**Forecast horizon**	**Estimated measure**	**Company ticker**	**Fcst. quarter end date**	**Estimate**	**…**
Mnemonic	*TICKER*	*STATPERS*	*FPI*	*MEASURE*	*OFTIC*	*FPEDATS*	*MEANEST*	…
Instance	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱
	AAPL	2022-11-17	6 *(Q1)*	SAL	AAPL	2022-12-31	125,688.70	…
	AAPL	2022-11-17	7 *(Q2)*	SAL	AAPL	2023-03-31	97,700.19	…
	AAPL	2022-11-17	8 *(Q3)*	SAL	AAPL	2023-06-30	87,304.60	…
	AAPL	2022-11-17	9 *(Q4)*	SAL	AAPL	2023-09-30	94,984.77	…
	AAPL	2022-12-15	6 *(Q1)*	SAL	AAPL	2022-12-31	123,109.48	…
	AAPL	2022-12-15	7 *(Q2)*	SAL	AAPL	2023-03-31	98,239.74	…
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱
	AAPL	2022-12-15	6 *(Q1)*	NET	AAPL	2022-12-31	31,655.86	…
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱
	IBM	2022-12-15	6 *(Q1)*	SAL	IBM	2022-12-31	16,365.33	…
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

Table 9 shows in gray the variables constituting the primary keys of the respective databases. Forecast horizon (FPI) values correspond to the number of quarters in the future for which the prediction is made (Dai, 2020)—i.e., 6: next quarter; 7: quarter 2; 8: quarter 3; 9: quarter 4. See Figure o for the chronological representation.

This implies that the IBES consensuses data structure must be reshaped to fit with the Compustat architecture. To proceed as such, the IBES consensuses data table is pivoted so that values in specific columns become new multi-level column headers. In our case, consensuses estimates values are stored in newly created columns, which result from the outer product of the forecast horizon and the estimated measure[10]. This means that we obtain columns for each horizon and measure combination (see Table 10)—e.g.: quarter 1 & Sales (6 & SAL); quarter 1 & Net Income (6 & SAL); quarter 2 & Sales (7 & SAL); etc.

[10] Note that forecast quarter results announcement dates and forecast quarter end dates get also stored in newly created columns, with each respective date categories (i.e., announcement and end dates) having one column for each horizon.

Table 10: Processed IBES Consensuses Data — After Reshaping

IBES Consensuses data												
Variable		Company identifier	Prediction date	Company ticker	Fcst. quarter end date			Estimate				...
	Forecast horizon	/	/	/	6 *(Q1)*	7 *(Q2)*	...	6 *(Q1)*	6 *(Q1)*	6 *(Q1)*	...	⋱
	Estimated measure	/	/	/	/	/	...	SAL	NET	...	...	⋱
Instance		⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋱	⋱	⋱
		AAPL	2022-11-17	AAPL	x	x	...	x	x	...	...	...
		AAPL	2022-12-15	AAPL	x	x	...	x	x	...	...	...
		⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋱	⋱	⋱
		AAPL	2022-12-15	AAPL	x	x	...	x	x	...	...	...
		⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋱	⋱	⋱
		IBM	2022-12-15	IBM	x	x	...	x	x	...	...	...
		⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋱	⋱	⋱

Table 10 shows in gray the variables constituting the revised primary key of the IBES consensus database after undergoing reshaping. The primary key of the IBES consensus database is currently formed by a company-month combination. To align with the architecture of the Compustat database, it will be transformed into a company-quarter combination through the filtering process described hereafter.

Moreover, consensus data is available on a monthly basis (i.e., when the calculated summary consensuses statistics were finalized), but predictions have quarterly time steps—as shown in the column *Forecast quarter end date* in Table 9. This means several forecasts are made for (i) a given forecast horizon and (ii) a given forecast quarter end date (see Figure o). As one consensus data point is available every month, this means there are usually three redundant forecasts—as there are three months per quarter.

Figure o: Illustration of the Temporal Relationship between Prediction Dates, Forecast Quarter End Dates, Forecast Quarter Results Publication Dates and the Forecast Horizon

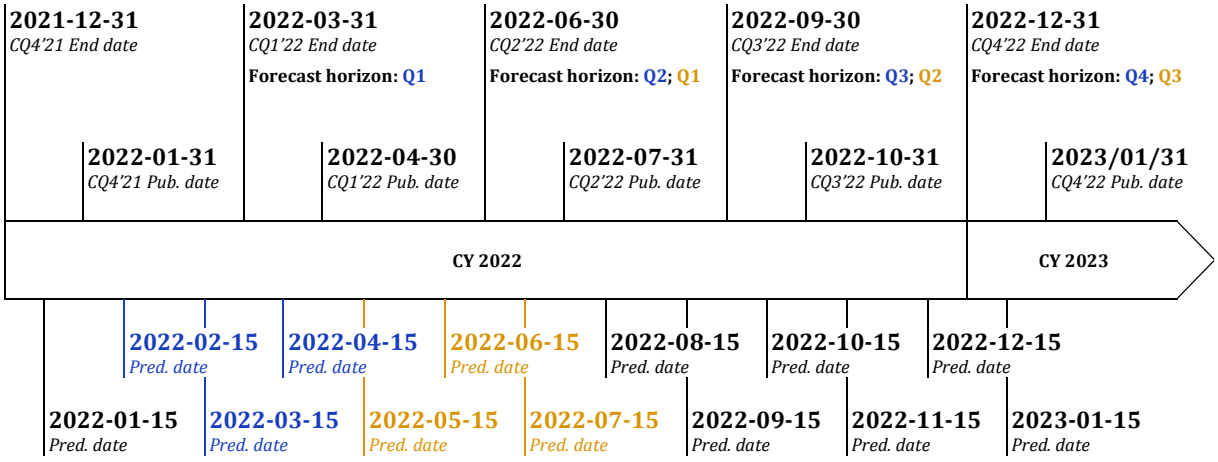


Figure o illustrates the chronological relationship between prediction dates, forecast quarter end dates, forecast quarter results publication dates and the forecast horizon for each prediction date for the year 2022. For example, the Q1 prediction corresponds to the earliest quarter whose results have not been published yet. Thus, forecasts at a one-quarter horizon for May 15th, Jun. 15th, and Jul. 15th (yellow above) are made for 2022's second calendar quarter (end date: Jun. 30th), as these dates fall between the 2022's first and second calendar quarter results publication dates—i.e.,

Apr. 30th and Jul. 31st. The same holds for greater horizons, implying that forecasts for every horizon made on these three dates are necessarily redundant.

We figured out that selecting the last of them (e.g., usually 8-9 months before the Q4 forecast quarter, see example in blue in Figure o), and forward filling using past values makes up for a large portion of missing values. The reason is that we have on average three predictions for (i) a given forecast horizon and (ii) a given forecast quarter end date. Of course, this implies that the selected prediction date takes place less than a year before the actual forecast quarter end date—thus reducing the forecast horizon—, but such a choice was made to ensure a greater data availability while avoiding look-ahead bias.

D.3. Compustat Preprocessing

This piece of code deals with *Compustat* Fundamental data specific preprocessing. Four main processes were carried out: (i) the treatment of year-to-date items (YTD); (ii) the computation of a *Free-Cash-Flow* (FCF) variable; (iii) the removal of duplicated rows induced by the change in fiscal year / fiscal quarter; and (iv) the transformation of certain items from quarterly figures into Trailing Twelve Months (TTM) figures.

Some cash-flow statements item feature YTD figures, meaning that metrics are computed from the beginning of the current fiscal year up to the fiscal quarter. However, so that models can assess differences in variables between different quarters of a given company, the accumulating effect must be removed. In other words, these items have to be converted from YTD figures to quarterly figures—i.e., the value of a cash-flow item on a given quarter should only reflect the value of that specific quarter.

In practical terms, this means that, for a given fiscal year, YTD data from the previous fiscal quarter should be deducted for every quarter, apart from the first fiscal quarter, as the quarterly figure is equivalent to the YTD figure.

Using these treated cash-flow items, we computed an approximation of a *Free-Cash-Flow* item, defined as the difference between the *Operating Cash Flow* and the *Capital Expenditure* (CapEx), using an approach proposed by Chauhan et al. (2020). Proceeding as such keeps a single metric that evaluates a company's ability to generate cash returns for its investors.

Next, an important task is to address duplicated rows that stem from a change in fiscal year-end. Let us consider a scenario where a company decides in December 2022 to shift

its fiscal year-end from December 31st to September 30th. Previously, the fiscal quarters aligned with the calendar quarters, but now, fiscal quarter 4 (FQ4) will correspond to calendar quarter 3 (CQ3). To ensure the chronological continuity of fiscal quarters, Compustat duplicates the data for CQ4'22 and associates it with both FQ4'22 and FQ1'23[11]. Since we are primarily interested in the calendar quarter data, only one record is retained for each calendar quarter. As a result, the choice of a specific fiscal year-end by a company essentially becomes irrelevant.

Lastly, to account for seasonal fluctuations, it is necessary to convert cash-flow (including the derived Free-Cash-Flow) and income statement items from quarterly figures to trailing twelve months (TTM) figures. This essentially means that, for each company-quarter combination and for each of these items, we sum-up the values of the past four consecutive quarters (additional detail is provided in 2.2.1).

Note that we do not check whether quarters are consecutive at this stage. This means that, if there is a missing quarter, an incorrect TTM figure will be computed. However, when creating lagged variables (as described in section D.4), the misalignment caused by a missing quarter[12] will be addressed. This means that the affected rows will eventually be dropped.

D.4. Merged Preprocessing

This script aims at performing preprocessing tasks on the complete dataset. Three steps can be distinguished here, namely: (i) linking the various retrieved datasets; (ii) performing an Exploratory Data Analysis (EDA); and (iii) performing remaining data cleaning and integration tasks.

The first step consist in merging four different datasets (i), namely: *preprocessed Compustat Fundamentals* (see A.4 & D.3); *Compustat Market Securities* (see A.5); *preprocessed IBES Consensuses Forecast* (see A.6 & D.2); and *Compustat-IBES linking table* (see A.3).

---

[11] If the fiscal year-end is instead extended further into the future, it does not pose any issues. Indeed, in this case, some calendar quarters are not linked to any fiscal quarters, but no deletion or duplication of quarterly financial data takes place.

[12] A misalignment could also be due to duplicated quarters, but this situation has been explicitly handled earlier.

Two points of interest are worth noting when it comes to linking security data with the corresponding fundamental data for a given company. First, there is a mismatch between the chronological basis on which fundamental and market data are available: merging the two necessitated aggregating monthly market data into its corresponding quarterly equivalent. In practice, we selected the securities data corresponding to the quarter end date. Secondly, as several companies can feature several stocks, a selection had to be made between these securities. The choice here was to select the most traded security, pursuant to what IBES is currently doing[13].

Subsequent merging with IBES Consensuses data required first to obtain the corresponding IBES key (*'ibtic'*), which was retrieved *via* the Compustat-IBES linking table. However, to link each Compustat record to an IBES record requires more than simply a company key, as these two need to be chronologically associated as well—i.e., the fundamentals for a given company-quarter combination must be associated with forecast consensuses for the same company-quarter combination. Therefore, the two datasets are joined on the IBES company key (*'ibtic'*) and—respectively for the Compustat-based table and the IBES table—the quarter end date / Q4 forecast quarter end date (Table 9 and Table 10 can help understanding the date keys on which the two datasets are joined). This essentially means that the two datasets are merged so that the actual fundamental quarter end date corresponds to the Q4 forecast quarter end date (see Figure p). In other words, *forecasted* fundamental values (i.e., IBES data) are associated with *actual* fundamental values (i.e., Compustat data) such that these *forecasts* are made approximately 8-9 months before the quarter end date of the *actual* fundamentals. The idea is that some of the *actual* fundamentals will eventually serve as target variables, and thus that consensuses *forecasts* serve as estimates for the *actuals*, 8-9 months prior.

[13] Thomson Reuters uses the class of shares that is generally available to investors.

Figure p: Illustration of the Temporal Relationship between Prediction Dates, Consensuses Forecasts Quarter End Dates and Actuals Quarter End Dates

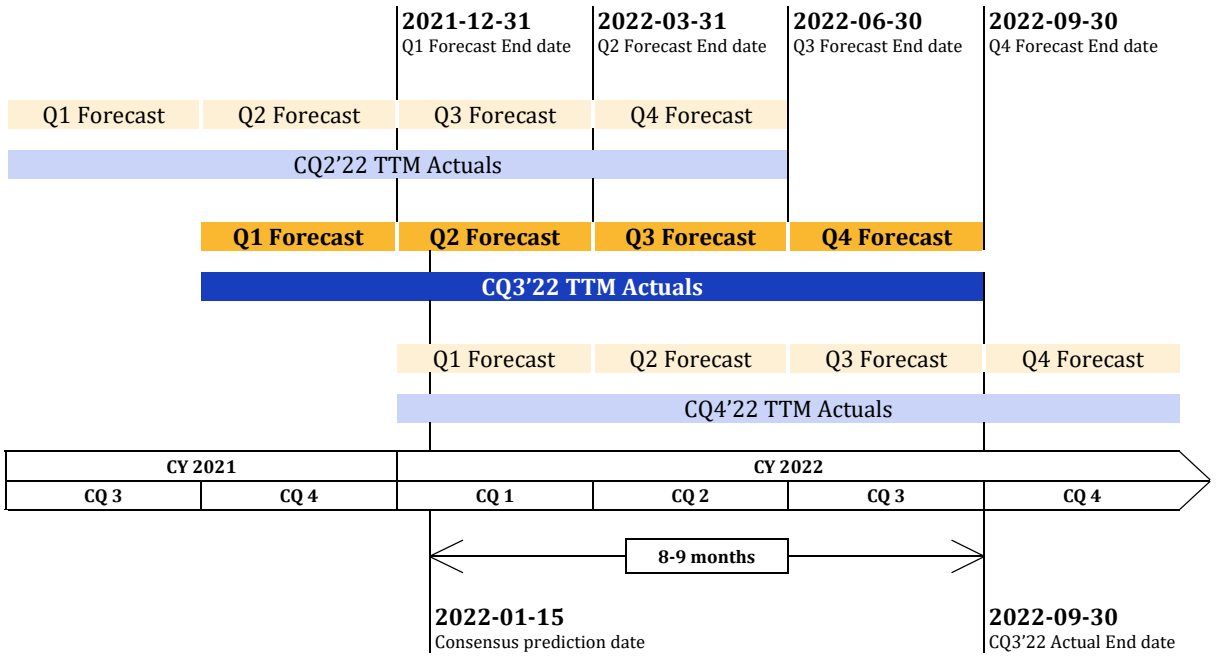


Figure p illustrates the chronological relationship between prediction dates, consensuses forecasts quarter end dates and actuals quarter end dates for a consensus prediction made on Jan. 15th, 2022. By pairing databases so that the actual fundamental quarter end date corresponds to the Q4 forecast quarter end date, consensuses predictions tend to be carried out on average 8-9 months before the end date of the TTM actual fundamentals—this duration varies depending on the publishing date of the Q1 Forecasted actuals (see Figure o).

The next logical step is to sum up Q1, Q2, Q3 and Q4 forecasts so that we obtain trailing twelve months (TTM) figures instead of quarterly figures. This process is essentially the same as the one described for Compustat cash-flow and income statements items (refer to D.3), the difference being that the sums are calculated column-wise rather than row-wise. In other words, for each company-quarter record and for each forecasted item (e.g., Sales, Net Income…), Q1 to Q4 forecasts are summed in order to obtain a moving 4-quarters prediction (referring to Figure p can help visualizing the issue at hand).

The second step (ii) comprises of an Exploratory Data Analysis. No noticeable observation deserves to be commented here as we are still working with level (non-scaled) values. This means a great disparity in fundamental values can be observed here, given the differences in size between companies. Still, a particular emphasis will be placed on missing values later on, as figures in 2.3.3 show.

The third step (iii) ensures that the data is ready to be used in the modelling phase. Six elements are worth to be mentioned: computation of the relative change of variables;

treatment of outliers and infinite values; removal of variables with too many missing values; filtering of rows with missing earning consensus forecasts; imputation of remaining missing values; creation of lagged variables. This appendix does not detail further this process, as part 2 about data preprocessing already explain extensively the followed procedure.

D.5. Modeling

This script aims at performing modelling tasks on the complete dataset obtained previously.

First, in-sample and out-of-sample data get split based on the specified date. The same process takes place for explanatory and target variables, based on the role these variables play. Then, analysts' consensus forecasts get scored against actual values (baseline scenario), and OLS model metrics are derived for both scenarios to serve as an additional point of comparison.

Finally, for both scenarios, a hyperparameter search grid is defined, and hyperparameters get optimized by randomly selecting a set of 500 combination candidates. To reduce the required processing power, this process is carried out using successive halving (see 3.3.2 for explanations) and an expanding window walk-forward validation scheme (see 1.3.4). The model with optimized hyperparameters is then fitted on the entire training set belonging to the *in-sample* period. It is then scored on the test set belonging to the *out-of-sample* period.

Additionally, learning curves on the training set were plotted, in order to visualize how the model's performance changed as the amount of training data increased (see 4.2).

Abstract:

Value investing is an investment approach that involves selecting undervalued assets based on fundamental analysis, with the expectation that their true intrinsic value will be recognized by the market over time, leading to abnormal long-term returns. We apply machine learning techniques, specifically histogram-based gradient boosting regression trees, to predict one-year net income growth using a trailing three-year window of historical company fundamentals. The premise is that an accurate earning prediction leads to the selection of better-performing stocks, enabling the development of a long-holding, systematic value investing strategy. We assess the predictive performance of the approach with that of analysts' consensus forecasts alone and a similar approach combining analysts' consensus forecasts and company fundamentals as model input data. With respective R^2 values of 0.045 and 0.232, results do not show that making use of fundamental data alone can offer comparable results to analysts' forecasts. Moreover, while the combined approach shows marginal improvements over analysts' consensus forecasts, the potential of combining consensus forecasts with historical fundamentals remains underutilized with the chosen configuration. These findings suggest future research should make use of more sophisticated models incorporating historical fundamentals alongside analysts' consensus.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Louvain School of Management

Place des Doyens, 1 bte L2.01.01, 1348 Louvain-la-Neuve
Boulevard Emile Devreux 6, 6000 Charleroi, Belgique
Chaussée de Binche 151, 7000 Mons, Belgique

www.uclouvain.be/lsm