Louvain School of Management

Forecasting Revenue Surprises using Time Series Forecasting models in the context of an active investment strategy

Author(s): Louie Marrazza
Supervisor(s): Corentin Vande Kerckhove
Academic year 2022-2023
Dissertation for the master of
Master (120) in Business Engineering
Daytime schedule

Abstract

In the context of active investing, many traders have attempted to forecast stock values using machine learning techniques. However, it has always been seen as a hard and challenging task. In this thesis, we aim to predict revenue surprises prior to quarterly revenue publications using time series forecasting models based on brokers' estimates and macro-economic indicators. The main objective of these forecasts is to capitalize on the direct impact of revenue surprises on stock prices through an active investment strategy. Studies cited and exploited in this research showcase the significance of forecasting surprises on companies' fundamentals, rather than focusing directly on these fundamentals. Our results, however, do not conclusively establish our model's superiority over random investment decisions. In summary, the results of this thesis suggest that enhancing the quality of datasets used in predictive models, along with leveraging more powerful forecasting models, could improve the quality of our predictions.

Acknowledgements

I would like to express my gratitude to Professor Vande Kerckhove for his invaluable assistance in guiding me through the process of writing my thesis.

My thanks also go to my family and friends for their constant support and encouragements.

I would also like to thank the professors from UCLouvain that have provided me with the educational tools that facilitated my personal growth and enabled me to compose this thesis.

Table of contents

List of Tables

List of figures

List of Appendix

Introduction

The aim of this study is to utilize time series forecasting models and brokers' projections of fundamental values to anticipate surprises in companies' quarterly financial statements. The focus point will be predicting revenue surprises, which entail the difference between a company's actual reported quarterly revenue and the revenue projected by brokers. Given that revenue surprises directly influence stock values, the purpose is to capitalize on these anticipated surprises to actively engage in stock market trading, thereby taking advantage from the market's reactions to these revenue deviations.

Subsequently, the primary goal involves training our model using historical data using brokers' estimates of all kind and macroeconomic indicators available just prior to the quarterly financial report's release.

To facilitate a performance comparison and determine the model's statistical significance, we will establish a basic random model characterized by volatility equivalent to the historical volatility of revenue surprises. This will be a main point to assess some conclusions as if the model is outperforming our random model, that means that we could affirm that our model is significantly efficient.

The first focus of our work was the collection of data. Subsequently, the second section of this study explore the importation, selection, and various transformations applied to the data. This segment elucidates the process of importing data from Refinitiv's Excel plugin and outlines the steps taken to convert it into a format conducive to analysis. The subsequent discussion encompasses the process of selecting pertinent variables.

Moving on to the following chapter, the predictive methodologies are introduced. This includes a detailed description of the ARIMAX time series forecasting model and a clear definition of the concept of stationarity. This chapter concludes with an exploration of the model calibration process, elucidating parameter selection and the presentation of diverse performance metrics.

The concluding two chapters of this research provide an intricate analysis of results. This entails a comparative assessment of our outcomes against those of the implemented naive random model. Subsequently, the research concludes with a final synthesis of findings.

1 State-of-the-art

1.1 Stock market

The origins of the stock market can be traced back to the 17th century, when the first formal stock exchanges emerged in Europe. One of the most famous examples is the Amsterdam Stock Exchange, established in 1602, which facilitated trading in shares of the Dutch East India Company. This marked the birth of modern-day stock markets, as investors sought opportunities to participate in the company's profits and growth.

Over the centuries, stock markets expanded and flourished, becoming key pillars of capitalism. One notable milestone in the development of the stock market occurred in 1792 when the New York Stock Exchange (NYSE) was founded. Operating on Wall Street in New York City, the NYSE quickly grew to become one of the world's largest and most influential exchanges, setting the stage for the rise of American capitalism.

It serves as a vital mechanism for companies to raise capital by issuing shares to the public, allowing them to fund expansion, research and development, and other ventures. In return, investors gain the opportunity to profit from the success of these companies through capital appreciation and dividends.

In recent times, the stock market has witnessed unprecedented technological advancements such as trading platforms or easy access to historical data that have revolutionized trading. It allowed investors to participate more actively in the stock market, democratizing access to investment opportunities.

To unlock profitable opportunities, investors harness a variety of forecasting techniques, aiming to predict market movements and leverage that knowledge to their advantage. However, it is important to acknowledge that existing forecasting techniques have their limitations in directly predicting the stock market. This insight is discussed in 1) and 2) by Umstead (1977) and Agrawal et al. (2013)

The dynamic and competitive nature of the market presents obstacles that hinder the guaranteed long-term success of these models. The stock market is subject to constant fluctuations, influenced by a myriad of factors such as economic conditions, geopolitical events, and investor sentiments. These intricate and ever-changing dynamics make it challenging for forecasting models to provide consistently satisfactory results in predicting stock market movements.

1.2 Stock fundamentals

Stock fundamentals refer to key metrics or factors that provide insights into the financial health and performance of a company 3). These are the qualitative and quantitative factors that give its intrinsic or real value to the company. For example, these fundamentals can be cash flow, return on assets, EBIT or any other factor that give value to a company.

In 2004, Gentry et al. 4) made an experiment where they will buy underpriced REIT stocks and short-selling overpriced REIT stocks based on their net asset values (NAV). They concluded that implementing this strategy can lead to monthly excess returns from 0.9% to 1.8% and therefore that looking at the fundamental value of the stock instead of only the stock price, can present more profitable strategies.

Moreover, Albert and Lipton 5) made research on the application of deep neural networks for forecasting company fundamentals in quantitative investing strategies. They proposed a strategy that construct portfolios based on predicted future fundamentals. The study showed that the use of deep neural networks on these fundamentals can make interesting predictions that achieve a significant improvement in the mean squared error (MSE).

Focusing on stock fundamentals, such as cash flow, return on assets, and EBIT, provides valuable insights into a company's financial health and intrinsic value. By prioritizing fundamentals over stock prices, investors can make more informed and potentially profitable investment decisions.

1.3 The concept of surprise

As expressed by Jegadeesh and Livnat 6), a surprise refers to the difference between actual revenue or earnings reported by a company and the market's expectation or forecast for those metrics. A positive surprise occurs when the reported revenue or earnings exceed expectations, while a negative surprise occurs when they fall short.

In their lecture, they use standardized unexpected earnings (SUE) as their measure of earnings surprise :

$$SUE_{i,t} = \frac{Q_{i,t} - E(Q_{i,t})}{\sigma_{i,t}}$$

Where $Q_{i,t}$ is the quarterly earnings per share, $E(Q_{i,t})$ is the expected quarterly earnings per share prior to earnings announcement, and $\sigma_{i,t}$ is the standard deviation of quarterly earnings growth.

It is observed that stock price reactions on earnings announcement dates are significantly related to contemporaneous and past revenue surprises. Stocks with large revenue surprises tend to experience significant abnormal returns in the post-announcement period, even after controlling for earnings surprises.

In 2006, Livnat and Jegadeesh 7) conducted a study where they examine the impact of sales surprises on the drift in security returns following the disclosure of earnings. The study emphasizes the economic significance of the findings by presenting annual returns on the combined earnings and sales hedge portfolio. The results show significant mean returns of approximately 19% and 27% annually for different samples. The earnings and sales hedge portfolio outperform the earnings only hedge portfolio, with the former yielding higher returns. This study highlights the importance of incorporating sales surprises into the analysis of earnings drift in security returns. The magnitude of the drift is found to be larger when the sales surprise confirms the earnings surprise, indicating the persistence of revenues and the heterogeneity of expenses as contributing factors.

Moreover, a study leaded by Da et al. 8) in 2011 on the usefulness and efficiency of surprise prediction in investment strategy using search volume index (SVI) data concluded that SVIs were promising indicators for revenue surprises and announcement returns and that identifying these surprising situations could become a useful tool for forecasting fundamental stock values.

In general, there is a strong market reaction to earnings surprises but also to revenue surprises even after controlling for earnings surprises 9) as proved by Henderson and Marks with a study on market reaction to earnings and revenue surprises in 2013. In addition, another study conducted in 2004 by Rees et al. outlined that the market response to meeting or beating earnings forecasts is accentuated when revenue forecasts are also met, while not meeting revenue forecasts generally results in a significantly negative market penalty 10).

This is why taking an interest in surprise prediction or analyst error rather than simply fundamental values or stock prices could be an interesting direction to explore for the rest of this work.

1.4 Time-series forecasting models

Time series forecasting models are statistical models used to make predictions about future values of a time-dependent dataset based on its historical patterns. These models take into account the sequential nature of the data, where the values are observed at regular intervals over time. There are several time-series forecasting models, each has specific features.

The autoregressive (AR) model uses the previous values of the time series to predict future values. It assumes that future values are linearly dependent on past values, with the relationship determined by the model's order. An AR(p) model, an autoregressive model of order p can be written as:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \epsilon_t$$

The moving average (MA) model calculates the average of past observations within a sliding window and uses it to forecast future values. It assumes that future values will exhibit similar fluctuations as the past values. MA(q) can be written as:

$$y_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q}$$

The most used time series forecasting model is the autoregressive integrated moving average (ARIMA) model. ARIMA models are popular due to their ability to capture a wide range of time series patterns, including trends and seasonality. They can handle both stationary and non-stationary data by incorporating differencing to remove trends or seasonality.

In 2014, a study 11) conducted by Mondal et al. was focused on forecasting stock prices of Indian stocks from different sectors using ARIMA models. The study demonstrated the effectiveness of the ARIMA model in forecasting stock prices. However, it also observed that there is room for improvement.

Another study 12) conducted in 2014 by Rotela Junior et al. where the focus was on forecasting the Bovespa Index (Index of shares of the Sao Paolo Stock Exchange) also obtained interesting results for time-series forecasting. Indeed, the AR(1) model obtained a MAPE (mean absolute error percentage) lower than all other models used for comparison.

1.5 Research question

This research has served as the foundation for developing the methodology employed throughout this work. The approach adopted in this study involves utilizing time series forecasting models to predict surprises in the fundamental values of a company. More specifically, the focus will be on quantifying the differences between the predicted values and the values forecasted by analysts prior to each quarter. By employing these forecasting models, we aim to gain insights into the accuracy of predictions and assess the level of surprise associated with the fundamental values.

The specific objective of this research will involve the application of time series forecasting models using broker's quarterly fundamental estimates, including revenue or EBIT. These forecasts will be utilized to capitalize on market responses to revenue surprises, enabling active trading in the stocks of relevant companies. This research addresses a void in the literature by focusing specifically on forecasting this surprise with time-series forecasting models. The focus of this research paper will be the following: “Forecasting Revenue Surprises using Time Series Forecasting models in the context of an active investment strategy.”

2 Data Collection

2.1 Data importation

In the context of this research, our objective is to predict surprises on stock fundamentals, with revenue as main focus, in order to make proactive investment decisions. To achieve this, the first step is to import companies' historical data such as stock fundamentals but also to import the brokers' estimates for these fundamentals. This will enable us to run our tests and help us to improve our results step by step.

All the companies' historical data come from Refinitiv Eikon. We opt for the Excel add-on integrated with the Screener application to efficiently import and manage large datasets. By employing advanced analytical methods and utilizing an array of financial metrics, we aim to develop a robust predictive model that can improve investment strategies and contribute to the broader understanding of financial market dynamics.

To import data from Refinitiv Eikon to my Excel file, we used the following process.
With access granted from the LFIN research center (Louvain Finance), a connection to the Refinitiv application has been established.

With this connection between Excel and Refinitiv Eikon, we navigated to the Screener App to initiate the data retrieval process. To ensure comprehensive coverage, careful selections were made, opting for both public and private companies as part of the universe. This approach aimed to capture any potentially interesting company for analysis, avoiding any omissions.

Further refining the dataset, we applied filters to narrow down the focus. As working with data-rich markets was primordial, we selected "United States of America" as the "Country of Headquarters." The American market offered a wealth of data, particularly in terms of brokers' estimations, which was crucial for our research objectives.

To manage the dataset's size effectively, we applied a "Company Market Capitalization" filter, specifying a threshold of 150,000 (in millions of dollars). This filtering criterion resulted in a dataset comprising 48 companies, striking a balance between having enough data for meaningful testing and comparisons, and avoiding excessive complexity.

With the dataset curated to my requirements, we proceeded to select the specific features or variables we intended to analyze (Revenue surprise, Revenue – actuals, Revenue – mean, EBIT – mean, Net debt – mean, EBITDA - mean) and imported the 42 last fiscal quarters of data starting from FQ0 (which means last fiscal quarter). The variables followed by "mean" means that it is the mean of all broker's estimates, so " Revenue – mean" is the mean of all broker's revenue estimations for the concerned quarter.

Two macro-economic variables have also been imported, the unemployment rate and the nominal broad U.S. Dollar Index (USD index) from the "Federal Reserve Economic Data (FRED) St. Louis Fed" directly using the "quantmod" package in R. Table 1 gives the detail of our imported data.

Variable imported	Period	Time unit	# of periods	# of companies	Source
Revenue - actuals	2013-2023	Quarters	42	48	Refinitiv Eikon
Revenue - mean	2013-2023	Quarters	42	48	Refinitiv Eikon
Revenu - actual surprise	2013-2023	Quarters	42	48	Refinitiv Eikon
EBIT - mean	2013-2023	Quarters	42	48	Refinitiv Eikon
Net debt - mean	2013-2023	Quarters	42	48	Refinitiv Eikon
EBITDA - mean	2013-2023	Quarters	42	48	Refinitiv Eikon
Unemployment rate	2013-2023	Quarters	42	NA	FRED St. Louis Fed
USD index	2013-2023	Quarters	42	NA	FRED St. Louis Fed

Table 1 : Imported data summary table.

After curating the dataset to meet our research objectives, we imported it into Excel and therefore exported it into a CSV file. Once the data was in this format, we could import it into R and manipulate the data for further analysis.

2.2 Creation of time series

The primary goal of this research is to predict revenue surprises using time series forecasting models. To achieve this objective, it is essential to organize the imported data in a structured manner. Specifically, we consolidate the data for each company into a single data frame, where each column represents a distinct variable (e.g., revenue surprise, EBIT - mean, etc.), and each row corresponds to a specific date.

The initial raw data acquired from Refinitiv is structured in a manner where each company's information is organized as rows, and different attributes or features are represented as columns, forming a tabular structure. To facilitate our analysis, the data was thoughtfully imported in a way that ensures the corresponding time data aligns with each feature.

Revenu surprise	FQ0	FQ-1	FQ-2	FQ-3
Company 1	5.90%	-7.18%	-4.61%	-2.02%
Company 2	0.29%	-8.21%	-0.76%	2.88%
Company 3	-2.06%	4.72%	0.64%	-2.15%
Company 4	-3.06%	-9.38%	-2.85%	-4.76%
Company 5	-3.95%	6.83%	-1.65%	-10.88%
Company 6	1.16%	-0.64%	5.19%	-4.61%

Table 2 : Example of raw imported data from Refinitiv reduced to 4 periods and 6 companies.

To transform our data into the desired format, we will utilize a list to categorize and assign the various characteristics we intend to analyze. This list will then serve as input for a function named "companies_list_fct". The function's output will yield a list comprising 48 elements, with each element corresponding to a different company. Within each element, every variable linked to this particular company will be sorted together. This organized structure will streamline our analysis, enabling us to efficiently explore and extract valuable insights from the dataset for each individual company. The structure of each companies' dataset will be as presented in Table 3.

Company 3	Revenue surprise	EBIT - mean	Unemployment rate
FQ-5	11.01%	4217.55	2.96%
FQ-4	3.02%	5173.42	2.60%
FQ-3	-2.15%	7619.73	3.08%
FQ-2	0.64%	7136.48	2.67%
FQ-1	4.72%	6454.11	3.14%
FQ-0	-2.06%	5412.40	3.47%

Table 3 : Example of sorted per company data reduced to 3 variables and 6 periods.

This organized format allows us to transform the dataset into time series data, facilitating the application of time-dependent analytical techniques. By leveraging this time series data, we aim to uncover meaningful patterns and trends that will contribute to our revenue prediction models.

2.3 Variable selection

As explained in the first part of this research, the objective is to predict the surprise on fundamental values. To obtain a reliable prediction the data we will use needs to be available before the publication of the quarterly results from the company.

In this case, we will use historical data of revenue surprise "Revenue – Actual Surprise" available on Refinitiv which corresponds to the difference between the actual value of the revenue and the broker's estimation mean. This will be our dependent variable, which means that this is the variable we are going to try to predict or explain in the forecasting model.

Our model will be composed with a various range of external independent variables. To start, our model will be composed by the broker's estimation of the revenue itself, the EBIT and EBITDA. These market estimations are valuable because they represent the consensus market expectations regarding a company's financial performance. By incorporating these estimations, we are considering the collective insights and forecasts from financial experts and analysts, which can serve as a benchmark for evaluating actual revenue outcomes.

On the other hand, we will also use macro-economic variables, and to remain realistic we will use the information that is available just before the announcement of the company's quarter result. Macroeconomic indicators provide valuable insights into the overall economic conditions in which companies operate. The unemployment rate serves as a gauge of labor market conditions, while the Broad USD Index reflects the performance of the U.S. dollar against a basket of major currencies. Changes in these indicators can impact consumer's spendings and business investments, the international trade, all of which can influence a company's revenue generation.

The sample will cover historical data of the 48 companies with the largest market capitalization in the United States market from 2013 to 2023 on a quarterly basis. Coordinating the dates of the different variables is an essential element given that the models explored in this work are time series models.

3 Prediction process

In this section, the forecasting process will be presented. After the data collection part, we will try to predict future values for our dependent variable.

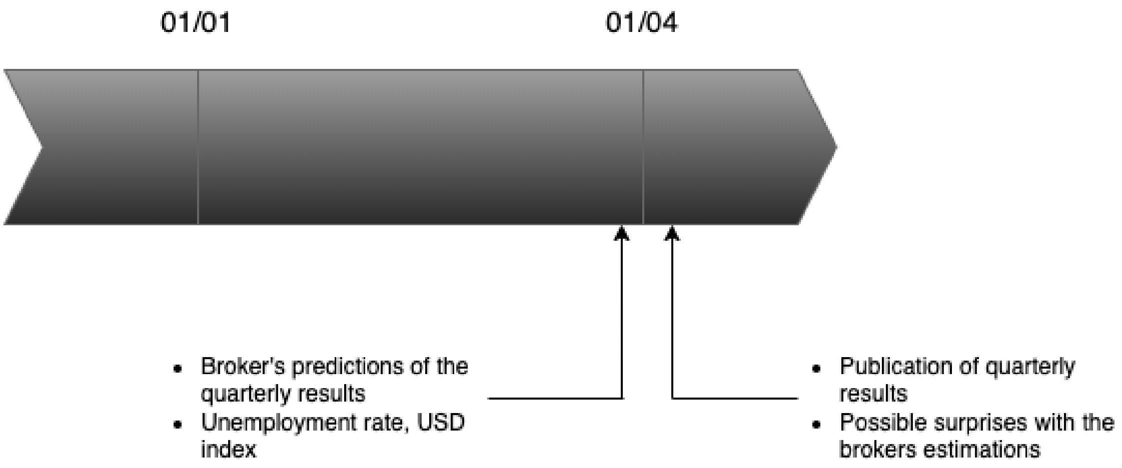

Figure 1 : Timeline of the prediction process.

It is important to clarify the time context, as it is presented in Figure 1. We are at the end of the quarter and have the following information: the average of brokers' estimates for company fundamentals and macroeconomic data at that time. Using this information, we are trying to predict the sign of the surprise on the company's earnings so that we can go long in the event of a positive surprise or short in the event of a negative surprise, given that this surprise will more than likely have a direct impact on the share price linked to the company.

3.1 Models

Time series forecasting models are vital for our study, and among them, the ARIMA (Auto Regressive Integrated Moving Average) model holds a prominent position. Its widespread use can be attributed to its simplicity, flexibility, and impressive ability to model and predict time series data. The ARIMA model is specifically designed to analyze individual time series independently, making it a preferred choice for univariate forecasting tasks that don't involve external factors or other time series.

One of the key strengths of the ARIMA model lies in its capability to capture and forecast trends in time series data. Additionally, it excels at incorporating seasonal components, making it highly adaptable to data with recurring patterns, be it monthly, quarterly, or yearly. This feature is particularly advantageous for our study as it allows us to consider and account for seasonal fluctuations in our time series data.

Furthermore, ARIMA is remarkably well-suited for short to medium-term forecasting, which aligns perfectly with our final objective of developing an active investment strategy. The model can reliably provide accurate predictions for a few periods ahead, enabling us to make well-informed decisions for our investment approach.

Another appealing characteristic of ARIMA is its relative robustness even when dealing with limited historical data points. Given that we are working with quarterly data, which naturally restricts the number of available data points, this resilience becomes an invaluable asset for ensuring reliable and meaningful forecasts.

However, the objective of this work is to forecast the sign of the surprise on revenue of a company. In this case, the ability to capture and forecast trends is not especially the most important but the strong use of historical data points in multiple ways will be very useful in our case. However, this surprise could not be predicted correctly without any external factors.

Indeed, a forecast of this surprise is possible only with external factors. Moreover, we have to be careful to avoid the "look-ahead bias" situation. This bias occurs when one unknowingly or intentionally includes future information in the historical dataset used for model building or parameter estimation. As a result, the forecasting model appears to perform better than it would in a real-world scenario where future information is not available.

The ARIMA model has been defined to introduce the model we will use in this research, which is an extension of it, namely the ARIMAX. This model allows multivariate forecasting by incorporating external variables but has also all the interesting ARIMA model characteristics discussed in this section. The ARIMAX model is driven by the following formula: (ARIMAX(p,1,q) example)

$$\Delta Y_t = c + \sum_{k=1}^{m} \beta_k X_{k,t} + \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \epsilon_t$$

Where :

- Δ represent the first difference (needed because the dependent variable needed 1 differencing to be stationary)
- Y_t represent the value the dependent variable at time t
- c is the constant term
- p is the order of the autoregressive component
- ϕ_i are the autoregressive coefficients
- q is the order of the moving average component
- θ_j are the moving average coefficients
- m is the number of exogenous variables
- β_k are the coefficients corresponding to each exogenous variable $X_{k,t}$
- ϵ_t is the error term at time t

Compared to the ARIMA model formula which is :

$$\Delta Y_t = c + \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \epsilon_t$$

The only difference from a univariate ARIMA model will therefore be the integration of exogenous variables into our model. It can be economic indicators, weather data, marketing campaigns, or other relevant data that can influence the targeted variable behavior. In this case, we will use broker’s estimation data (EBIT, EBITDA and revenue) and also use macroeconomic indicators (unemployment rate and USD broad index) to see how they can influence our model.

The rationale for incorporating macroeconomic factors into our model stems from the recognition that surprises on a company’s revenue can be influenced by real-world economic conditions and the overall state of the country's economy. By including these relevant macroeconomic indicators, we aim to capture the broader economic context, which can significantly impact the accuracy and comprehensiveness of our forecasting model.

3.2 Stationarity

In the context of time series analysis, stationarity refers to a property of a time series where statistical properties remain constant over time. Specifically, a stationary time series exhibits constant mean, constant variance, and an autocovariance function that depends only on the time lag between observations, not on the absolute time at which the observations were made. In simpler terms, the data's behavior does not change over time and does not show any long-term trends, periodic patterns, or systematic changes in variability.

This property is essential because stationary time series are more predictable, making it easier to forecast future values using historical data and statistical models. Moreover, stationary data allows us to apply simpler statistical models because the statistical properties remain consistent over time. Non-stationary data requires more complex models to capture trends and seasonality. But more importantly, many time series models and statistical tests assume stationarity. Violating stationarity assumptions can lead to unreliable results and incorrect inferences.

In the context of our research, the dependent variable in an ARIMAX model must be stationary because ARIMAX models assume that the time series and its related components (such as trend and seasonality) exhibit stationarity. Stationarity implies that the statistical properties of the series, such as mean and variance, do not change over time. If the serie is not stationary, the assumptions underlying ARIMAX models are violated, which can lead to unreliable forecasts.

When the dependent variable is stationary, it allows us to use the ARMA (Auto Regressive Moving Average) components effectively, as these components assume stationarity to estimate the relationships between past observations and forecast future values accurately. It is essential to preprocess the dependent variable by applying differencing (integration) to make it stationary before fitting the model. Differencing helps remove trends and ensures that the ARIMAX model can capture the underlying relationships between the variables and produce reliable forecasts.

To ensure stationarity, the dependent variable of our model will be the input of a created function make_stationary(data), that will test stationarity with the augmented Dickey-Fuller test (ADF test) and will difference the time series until the ADF test indicates that our series is stationary. As output of this function, we will obtain a list with the differenced data and the number of differentiations that was needed.

3.3 Model calibration

Selecting the appropriate values for the parameters p, q, and other components in an ARIMAX model involves a process called model calibration. The goal is to find the best fitting ARIMAX model that adequately captures the underlying patterns and relationships in the data.

As explained in the previous part, the first step to correctly calibrate our model, is to ensure that the dependent variable is stationary. If our data series are not stationary, we apply differencing (integration) to make it stationary. This process involves determining the order of differencing (d) required to achieve stationarity.

Then, to determine p (AR) order and q (MA) order, we need to Analyze the autocorrelation function (ACF) and partial autocorrelation function (PACF) plots of the stationary dependent variable to gain insights into potential AR (Auto Regressive) and MA (Moving Average) orders.

- ACF shows the correlation between the series and its lagged values. Significant correlations at specific lags suggest potential MA terms (q).
- PACF shows the correlation between the series and its lagged values while controlling for intermediate lags. Significant correlations at specific lags suggest potential AR terms (p).

Based on these ACF and PACF plots, we can select tentative values for the AR (p) and MA (q) orders. Tentative values are typically the lags at which significant spikes are observed in the plots.

To rely on quantitative information for this selection process, we can see a model selection criterion, such as Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC), to compare different ARIMAX models with varying combinations of p, q, seasonal AR, seasonal MA, and exogenous variables. The model with the lowest AIC or BIC value is usually preferred as it balances model fit and complexity.

With the R function auto.arima(), by indicating our dependent variable and the exogenous variables, the model combination with the lowest AIC criterion will automatically be selected. Indeed, this function is reviewing automatically lots of combinations of p,q and exogenous variables and is selecting the one with the lowest information criteria.

3.4 Backtesting

Our time series data will be divided into a training set and a test set. This division enables us to train our model using historical data, forecast our variable with the trained model and then assess its performance on a test set of historical data. This approach assumes that if the model performs well on historical data, it should maintain a similar level of performance in new real-world scenarios. This concept is known as backtesting.

Within this framework, we will utilize two distinct approaches. The first approach involves a conventional train-test split, aimed at evaluating the effectiveness of our ARIMAX model. Data splitting will involve the creation of two sets – a training set covering the period from January 2013 to June 2020, and a test set spanning from July 2020 to June 2023.

Subsequently, our model will undergo training using the training set, and its performance will be evaluated using the test set.

Figure 2 : Classical train-test split timeline.

The second approach involves employing an expanding window backtesting method. This method employs three distinct iterations to train the model. During the initial iteration, the training set remains consistent with the traditional train-test split, spanning from January 2013 to June 2020. However, the test set is limited to only 4 quarters, extending from July 2020 to June 2021.

In the subsequent iteration, the training set expands by an additional 4 quarters (from January 2013 to June 2021), while the subsequent test set covers the following 4 quarters (July 2021 to June 2022). In the third and final iteration, the training set further expands by 4 quarters (from January 2013 to June 2022), and the test set again encompasses the subsequent 4 quarters (July 2022 to June 2023).

This methodology permits us to assess the model's performance across various prediction horizons. Additionally, it captures the impact of incorporating updated training data over time.

Figure 3 : Expanding window backtesting timeline.

It's important to note that while backtesting is a powerful tool for evaluating strategies, it has limitations. Past performance does not guarantee future results, and backtesting assumes that market conditions and dynamics will remain consistent. Changes in market behavior, data quality, and other unforeseen factors can impact the real-world performance of a strategy.

3.5 Model performance

3.5.1 Metrics

To evaluate the performance of our model, we will use some forecast accuracy metrics.

First, the Mean Absolute Error (MAE) is a metric that measures the average absolute difference between the predicted values and the actual values. It quantifies the average magnitude of the errors, regardless of the direction. It is calculated as:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}|$$

where n is the number of data points, y_i the actual and $\hat{y_i}$ the predicted. MAE provides a straightforward way to measure the forecast accuracy in the same units as the original data.

The Mean Squared Error (MSE) is another commonly used metric that measures the average squared difference between the predicted values and the actual values. It gives more weight to larger errors, making it more sensitive to outliers. MSE is calculated as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2$$

We also have the Root Mean Squared Error (RMSE) which is the square root of the MSE and is probably the most widely used metric for evaluating forecast accuracy. It is expressed in the same units as the original data and has the advantage of penalizing large errors more than MAE. RMSE is calculated as the square root of the MSE:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2}$$

Finally, we will also determine the Mean Absolute Percentage Error (MAPE) which is a percentage-based metric that measures the average absolute percentage difference between the predicted values and the actual values. It gives a sense of the relative forecast accuracy and is commonly used when dealing with data on different scales. MAPE is calculated as:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

3.5.2 Comparison model: naive model

We will conduct an evaluation of our forecasting model by creating a naive benchmark model for comparison. This benchmark model is designed to simulate an investor's random betting strategy for different companies. To construct the benchmark, we generated random numbers from a normal (Gaussian) distribution, incorporating the historical volatility of each company's revenue surprises.

To be specific, for each company 'i', we generated random surprises values with the following parameters:

$$\mu = 0$$

$$volatility = sd(revenue\_surprise_i)$$

By employing this benchmark model, we establish a baseline that represents an investor making random bets on various companies. Comparing the performance of our ARIMAX model against this benchmark provides a crucial measure of its effectiveness. If our ARIMAX model consistently outperforms the random benchmark, it indicates significant predictive capability and a valuable edge over random guessing.

This comprehensive evaluation allows us to gain insights into the model's reliability and potential for making informed investment decisions. A successful comparison with the random benchmark reinforces our confidence in the model's ability to provide actionable forecasts and meaningful recommendations for investors seeking profitable opportunities in the market.

4 Empirical findings

In this section, the whole testing process will be presented, followed by an analysis of the results.

To begin with, the data imported from Refinitiv Eikon will undergo several modifications. It will be passed through the function "companies_list_fct()" to organize the data into lists, each corresponding to a specific company. Then, the function "make_stationary()" will be applied to differentiate the revenue surprise variables until the data meets the stationarity requirement, as determined by the adf.test().

Following this, the "make_complete_dataset()" function will be implemented to further refine and organize the data, making it suitable for subsequent analysis.

As we proceed, it's important to note that a separate model will be constructed for each company. The next step involves categorizing the variables into two groups: the dependent variable, which we aim to predict, and the independent variables (exogenous variables). Both groups will be divided into training and test sets, with the test set used to evaluate the performance of the model developed using the training set.

Next, the "auto.arima()" function, incorporating our exogenous variables, will be used to calibrate the ARIMAX model based on the lowest AIC criterion. This calibrated model will then be utilized to make predictions. This entire process will be executed as a loop for each company. With one calibration taking approximately 16 seconds, the total time to process this loop for all companies is estimated to be 12 minutes and 48 seconds.

4.1 Summary of results

After this time period has elapsed, all the models have been meticulously calibrated using the available data. As an output, we obtain 12 quarters of forecasted surprise for each company, stored in a matrix where each row represents a company, and each column corresponds to a quarter. This matrix will be invaluable for conducting further analysis.

Let us now explore the results graphically. In the charts, we compare the actual revenue surprise to the forecasted revenue surprise, providing us with an initial understanding of the model's performance. Some of the charts obtained from both the classical train-test split and the expanding window forecasting strategy are in some cases interesting as the following charts are showing.

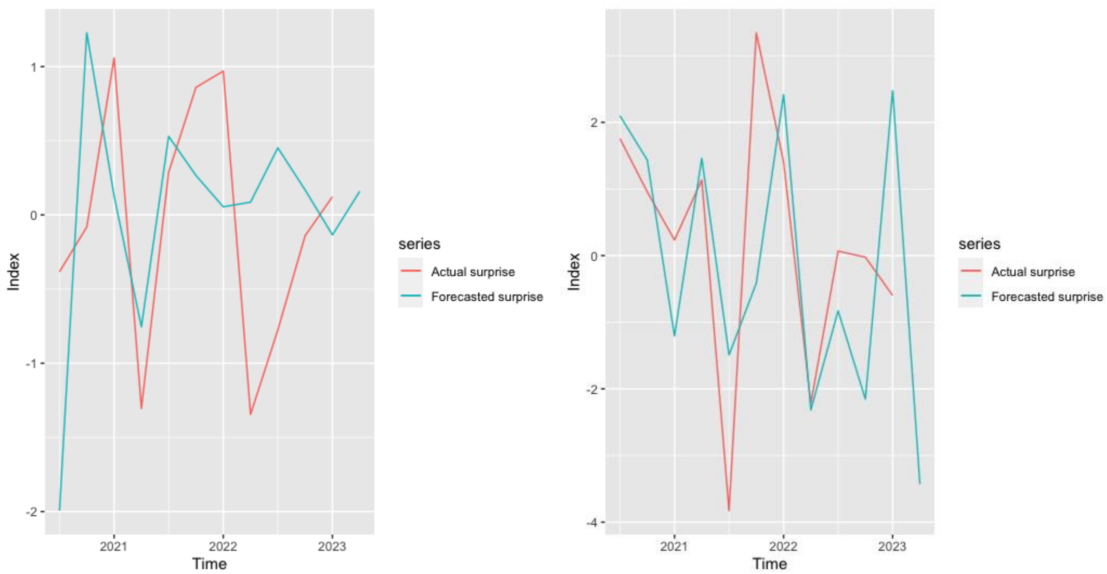

Figure 4 : Forecasted surprise compared to the actual surprise with train-test split model for Broadcom and Visa respectively.

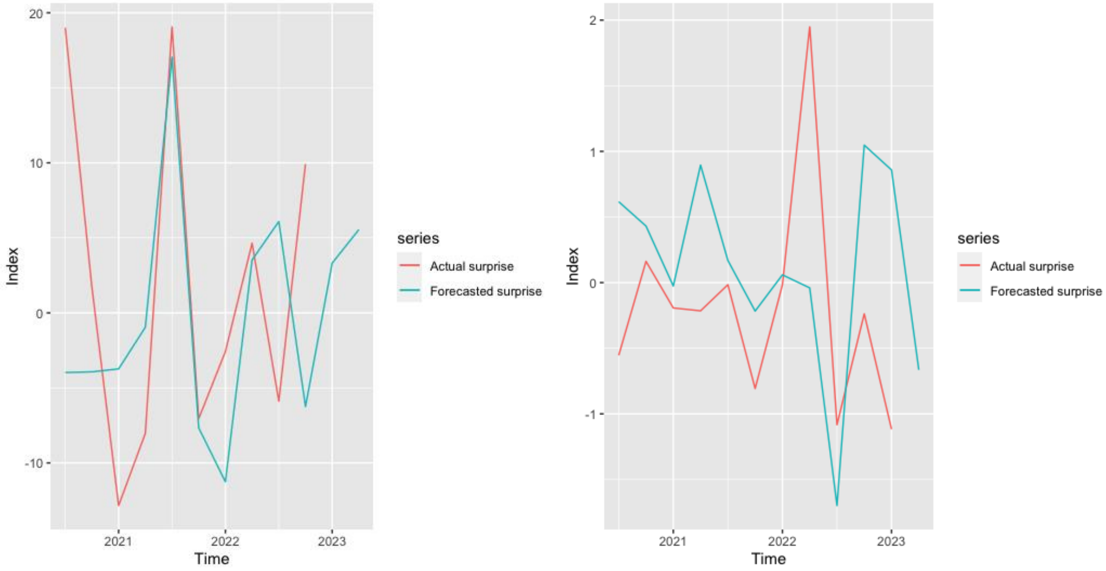

Figure 5 : Forecasted surprise compared to actual surprise with expanding window strategy model for Apple and Netflix respectively.

However, upon examining the outcomes for other companies, we find that they are not as informative as the results showed higher. The following examples for both the strategies are way less performing.

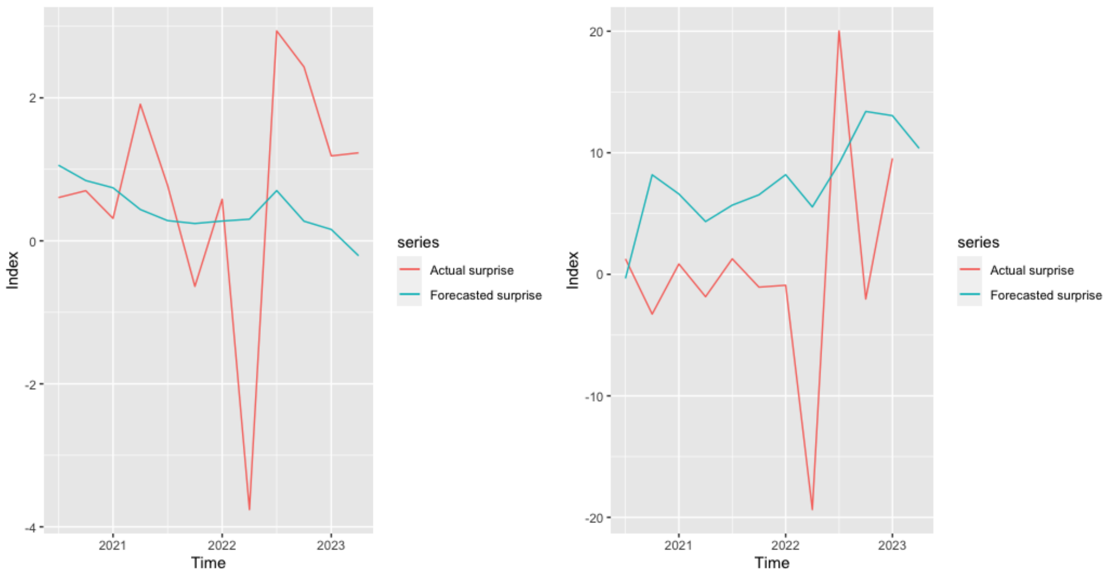

Figure 6 : Forecasted surprise compared to the actual surprise with train-test split model for NVIDIA corp and Cisco Systems respectively.

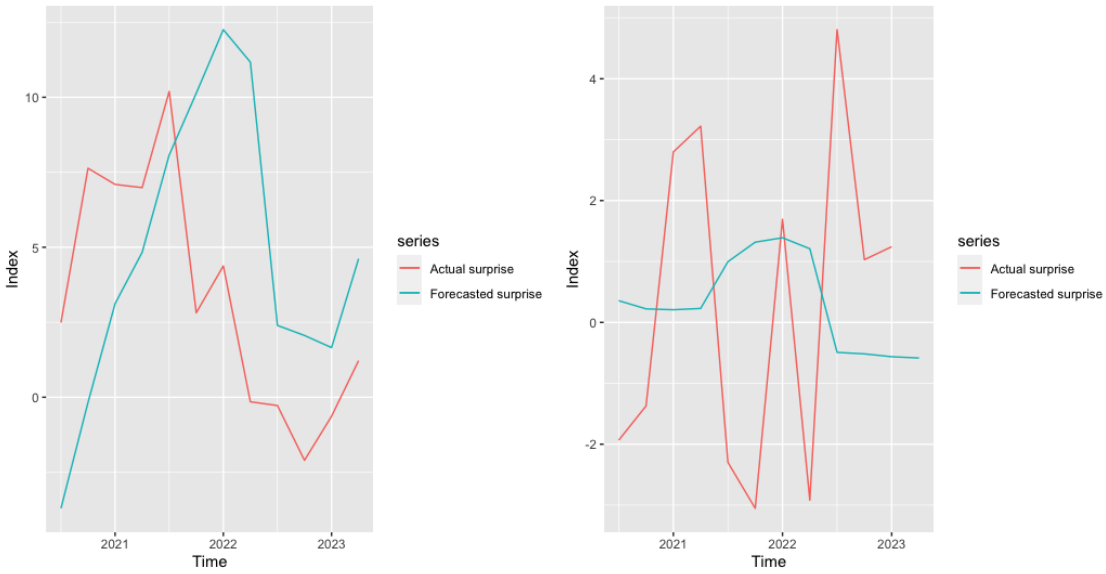

Figure 7 : Forecasted surprise compared to actual surprise with expanding window strategy model for Alphabet and McDonald's respectively.

However, given that the results shown on the graphs are mixed and that there are both positive and negative examples, the conclusions we can draw from them are fairly limited.

Continuing with our analysis, we have calculated the Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) for our forecasted values. Surprisingly, we found that these metrics obtained for the train-test split model are higher than the corresponding values obtained from the naive random model.

MSE	MAE	RMSE	MAPE
72.1194	5.0719	6.1621	905.5052

Figure 8 : ARIMAX model error metrics.

MSE	MAE	RMSE	MAPE
46.0775	4.4383	5.5170	674.1071

Figure 9 : ARIMAX model with expanding window error metrics.

MSE	MAE	RMSE	MAPE
39.8862	4.0773	5.0507	682.7822

Figure 10 : Naive model error metrics.

The implications of this outcome are significant. It suggests that our forecasting model, despite its sophistication, is performing less accurately than a basic naive random model that makes predictions based on random chance. Such a result indicates that our forecasting model may not be capturing the underlying patterns and trends in the data effectively.

However, with the expanding window forecasting strategy, the results obtained are quite different. Indeed, we can observe that the error metrics are lower than for the classical train-test split and are close to the naive random model. Nevertheless, the metrics obtained still does not show that this second approach gave a significantly better performing model than the naive approach.

Subsequently, a dataset was generated, containing a value of 1 if the forecasted surprise matched the actual surprise in sign, and 0 if the signs differed. This methodology was also applied to the naïve random model, facilitating a comparison between the two approaches. The objective was to ascertain whether our technique surpassed random stock investments to a statistically significant degree. An illustrative example featuring the first six companies is presented.

Companies	apr-2020	jul-2020	oct-2020	jan-2021	apr-2021	jul-2021
Abbott Laboratories	0	0	1	1	0	0
Abbvie Inc	0	0	1	1	0	0
Adobe Inc	0	1	0	0	0	0
Advanced Micro Devices Inc	1	1	0	1	1	0
Alphabet Inc	0	0	1	1	1	1
Amazon.com Inc	0	1	0	0	1	1

Table 4 : Example of result table reduced to 6 periods and 6 companies.

Unfortunately, upon examining the accuracy of the forecasted surprise signs, the ARIMAX model achieved a correctness rate of 47.42%. When we extend our approach to the expanding window, we only obtain a low improvement with 47.65% of correctness rate, while the naive random model achieved a rate of 49.30%. Regrettably, this indicates that the model is not functioning optimally, hindering the possibility of implementing an active investment strategy with these companies.

5 Conclusion

In this study, our objective was to forecast quarterly revenue surprises using time series forecasting models. We conducted our analysis on a database comprising 48 different companies and experimented with various models. Ultimately, the ARIMAX model emerged as the most suitable option for our specific situation and dataset.

Two different approaches were explored. Initially, a conventional train-test split was employed, followed by experimentation with an expanding window backtesting strategy. The second approach exhibited lower errors compared to the first one but had still similar error metrics than those of a naive random model.

Despite our efforts, we discovered that the ARIMAX model does not significantly outperform a random model in predicting revenue surprises. Consequently, relying on our model to implement an active investment strategy would carry substantial risk.

For future investigations, exploring alternative forecasting models that may offer greater reliability for this purpose could prove valuable. Additionally, venturing beyond time series forecasting and exploring other modeling approaches could provide new perspectives.

A critical limitation we encountered in this research was the size of our data. Working with very recent quarterly data restricted the comprehensiveness of our databases. This limitation constrained our model selection as many models require a larger amount of data points. For further research, our techniques could be improved with a larger amount of quarterly brokers' estimates, it will also enable the use of other time series forecasting techniques such as Vector Autoregressive models.

Another way to explore the improvement of our models is in the variable selection. Our techniques could be tested with other independent variables. Exploring the incorporation of different fundamental value expectations or a broader range of macroeconomic indicators could offer interesting possibilities.

In conclusion, our results did not show that our models were significantly outperforming random investing decisions. Nevertheless, further research could improve these models by using other variables and a larger amount of data points.

6 Bibliography

1) Umstead, D. A. (1977). Forecasting Stock Market Prices. *The Journal of Finance, 32*(2), 427–441. https://doi.org/10.2307/2326776

2) Agrawal, J. G., Chourasia, V., & Mittra, A. (2013). State-of-the-art in stock prediction techniques. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 2(4), 1360-1366.

3) McClure, B. (2021, December 12). What Are Stock Fundamentals? Investopedia. Retrieved from https://www.investopedia.com/articles/fundamental/03/022603.asp

4) Gentry, W. M., Jones, C. M., & Mayer, C. J. (2004). Do Stock Prices Really Reflect Fundamental Values? The Case of REITs. *National Bureau of Economic Research*. Retrieved from https://doi.org/10.3386/w10850

5) Alberg, J., & Lipton, Z. C. (2018). Improving Factor-Based Quantitative Investing by Forecasting Company Fundamentals (No. 1711.04837). Retrieved from arXiv preprint archive.

6) Jegadeesh, N., & Livnat, J. (2006). Revenue surprises and stock returns. Journal of Accounting and Economics, 41(1-2), 147-171. https://doi.org/10.1016/j.jacceco.2005.10.003

7) Livnat, J., & Jegadeesh, N. (2006). Post-Earnings-Announcement Drift: The Role of Revenue Surprises. Financial Analyst Journal, 62(2).

8) Da, Zhi and Engelberg, Joseph and Gao, Pengjie, In Search of Fundamentals (March 15, 2011). AFA 2012 Chicago Meetings Paper, Available at SSRN: https://ssrn.com/abstract=1589805 or http://dx.doi.org/10.2139/ssrn.1589805

9) Henderson, B. J., & Marks, J. M. (2013). Predicting forecast errors through joint observation of earnings and revenue forecasts. Journal of Banking & Finance, 37(11), 4265-4277. https://doi.org/10.1016/j.jbankfin.2013.07.005

10) Rees, Lynn L. and Sivaramakrishnan, Shiva, The Effect of Meeting or Beating Revenue Forecasts on the Association between Quarterly Returns and Earnings Forecast Errors (August 2004). Available at SSRN: https://ssrn.com/abstract=655421 or http://dx.doi.org/10.2139/ssrn.655421

11) Mondal, P., Shit, L., & Goswami, S. (2014). Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices. International Journal of Computer Science, Engineering and Applications, 4(2), 13.

12) Rotela Junior, P. , Salomon, F. and de Oliveira Pamplona, E. (2014) ARIMA: An Applied Time Series Forecasting Model for the Bovespa Stock Index. *Applied Mathematics*, 5, 3383-3391. doi: 10.4236/am.2014.521315.

13) Candelon, B. (2021). Forecasting. Louvain School of Management.

14) Mehandzhiyski, V. (2023). What is an ARIMAX model ? *365 Data Science*. https://365datascience.com/tutorials/python-tutorials/arimax/

Note

This thesis was produced with the help of artificial intelligence during the research and writing phase. The following AIs were used: Grammarly, Deepl, ChatGPT and LanguageTool.

7 Appendix

R functions

Appendix 1 : companies_list_fct

This function is used just after the importation of our data, in our data transformation process. This function takes as input a list with the datatables (Revenue_surprise, Revenue_mean, Ebit_mean, Ebitda_mean) and sort every information per company. The return of this function is a list where each element corresponds to a company and these elements are organized in a way that every variable will be sorted in its corresponding company.

```
#Function that sort every datapoint with his corresponding company
companies_list_fct <- function(list){
  n_data <- length(list_data_points)
  n_companies <- nrow(list_data_points[[1]])
  cl <- vector("list",n_companies)
  for (i in 1:n_data) {
    for (j in 1:n_companies) {
      data_vector <- as.data.frame(t(list_data_points[[i]][j,]))
      cl[[j]][i] <- data_vector
    }
  }
 return(cl)
}
```

Appendix 2 : make_stationary

This function will be useful during our testing phase. Indeed, it takes some data as input, makes it stationary according to the adf.test() and returns the stationary variable and the number of differencing that was needed.

```
#make data stationary
make_stationary <- function(data){
  n_diff <- 0
  adf_test <- adf.test(data)
  pval <- adf_test$p.value
  while(pval>0.05){
    data <- diff(data)
    adf_test <- adf.test(data)
    pval <- adf_test$p.value
    n_diff <- n_diff +1
  }
  return_list <- list(data,n_diff)
  return(return_list)
}
```

Appendix 3 : make_complete_dataset

This function is used in the data transformation phase, it adjusts the order of our dataset, and makes it suitable for time series objects.

```
make_complete_dataset <- function(dataset){
  df <- as.data.frame(dataset)

  len <- length(df)

  lenrow <- length(df[,1])
  result <- data.frame(matrix(vector(), ncol = len, nrow =lenrow))

  for(i in 1:len){
    test_data <- as.data.frame(dataset[i])
    inv <- test_data[nrow(test_data):1,]
    result[i] <- inv
}
  return(result)
}
```

Appendix 4 : same_sign

This is a basic function that is useful in the results analysis, it returns a '1' if both values are the same sign, 0 elsewhere. This allows us to make a dataset that outlines for each datapoint if the values are the same sign.

```
#Same sign function
same_sign <- function(num1, num2) {
  if ((num1 >= 0 & num2 >= 0) | (num1 < 0 & num2 < 0)) {
    return(1)
  } else {
    return(0)
  }
}
```

Appendix 5 : count_0s_1s

This is a function used in the result analysis. It gives us the number of values that are the same sign and the number of values that are not.

```
#Count 0 and 1 function
count_0s_1s <- function(input_matrix) {
  # Count the occurrences of 0 and 1 in the input matrix
  count_zeros <- sum(input_matrix == 0, na.rm = TRUE)
  count_ones <- sum(input_matrix == 1, na.rm = TRUE)

  # Return the counts as a named list
  result_list <- list(Zeros = count_zeros, Ones = count_ones)
  return(result_list)
}
```

Appendix 6 : make_testset

This function is useful for the result analysis. It returns a complete test set, allowing us to compare with our forecasts when we analyze our results.

```
make_testset <- function(data,vectordata){
  df_save_test <- matrix(nrow=48,ncol=12)
  for (i in vectordata){
   complete_data <- make_complete_dataset(data[[i]])
   y_part <- complete_data[,1]
   list_stat <- make_stationary(y_part)
   y_part <- list_stat[[1]]
   y_test_set <- ts(y_part[31:42], start = c(2020,3), frequency = 4)
   df_save_test[i,] <- as.vector(y_test_set)

  }
  return(df_save_test)
 }
```

Appendix 7 : backtest_function_3times

This is the function that is making the forecasts with an expanding window strategy instead of a classical train-test split. Takes as input the dataset with all our variables and a vector with the company numbers without NAs and outputs a list with the forecasts and a table where a 1 means that the forecast sign and the actual sign are the same and 0 elsewhere.

```r
backtest_function_3times <- function(cdata, gvect){
  backtest_levels <- list(c(31:34),c(35:38),c(39:42))

  date_vect <- seq.Date(from = ymd("2013-01-01"), by = "3 months", length.out = 42)

  total_forecast <- data.frame(matrix(nrow = length(gvect),ncol = 12))

  backtest_nright <- data.frame(matrix(nrow = length(gvect),ncol = 12))

  for(h in 1:length(backtest_levels)){
    trainset_enddate <-date_vect[backtest_levels[[h]][1]-1]
    testset_begin <- date_vect[backtest_levels[[h]][1]]
    for (i in gvect) {

      complete_data <- make_complete_dataset(cdata[[i]])
      y_part <- complete_data[,1]
      list_stat <- make_stationary(y_part)
      y_part <- list_stat[[1]]

      y_train_set <- ts(y_part, start = c(2013,1), frequency = 4, end=c(year(trainset_enddate),(month(trainset_enddate)+2)/3))
      y_test_set <- ts(y_part[backtest_levels[[h]]], start = c(year(testset_begin),(month(testset_begin)+2)/3), frequency = 4)

      #Here, we make a dataset with what we will us as external regressors
      xreg_part <- data.frame(complete_data[,2:4],unemploy_rate$`UNRATE["2013-01/2023"].Close`,usd_index$`DTWEXBGS["2013-01/2023"].Close`)
      xreg_train_set <- ts(xreg_part, start = c(2013,1), frequency = 4, end=c(year(trainset_enddate),(month(trainset_enddate)+2)/3))
      xreg_test_set <- ts(xreg_part[backtest_levels[[h]],], start = c(year(testset_begin),(month(testset_begin)+2)/3), frequency = 4)

      jenkins_arimax <- auto.arima(y_train_set, xreg = xreg_train_set, stepwise = FALSE, approximation = FALSE, seasonal = TRUE, trace = TRUE)
      prediction_arimax <- predict(jenkins_arimax, n.ahead = 4, newxreg = xreg_test_set)

      total_forecast[i,(4*h-3):(4*h)] <- prediction_arimax$pred


      for (j in 1:sum(!is.na(y_test_set))){
        backtest_nright[i,(h-1)*4+j] <- same_sign(prediction_arimax$pred[j],y_test_set[j])
      }

    }
  }
  return_list <- list(total_forecast,backtest_nright)
  return(return_list)
}
```

Result tables

This is the complete result table of our forecasted values, ‘1’ means that the forecasted value and real value are the same sign, 0 means that they are not. The “NA”s happens at the end of a forecast when there has been differencing in the testing set or when the imported data from Refinitiv was composed by missing values, in order to keep a relevant forecast, the companies with missing values were not taken into account into the forecasting model.

Appendix 8 : Result table for ARIMAX with train-test split

Companies	FQ 1	FQ 2	FQ 3	FQ 4	FQ 5	FQ 6	FQ 7	FQ 8	FQ 9	FQ 10	FQ 11	FQ 12
Abbott Laboratories	0	0	1	1	0	0	1	0	0	1	0	NA
Abbvie Inc	0	0	1	1	0	0	0	0	0	1	1	NA
Adobe Inc	0	1	0	0	0	0	1	0	0	1	0	NA
Advanced Micro Devices Inc	1	1	0	1	1	0	1	1	1	0	1	NA
Alphabet Inc	0	0	1	1	1	1	1	0	0	0	0	1
Amazon.com Inc	0	1	0	0	1	1	1	1	1	0	1	NA
Apple Inc	0	0	1	1	0	0	0	1	1	1	NA	NA
Bank of America Corp	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Berkshire Hathaway Inc	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Broadcom Inc	1	0	1	1	1	1	1	0	0	0	0	NA
Chevron Corp	0	0	0	1	0	1	1	0	0	0	1	NA
Cisco Systems Inc	1	1	1	1	1	0	1	0	1	1	1	0
Coca-Cola Co	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Comcast Corp	1	1	0	1	0	0	1	1	0	0	0	NA
Costco Wholesale Corp	1	0	0	0	0	0	1	0	0	1	NA	NA
Danaher Corp	1	0	1	0	0	0	0	1	1	0	0	NA
Eli Lilly and Co	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Exxon Mobil Corp	0	1	0	0	1	1	1	0	1	0	0	NA
Home Depot Inc	0	0	1	1	0	1	1	1	0	1	NA	NA
Johnson & Johnson	1	0	1	1	1	1	1	0	1	0	0	NA
JPMorgan Chase & Co	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Mastercard Inc	0	1	1	0	0	0	0	1	1	1	0	NA
McDonald's Corp	0	0	1	1	0	0	1	0	1	1	1	NA
Merck & Co Inc	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Meta Platforms Inc	0	1	1	1	1	0	1	0	1	0	0	0
Microsoft Corp	0	1	0	1	0	0	0	0	1	0	1	NA
Morgan Stanley	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Netflix Inc	0	1	1	0	0	0	1	1	0	0	1	NA
Nike Inc	0	0	1	0	0	0	1	0	0	0	1	NA
NVIDIA Corp	0	0	1	0	1	0	0	0	1	0	1	NA
Oracle Corp	0	1	1	1	1	1	0	1	0	1	0	NA
PepsiCo Inc	0	0	1	1	1	0	0	0	0	0	0	NA
Pfizer Inc	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Philip Morris International Inc	1	0	0	0	0	0	1	1	1	1	1	NA
Procter & Gamble Co	1	0	0	1	0	0	0	1	0	1	0	NA
Salesforce Inc	1	0	0	0	1	0	1	0	1	0	1	NA

Starbucks Corp	1	0	1	1	0	1	0	0	1	0	1	NA
Tesla Inc	1	0	0	0	0	0	0	1	1	0	0	0
Texas Instruments Inc	0	1	1	0	1	0	1	0	0	1	1	NA
Thermo Fisher Scientific Inc	0	0	0	0	1	1	0	1	1	0	NA	NA
T-Mobile US Inc	1	0	1	0	0	1	1	0	0	1	1	NA
United Parcel Service Inc	0	1	1	0	0	1	0	1	0	1	NA	NA
UnitedHealth Group Inc	1	1	0	1	0	0	1	0	1	0	NA	NA
Verizon Communications Inc	0	1	1	1	0	0	1	0	0	1	1	NA
Visa Inc	1	1	0	1	1	0	1	1	0	1	0	NA
Walmart Inc	0	0	0	0	1	1	0	1	0	1	NA	NA
Walt Disney Co	0	1	1	0	1	0	1	0	1	0	1	
Wells Fargo & Co	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Appendix 9 : Result table for ARIMAX with expanding window backtest strategy

Companies	FQ 1	FQ 2	FQ 3	FQ 4	FQ 5	FQ 6	FQ 7	FQ 8	FQ 9	FQ 10	FQ 11	FQ 12
Abbott Laboratories	0	0	1	1	1	1	1	0	0	1	0	NA
Abbvie Inc	0	0	1	1	1	0	0	1	1	1	0	NA
Adobe Inc	0	1	0	0	0	0	0	1	1	0	0	NA
Advanced Micro Devices Inc	1	1	0	1	1	0	1	1	1	1	0	NA
Alphabet Inc	0	0	1	1	1	1	1	0	0	0	0	1
Amazon.com Inc	0	1	0	0	1	0	0	0	1	0	1	NA
Apple Inc	0	0	1	1	1	1	1	1	0	0	NA	NA
Bank of America Corp	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Berkshire Hathaway Inc	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Broadcom Inc	1	0	1	1	1	0	0	0	0	1	0	NA
Chevron Corp	0	0	0	1	0	1	0	1	0	0	0	NA
Cisco Systems Inc	1	1	1	1	1	0	1	0	1	1	0	0
Coca-Cola Co	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Comcast Corp	1	1	0	1	0	0	1	1	1	0	0	NA
Costco Wholesale Corp	1	0	0	0	0	0	1	0	0	1	NA	NA
Danaher Corp	1	0	1	0	0	0	1	1	0	1	1	NA
Eli Lilly and Co	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Exxon Mobil Corp	0	1	0	0	1	1	0	1	0	0	0	NA
Home Depot Inc	0	0	1	1	0	1	0	1	0	0	NA	NA
Johnson & Johnson	1	0	1	1	0	0	1	0	1	0	0	NA
JPMorgan Chase & Co	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Mastercard Inc	0	1	1	0	0	0	0	1	1	1	1	NA
McDonald's Corp	0	0	1	1	0	0	1	0	0	0	0	NA
Merck & Co Inc	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Meta Platforms Inc	0	1	1	1	1	0	1	0	0	1	1	1
Microsoft Corp	0	1	0	1	1	1	1	0	0	1	0	NA
Morgan Stanley	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Netflix Inc	0	1	1	0	0	1	0	0	1	0	0	NA
Nike Inc	0	0	1	0	1	0	1	0	0	1	0	NA
NVIDIA Corp	0	0	1	0	0	1	1	1	1	0	1	NA
Oracle Corp	0	1	1	1	0	1	1	1	0	1	0	NA
PepsiCo Inc	0	0	1	1	1	0	0	0	1	0	0	NA
Pfizer Inc	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Philip Morris International Inc	1	0	0	0	0	1	1	1	0	0	1	NA
Procter & Gamble Co	1	0	0	1	1	0	0	1	1	0	0	NA
Salesforce Inc	1	0	0	0	1	0	1	1	1	0	1	NA
Starbucks Corp	1	0	1	1	1	1	0	1	0	1	0	NA
Tesla Inc	1	0	0	0	0	0	0	1	1	1	1	0
Texas Instruments Inc	0	1	1	0	1	0	1	0	1	0	1	NA
Thermo Fisher Scientific Inc	0	0	0	0	0	0	1	1	0	1	NA	NA
T-Mobile US Inc	1	0	1	0	0	1	0	0	0	0	0	NA
United Parcel Service Inc	0	1	1	0	1	0	0	0	0	0	NA	NA
UnitedHealth Group Inc	1	1	0	1	1	1	1	1	0	0	NA	NA
Verizon Communications Inc	0	1	1	1	0	1	1	0	0	0	1	NA
Visa Inc	1	1	0	1	1	0	1	1	1	1	0	NA
Walmart Inc	0	0	0	0	1	0	0	1	0	1	NA	NA
Walt Disney Co	0	1	1	0	1	0	1	0	0	1	0	NA
Wells Fargo & Co	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA

Abstract:

In the context of active investing, many traders have attempted to forecast stock values using machine learning techniques. However, it has always been seen as a hard and challenging task. In this thesis, we aim to predict revenue surprises prior to quarterly revenue publications using time series forecasting models based on brokers' estimates and macro-economic indicators. The main objective of these forecasts is to capitalize on the direct impact of revenue surprises on stock prices through an active investment strategy. Studies cited and exploited in this research showcase the significance of forecasting surprises on companies' fundamentals, rather than focusing directly on these fundamentals. Our results, however, do not conclusively establish our model's superiority over random investment decisions. In summary, the results of this thesis suggest that enhancing the quality of datasets used in predictive models, along with leveraging more powerful forecasting models, could improve the quality of our predictions.

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