



TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

Diplomarbeit

Business Cycle-based Sales Forecasting: Creating a Tool for Predictive Accuracy Analysis

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines
Diplomingenieurs unter der Leitung von

Univ.Prof. Mag. Dr. Walter S.A. Schwaiger, MBA

Institut für Managementwissenschaften
Lehrstuhl für Finanzwirtschaft und Controlling

eingereicht an der Technischen Universität Wien
Fakultät für Maschinenwesen und Betriebswissenschaften

von

Michael Flatscher, BSc.

1227697

Markus Pribyl, BSc.

1126567

Wien, 8. Juli 2019

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass die vorliegende Arbeit nach den anerkannten Grundsätzen für wissenschaftliche Abhandlungen von mir selbst erstellt wurde. Alle verwendeten Hilfsmittel, insbesondere die zugrunde gelegte Literatur, sind in dieser Arbeit genannt und aufgelistet. Die aus Quellen wörtlich entnommenen Stellen, sind als solche kenntlich gemacht.

Das Thema dieser Arbeit wurde von mir bisher weder im In- noch Ausland einer Beurteilerin/einem Beurteiler zur Begutachtung in irgendeiner Form als Prüfungsarbeit vorgelegt. Diese Arbeit stimmt mit der von den Begutachterinnen/Begutachtern beurteilten Arbeit überein.

Verantwortlich für die Kapitel 1, 2, 4, 5.1 und 8 zeichnet sich Michael Flatscher. Verantwortlich für die Kapitel 3, 5.2, 6 und 7 zeichnet sich Markus Pribyl.

Michael Flatscher
Wien, am 8. Juli 2019

Markus Pribyl
Wien, am 8. Juli 2019

Affidavit

I declare in lieu of oath, that I wrote this thesis and performed the associated research myself, using only literature cited in this volume. If text passages from sources are used literally, they are marked as such.

I confirm that this work is original and has not been submitted elsewhere for any examination, nor is it currently under consideration for a thesis elsewhere.

Responsible for chapters 1, 2, 4, 5.1 and 8 is Michael Flatscher. Responsible for chapters 3, 5.2, 6 and 7 is Markus Pribyl.

Michael Flatscher
Vienna, July 8, 2019

Markus Pribyl
Vienna, July 8, 2019

Abstract

Following the approach of design research methodology, the design of a software tool that can construct sales forecasts using leading macroeconomic indicators and analyse the predictive accuracy of these sales forecasts will be considered an artefact. The software tool will be usable by people with average IT knowledge. The software tool will be applicable both for scientists and managers, enabling them to choose different methods and parameters during the process of constructing a sales forecast and subsequently enable them to determine the predictive accuracy. Additionally, the tool will be tested with real world data of a real company.

Kurzfassung

Dem Ansatz der Design Research Methodologie folgend, wird der Entwurf eines Softwaretools, das Umsatzprognosen unter Verwendung führender makroökonomischer Indikatoren erstellen und die Vorhersagegenauigkeit dieser Verkaufsprognosen analysieren kann, als Artefakt betrachtet. Das Software-Tool kann von Personen mit durchschnittlichen IT-Kenntnissen verwendet werden. Das Software-Tool ist sowohl für Wissenschaftler als auch für Manager geeignet, so dass diese während des Erstellungsprozesses einer Umsatzprognose verschiedene Methoden und Parameter auswählen und anschließend die Prognosegenauigkeit bestimmen können. Zusätzlich wird das Tool mit realen Daten einer wirklichen Firma getestet.

Contents

1. Introduction	8
2. Using Business Cycles for Sales Forecasts	10
2.1. Importance of Business Cycles	11
2.2. Introduction to Sales Forecasts	14
2.3. Issues with Macroeconomic Forecasts	18
2.4. Sales Forecasts Using Macroeconomic Indicators	20
2.5. Construction of Sales Forecasts	22
2.6. Use and Modification of Sales Forecasts	26
3. Objectives of GNOSIS	28
3.1. General Requirements	29
3.2. Construction Methods	31
3.3. Calibration Methods	40
3.4. Validation Methods	43
3.5. Usability	48
4. Design and Development of GNOSIS	49
4.1. Methodology	50
4.2. Structure of GNOSIS	52
4.3. Decision for R	57
4.4. Shiny	59
4.5. Functions	60
4.6. User Interface	63
4.7. Server	66

5. Demonstration of GNOSIS	68
5.1. Creating a Sales Forecast using GNOSIS	69
5.1.1. Data Acquisition	70
5.1.2. ROC Statistics	72
5.1.3. Data Normalisation	74
5.1.4. Indicator Correlations	75
5.1.5. Composite Index Construction	77
5.1.6. ROC Forecasts	79
5.1.7. Sales Forecasts	81
5.1.8. Report	83
5.2. Analysis of Differently Constructed Sales Forecasts	84
5.2.1. Length of In-sample Period	86
5.2.2. Length of Forecast Horizon	88
5.2.3. Normalisation Method	90
5.2.4. Correlation Method	92
5.2.5. Cross Correlation Value	94
5.2.6. Aggregation Method	96
5.2.7. Summary Parameter Variation	98
6. Analysis and Evaluation of GNOSIS	100
6.1. Analysing the Demonstration	101
6.1.1. Data Acquisition	102
6.1.2. ROC Statistics	103
6.1.3. Data Normalisation	104
6.1.4. Indicator Correlations	105
6.1.5. Composite Index Construction	106
6.1.6. ROC Forecasts	107
6.1.7. Sales Forecasts	108
6.1.8. Report	109
6.2. Fulfilment of Requirements	110
6.2.1. Construction	111
6.2.2. Calibration	112
6.2.3. Validation	113
6.2.4. Usability	114

7. Discussion	117
7.1. The Artefact GNOSIS	118
7.2. Forecasts Generated Using GNOSIS	120
7.3. Outlook	122
8. Conclusion	123
Abbreviations and Acronyms	124
List of Tables	125
List of Figures	126
Listings	128
Appendices	136
A. Deriving Sales Forecast from CI Calculated with ROC_{12/12}	137
B. Packages Needed for Server and UI of GNOSIS	138
C. List of Indicators for Demonstration	139
D. Parameter Variation: Deviation of Averaged Sales Forecast	140
E. Evaluation of GNOSIS - Usability: Additional Criteria	144

1. Introduction

This thesis will describe the design and development of a software tool for the construction of sales forecasts using time-lagged macroeconomic indicators. The first chapter will conduct an in-depth literature review concerning the influence of business cycles on companies, the current state-of-the-art in sales forecasting and the combination of using business cycles for enhanced sales forecasting. Additionally, this thesis will elucidate the use of sales forecasts after their construction, i.e. the judgemental adaptations of sales forecasts.

The next chapter will describe the requirements posed at the software tool, which are the calculation methods found in the literature with the addition of newly adapted methods, where no suitable ones were found in the literature. Here, the construction methods, the calibration methods and the validation methods the software tool must have implemented will be explained. Additionally, the topic of usability will be addressed and the requirements regarding the usability will be specified.

Building onto the requirements, the design and development process of the software tool will be described. Here a focus will be set on the explanation for the choice of the software, a description of the functionality and logic of the code with examples of the code of the software tool will be discussed.

The next chapter will serve as a demonstration of the software tool. For this, the sales data of a real company will be used and the functionality of the software tool will be explained by demonstrating the process of constructing, calibrating and validating a sales forecast. Additionally, a parameter variation will be performed to illustrate the impact of different possible choices on the outcome, in terms of the predictive accuracy of a ROC (Rate of Change) forecast and a sales forecast.

Based on the demonstration, the software tool will be analysed and subsequently evaluated. The focus of the evaluation of the software tool is to assess whether all the requirements defined prior to building the tool were fulfilled.

The last chapter will feature a discussion, concerning the software tool itself, the predictive accuracy of the forecasts generated with the software tool and provide a comprehensive overview of possible future topics for research.

2. Using Business Cycles for Sales Forecasts

The following sections are the basis for the development of a software tool for the predictive accuracy analysis of sales forecasts constructed with macroeconomic indicators. First it is necessary to examine the current literature concerning the importance of business cycles for the planning of companies, which is the content of the first section.

The second section will serve as an introduction to sales forecasts, their significance for companies and the different approaches found in the literature. The third section will explain, why companies can not just rely on business cycle forecasts as an addition to a standard sales forecast.

The fourth section will examine the current literature on the topic of sales forecasts constructed with macroeconomic indicators. The fifth section explains the different steps that are undertaken by researchers in order to construct a sophisticated sales forecast. The final section will examine the research on the further use and modification of sales forecasts in companies after their construction and calibration.

2.1. Importance of Business Cycles

The existence of business cycles has been a topic for research since the early 20th century, i.e. the typical duration of a business cycle was already discussed in 1926 (Mills (1926)). A business cycle can be described by four distinct phases. The first one being an economic expansion, the second one being a recession, the third one a depression (also called contraction) and the fourth one being a revival of economic activities. These phases are not strictly periodical and do not always occur in this order (Kufenko and Geiger (2016)). For instance, an expansion could directly follow a recession, or a depression could occur directly after a revival phase.

These business cycles have typically a medium length, meaning their duration ranges from one to twelve years, according to Kufenko and Geiger (2016). There are also longer business cycles, for instance the Kondratiev waves, which were found to be existent in metal commodity prices and exhibit a duration of 50-60 years (Maranon and Kumral (2019)). Due to their long duration, these longer business cycles do not abruptly change the economic situation and are therefore less disrupting for the accuracy for medium and short range sales forecasts. This thesis will therefore use the term “business cycle” for the business cycles with a typical duration of one to twelve years.

The underlying factors and reasons for business cycles are not entirely understood. There is an attempt at explaining the dynamics of business cycles by so called technological and non-technological shocks. These shocks can be new technologies, which mostly stimulate the economy in a positive way or non-technological shocks, for instance environmental hardships or war, which mostly have a negative impact. Forni and Reichlin (1998) examined the U.S. economy from 1958 to 1986 and found out, that over 50% of the dynamics could be explained by a technological shock and that sector-specific shocks can explain the high frequency dynamics. However, all factors responsible for business cycles are not fully understood yet.

According to Jiang et al. (2015), nearly half of the variation of the earnings of firms can be explained by macroeconomic variables. Navarro et al. (2010, p. 50) state, that “The relationship between business cycle management (BCM) and firm performance represents one of the most important but arguably least developed research streams in all

of management scholarship”. Here, they understand a set of countercyclical behaviours applied during the specific phases of a business cycle as the correct way of management during business cycles and state, that it is a way to improve a firm’s performance relative to its competitors.

This lack of research is also addressed by other researches. Deleersnyder et al. (2004) stated, that the marketing research does not focus enough on the effects of business cycles on individual companies or industries, which is quite problematic, considering that different industries are not affected with the same severity by changes in business cycles. They found out that consumer durables are “more sensitive to business-cycle fluctuations than the general economic activity, as expressed in an average cyclical volatility of more than four times the one in GNP”Deleersnyder et al. (2004, p. 347). (Here the abbreviation “GNP” stands for gross national product.) Additionally, this product category has an asymmetry regarding the reaction to business cycle changes. Their sales figures drop significantly faster during economic downward movements, than they recover in times of economic upward movements.

The scarce literature and research of a firm’s behavior during different phases of a business cycle is quite problematic, as Halling et al. (2016) point out. Contrary to popular belief, that a firm’s target leverage evolves pro-cyclically, it actually behaves the opposite way. The target leverage ratios have a counter-cyclically nature, once it is comprehensively measured.

The reason for these shortcomings concerning the current literature on business cycles could be the business cycles themselves. Kufenko and Geiger (2016, p. 44) found out that “the economic literature on business cycles is correlated with movements and changes in actual economic activity”. It appears that especially in times of crises or during recessions the interest of researchers for business cycles is relatively high and during prosperous times the amount of research into business cycles declines.

Although there is much need for further research of the effect of business cycles on individual companies, there is a lot of evidence that business cycles have a significant impact on consumers and subsequently on the sales of firms. There is also growing evidence that adjusting the marketing mix in response to macroeconomic business cycles is an important factor for a firm’s performance (Dekimpe and Deleersnyder (2017)).

The importance of business cycles for planning exceeds by far the marketing mix. According to Navarro et al. (2010, p. 52): “Expectations strongly influence many forward-looking business decisions. Sensible rules for staffing, capital investment, production levels, etc., all rely on expected sales volumes”. They further state that many companies have very misleading expectations during changes in the phases of a business cycle. This can be explained by the extrapolation of previous years’ sales into the future, without incorporating the macroeconomic changes into the sales forecast. The problem is not just that they have too high or too low expectations during phase shifts of the business cycle and miss chances or lose money because of that, but also that “having suffered from excess spending as the economy entered the recession, managers may be ‘gun shy’ about increasing spending during a recovery (Navarro et al. (2010, p. 52))”.

Therefore, it is important that the management of a firm is aware of the high impact of business cycles on their firm’s performance, actively manages the consequences of changes in the phases of a business cycle and is able to incorporate those changes into their planning and expectations (i.e. sales forecasts).

2.2. Introduction to Sales Forecasts

A key part of the planning processes of a firm is the sales forecast (H. L. Chen (2008)). Here, the typical duration of the sales forecast is 12 months into the future. This is important for the planning of inventory, production capacities and material purchases. Especially for supply chain management, these sales forecasts are a pivotal part of tactical forecasting (Sagaert et al. (2018)).

The sales forecast is one of the most important factor for a precise cost forecasting, which is very important to be able to control the budget of a company, as Lan et al. (2001) note. Their approach for cost forecasting in their study had an interesting twist, by implementing the given price change rate for the estimation of next quarter cost budget. The use of the rate of change (ROC) of certain factors can also be applied in the context of sales forecasting and will be addressed later on in this thesis.

A lot of research discussing sales forecasting focuses on statistical methods and does not necessarily focus on predicting the sales for a real world firm in a specific industry. This is for example also addressed by H. L. Chen (2008), who focused on the development of a sales forecasting model for the construction industry and could only find few studies, which were focusing on this industry. By focusing on the construction of a sales forecast using highly correlated financial and macroeconomic indicators, he was able to construct a sales forecast with a high predictive accuracy.

Another industry which is especially interesting in the context of sales forecasting is the grocery sector. Here, it is often argued that it is very difficult to predict the sales reliably, due to shorter life cycles of the individual products and the presence of promotions, which requires quite complicated models, according to Ali et al. (2009).

Trapero et al. (2015) proposed an interesting model for promotional forecasts, by using Principal Component Analysis and therefore reducing the complexity of the problem by reducing the dimensionality. Additionally, they developed a model that automatically identifies the demand dynamics, by selectively pooling information across established products and therefore being able to estimate the dynamics of newer products. Using

this approach they were able to outperform benchmark forecasts, some by experts and some using statistical methods based on weekly data.

It thus appears, that for the different industry specific problems during the construction of sales forecasts, there are also industry specific solutions. The missing literature concerning these solutions however, is a big problem for companies which want to build a reliable, scientifically sound and sufficiently accurate sales forecasting model.

Such an accurate sales forecasting model or system is now more important than ever, since these forecasts have become essential for planning in the entire supply chain (Hunchofsky et al. (2012) and Sanders and Manrodt (2003)). Nevertheless, Sagaert et al. (2018) point out that these models are still often based on historical patterns in the data and mention this as a critique aimed at the current state of sales forecasting in real companies. Additionally, it is pointed out by Guo et al. (2013) that especially in the context of dynamic supply chains it is a key factor to have a reliable forecast and by using it, being able to reduce the inventory and other related costs.

Aviv (2001) examined the benefits of collaborative forecasting along the supply chain in contrast to local forecasting of sales. This study found out, that the benefits depend on the accuracy of the forecast and that improved predictive accuracy brings strong benefits to the entire supply chain. However, it is highly dependent on the individual strengths in accurate forecasting of the participating companies. In essence, if a company brings an additional valuable technique to the table, the collaborative forecasting effort becomes worthwhile.

There are different possibilities what could constitute a valuable sales forecasting technique. Fildes et al. (2009, p. 3) noted, that the “most common approach to forecasting demand in these companies involves the use of a computerized forecasting system to produce initial forecasts and the subsequent judgemental adjustment of these forecasts by the company’s demand planners, ostensibly to take into account exceptional circumstances expected over the planning horizon”. Here a specialised computer program that produces more accurate predictions of future sales values could be such a valuable technique, but also an especially accurate judgemental adjustment could be something very valuable for collaborative forecasting.

Another example for a valuable contribution to collaborative forecasting in a supply chain, is an approach described by Hunschofsky et al. (2012). Here, it was reported, that the implementation of macroeconomic analyses into the planning process and the use of a macroeconomic planning tool resulted in an improved predictive accuracy.

A similar finding regarding the importance of macroeconomic factors was reported by Jiang et al. (2015). Here the accuracy of management earnings forecasts was examined with regard to the correlation to different business cycles. It was found that an economic recession was positively correlated with forecasting errors. Another paper underlying the potential of macroeconomic analyses for the improved forecasting accuracy was Zhang et al. (2019). It was noted, that due to the impact “macroeconomic conditions and users’ word of mouth”(Zhang et al. (2019, p. 1)) on the consumer behaviour, these factors can improve the predictive accuracy of sales forecasts, although the literature is scarce on this topic.

The main problem with univariate time series forecasting methods used for sales forecasting is that it assumes a constant process for the development of the sales, thus assuming to be able to accurately predict the future sales values by simply analysing the past values. However, this assumption is invalid due to the fact that “the univariate forecasting model cannot handle sudden changes caused by various influencing factors such as product attributes and economic environment” (Guo et al. (2013, p. 247)).

An example for the consideration of the impact of factoring in the economic environment into the planning process and the forecasting model is described by Rötheli (2017). He examined the benefits of business cycle forecasts for companies, both in the “short-run and long-run aspects” (Rötheli (2017, p. 1)). It was found out, that using publicly available information regarding the current business cycle phase and possible changes proved to be beneficial to the company. A “sharpened forecast” (Rötheli (2017, p. 1)) was even more beneficial, due to allowing the company to adjust the output mix even more accurately.

In the long-run it was important to accurately assess and predict the current and future business cycle to adjust the optimal size of operations and the optimal price for the products. It was further shown, that “even the small increase in predictive accuracy

that active business cycle forecasting has to offer may be well worth its costs”(Rötheli (2017, p. 12)). It should be noted, however, that the examined companies only forecasted the business cycles and did not forecast the sales using macroeconomic variables.

An example for using macroeconomic variables for the generation of a sales forecast with improved accuracy would be Hunschofsky et al. (2012). Here, a real world company’s motivation and approach of forecasting the sales is described. This company made the mistake in 2008, which was a record year in terms of sales, to linearly predict the growth of the sales for the year 2009. Resulting from this, they predicted a strong growth in sales and hired additional employees to be able to handle the increased demand. However, in 2009 the sales dropped significantly due to the financial crisis and this resulted in a predictive error of the sales forecast of approximately -40 %. When switching to a sales forecasting model that is using macroeconomic variables, the predictive error for the years 2010 and 2011 dropped to a monthly average value of 1.2 %.

2.3. Issues with Macroeconomic Forecasts

As seen in the previous section, the use of macroeconomic variables or the use of business cycles for sharpened sales forecasts is beneficial for a company. Due to the fact that macroeconomic forecasts already widely exist, it would be standing to reason to simply incorporate those macroeconomic forecasts into the planning process. However, there are several problems with those forecasts.

One problem is the missing consensus on what type of distribution should be used for the modelling of macroeconomic variables. The most common modelling assumes a symmetric distribution and uses a mean-variance analysis approach. This has been shown to not fully capture the distribution and therefore makes an analysis and a prediction based on this analysis a problem (Andreev et al. (2007)).

Similar to this, Franses et al. (2012) point out, that in the literature it is often assumed that macroeconomic forecasts are based on econometric models, however this is not entirely true. “In practice, however, most macroeconomic forecasts, such as those from the IMF, World Bank, OECD, Federal Reserve Board, Federal Open Market Committee (FOMC), and the ECB, are typically based on econometric model forecasts jointly with human intuition.”(Franses et al. (2012, p. 1)) They point out that this makes the forecasts often biased.

Deschamps and Ioannidis (2014) showed that the revision dynamics of macroeconomic forecasts by professional forecaster turn out to be not only autocorrelated, but also conditionally correlated with time lags. This means that a variable whose implementation or weight is altered in turn correlates with the change of other variables made at a later point in time. Thus rendering the impact or importance of such revisions questionable.

An approach to level out the bias in professional forecasters’ macroeconomic forecasts is to aggregate their forecasts into one. However, Lopez-Perez (2016, p. 9) argue, that the “survey of professional forecasters” literature assigns equal weights to the uncertainty perceived by the individual forecasters. This is problematic, since it does not factor in the historical different forecast accuracy. Therefore, this might result in a biased

forecast as well, by underweighting the forecasts with a high predictive accuracy and overweighting the underperforming forecasts.

To solve this problem, one could try to just take the forecasts of professional economists with a high reputation. However, Lamont (2002) found out, that as forecasters become older and more established the predictive accuracy of their forecasts actually significantly decreases. This phenomenon can be explained by the fact that they tend to produce more and more radical forecasts, which in turn is most of the times less accurate.

All of these problems make the approach of just using already produced macroeconomic forecasts by professional forecasters for the planning of the budget, product mix or production output of a company undesirable, especially considering that the maximal reliable forecast horizon of business-cycle forecasts is only a few months, according to Schreiber and Soldatenkova (2016).

2.4. Sales Forecasts Using Macroeconomic Indicators

Ali et al. (2009) group the different sales forecasting methods into three groups: judgmental, extrapolating and causal methods. Extrapolating methods analyse the time series and can be used to identify the seasonality or autocorrelations, however it is impossible to use them for the prediction of changes due to external factors. Currie and Rowley (2010) thus note, that this makes the accurate prediction of future sales using just historical data impossible in times of flux.

To counter this problem, judgmental adaptations are frequently used, however Sagaert et al. (2018) point out, that the managerial expert judgement “is well known to suffer from various biases, is expensive and not scalable” (Sagaert et al. (2018, p. 558)). The causal methods for sales forecasting are based on quantitative models using variables as input which are thought to describe the underlying dynamics and reasons for changes in sales volume. This could be promotions, business cycles or whatever seems to be describing the phenomena best.

One approach to deploying the causal forecasting method was described by Zhang et al. (2019). They created a sales forecast based on macroeconomic indicators and enhanced this forecast by using sentiment analysis on online reviews. They were able to show that their assumption that the state of the economy is the “driving force for sales in the medium to long term” (Zhang et al. (2019, p. 7)) was correct. When choosing the macroeconomic indicators, they adhered to two principles: “(1) There must be strong correlations between the selected macroeconomic indicator and the product sales volume and (2) there is no serious multicollinearity among the selected macroeconomic indicators” (Zhang et al. (2019, p. 3)).

Another interesting example is Gao et al. (2018). Here they used just four economic indicators for the prediction of Chinese automotive sales, namely the consumer confidence index, the consumer price index, the steel production and the gasoline price. They then used a monthly empirical analysis and used a vector error correction model to predict the sales. With a root mean square error of 0.1243 for a forecasting horizon of twelve months, this is a quite interesting result considering that only four economic indicators were used.

There is an example in the literature that uses even less economic indicators. Homareau (2015) predicted the sales of U.S. motor vehicles with the use of only the U.S. housing starts data. Although the results were quite promising, it should be pointed out that the author used this as an example to illustrate the construction and calibration of a sales forecast based on macroeconomic indicators and therefore tried to keep it simple by only using one indicator.

Sagaert et al. (2018) also used macroeconomic indicators for the construction of a sales forecast. Especially in the context of tactical sales forecasting a horizon of three to twelve months is relevant for the planning of resources, the financial planning and the adaptation of the product mix. In contrast to the previous examples a statistical framework is applied to filter out amongst ten thousands of possible indicators the ones with an appropriate leading time and correlation.

For this filtering of indicators, two different methods were applied. A pre filtering using the judgement of experts in the field of sales forecasting and the final selection by using the least absolute shrinkage and selection operator (LASSO) approach. When applied to a real case study the proposed approach by Sagaert et al. (2018) was able to generate 18.8 % accuracy gains compared to the forecasting process already in place at that company.

Thus it can be concluded that the causal method seems to be the most appropriate for sales forecasting in most settings across the different industries and the respective companies. In that context the use of macroeconomic indicators for medium to long range tactical sales forecasting seems to be the most suitable, with possible enhancement of the forecast using additional industry specific factors, for example promotions in the context of grocery sales.

2.5. Construction of Sales Forecasts

Lawrence et al. (2000) point out that according to surveys the most often used forecasting method remains management judgement, despite the fact that the same surveys reveal that the predictive accuracy is the most common criterion in the decision process for a forecasting strategy. This could be seen as an indication that the wide availability of forecasting software does not fulfil the needs of the managers' expectations.

Of course there is also the possibility of constructing a sales forecast without the use of a software especially designed for sales forecasting. Petropoulos et al. (2016) describe the construction of an extrapolating forecast as follows: "A typical forecasting process includes preprocessing and analysis of the data, which are usually in the form of time series, extrapolating the series with a suitable statistical method . . . , post-processing the statistical forecasts and monitoring and evaluating the outputs." (Petropoulos et al. (2016, p. 1))

However, as already shown in the previous chapters, the extrapolating method for construction of a sales forecast is often less suitable than the causal method. A description of a construction process of a causal forecast was done by Pareto (2013). He describes four distinct steps that have to be performed. The first step is to decide which phenomena are relevant for the development of the sales and what the resulting composite index should represent. The second step is the selection of the individual indicators which are believed to depict the dynamics of the underlying change process. However, not all indicators are suitable, since some of them are not accessible at the time they are needed. Another issue that is pointed out is the trade-off between choosing too little indicators and thus losing information or choosing too many indicators and thereby using indicators that depict mostly the same. Here it is valuable to study any possible cross-correlations between the indicators.

The third step Pareto (2013) describes is the normalization of the chosen indicators. This is important to make the data comparable, since the indicators can have different magnitudes and different measurement units. The fourth step is the aggregation of the normalized indicators into a composite index. This aggregation step can be done with different methods, ranging from additive methods to multivariate methods like Principal Component Analysis.

However, “forecasting multivariate time series using large numbers of predictors is an important and challenging problem” (Li and W. Chen (2014, p. 1011)). One of these problems is the question of how many indicators should be used. Elliott et al. (2013, p. 372) describe this problem as a “trade-off between model complexity and model fit”.

One approach to combat this problem is described by Stock and Watson (2002). They describe that when faced with a large amount of predictor time series, one can try to reduce them to a much smaller number of unobserved latent factors by modelling the covariability of the series and using linear regression to achieve a forecast. A similar approach was performed by Ali et al. (2009) who were able to show that data pooling is often improving the model performance. However, it should be pointed out that this method was very suitable for periods without promotions, but for the case of promotions being present, they showed that regression trees with explicit features were more suited to the problem.

An interesting counter point was described by Boivin and Ng (2006). They argued that the literature focuses on using data from large panels, since the “theory is developed for large N and T ” (Boivin and Ng (2006, p. 189)). Here, N denotes the number of factors or different time series and T denotes the number of time steps. However, they derived from simulations and empirical examples, that when using only 40 series the accuracy was not worse than when using 147 series.

This is in line with the findings of Sagaert et al. (2018). They demonstrated a method for the fully automatic selection of macroeconomic leading indicators from a very large set and compared this to the performance of pre-filtering the indicator set by experts. Here, the smaller data set resulting from the experts’ choice was improving the accuracy of the sales forecast.

Additionally, Sagaert et al. (2018) point out that the use of macroeconomic leading indicators has a lot of benefits. If the individual indicators would be predicted instead of aggregated with the individual time lag, this could introduce additional errors and would have computational disadvantages. The main benefit, however, is “that any hard-to-predict turning points and shocks captured in indicators are used as inputs to our model. If we were to predict the leading indicators this would not be possible.” (Sagaert et al. (2018, p. 562))

One drawback of using leading indicators is the constraint on the forecast horizon as only indicators with a time lag equal or greater than the forecast horizon should be used for the prediction of the sales. This is illustrated in figure 2.1. Here the first forecast with a forecast horizon of one month can use all the indicators, since they have a time lag of one, five and twelve months. The second forecast with a forecast horizon of 12 months can not incorporate the indicators with one or five months.

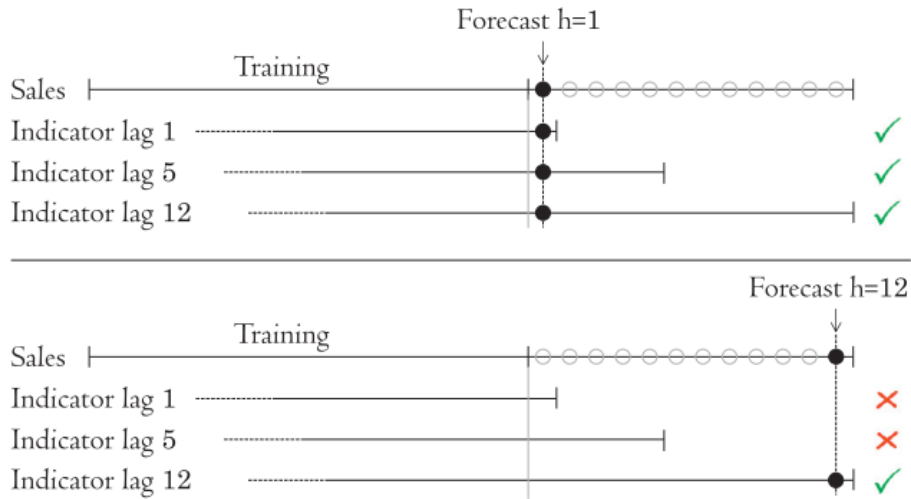


Figure 2.1.: Lagged Indicators (Sagaert et al. (2018))

Becker et al. (2015) state that the developer of a forecast faces several choices when constructing a composite index. These choices include the chosen theoretical framework, the selection of indicators, the normalization method and the aggregation method. Additionally, Bulligan et al. (2015) note that the accuracy of a forecast can be improved by carefully screening the used monthly indicators. Tarabusi and Guarini (2010) point out that the choice between using one composite index or a multitude of indicators is also something to consider carefully. On the one hand the composite index has the benefit of simplicity, while on the other hand a dashboard of indicators has the benefit of more information.

Finally, Pareto (2013) concludes that there are a lot of arbitrary choices that have to be made by the developer of a forecast, each of them having a significant effect on the predictive accuracy. However, there is no general best practice method, instead they note that the scientific community is beginning to agree “that there is not a composite

index universally valid for all areas of application, and, therefore, its validity depends on the strategic objectives of the research.” (Pareto (2013, p. 79)).

2.6. Use and Modification of Sales Forecasts

One aspect that is often overlooked is the subsequent use and adjustment of a sales forecast. It is not sufficient to just consider the statistical models used for the construction of a sales forecast if the resulting figures are adjusted greatly by the judgement of a sales manager. This consideration is not a new idea, already in 1987 it was found via a survey study by Dalrymple (1987) that most companies used subjective and extrapolating techniques. They found an improvement of predictive accuracy in companies using computers for the construction of sales forecasts, thus meaning also more sophisticated methods.

One would assume, that this means that nowadays all companies rely on computer-aided models employing sophisticated statistical methods and subsequently having reliable sales forecasts with high predictive accuracy. However, Leitner and Leopold-Wildburger (2011) note that a “multitude of surveys documents that judgement is a popular method for sales forecasting in companies.” (Leitner and Leopold-Wildburger (2011, p. 462)) They also note that judgemental methods tend to lead to limited predictive accuracy. This is not just due to the given reduced forecasting ability of humans compared to sophisticated statistical methods, but also because test-subjects in experiments tend to under-weight the statistical forecast and over-weight their judgement, even after it was shown to them that their judgemental accuracy is worse.

Similar to this, Sanders and Manrodt (2003) showed for different industries, firm sizes and product positioning strategies that companies which use quantitative methods for the forecast construction display significantly less forecast error rates than companies that rely on judgmental methods. This makes the findings of Petropoulos et al. (2016) surprising, as they concluded that many studies find that around 90 percent of forecasts in organisations are subject to judgemental interventions.

Klassen and Flores (2001) also found that most of the companies in the U.S. and Canada are using judgemental methods more often than causal, quantitative or other newer methods. Furthermore, they stated that “senior management revises the forecast frequently and believes that on average, accuracy is improved by the revision” (Klassen and Flores (2001, p. 163)). However, this improvement could not be shown, due to the high variation in the predictive accuracy of the resulting forecasts.

One of the reasons for the poor predictive accuracy of judgmental forecasts or judgmental adaptations to a statistical forecast is bias. Lawrence et al. (2000) examined thirteen manufacturing companies and found out that their judgemental sales forecasting were “not uniformly more accurate than a simple, un-seasonally adjusted, naive forecast” (Lawrence et al. (2000, p. 151)), which constitutes a very poor performance. The forecasters denied any bias, saying that they “try to be spot on” (Lawrence et al. (2000, p. 159)). However, when asking more closely they admit to bias without realising it by using phrases like “ ‘hedging a bet’ or ‘playing safe’ ” (Lawrence et al. (2000, p. 159)).

There are several reasons why forecasters tend to make judgmental adjustments to the sales forecast. Petropoulos et al. (2016) identify four distinct reasons. The first reason is that managers want to incorporate extraordinary events into the forecast like promotions, new products (either a product launch by the company or by a competitor), strikes and other events. This is in line with Fildes et al. (2009, p. 20), who noted that “forecast adjustment is the only practical way for most organisations to improve their incorporation of key drivers into their disaggregated sales forecasts”.

The second reason Petropoulos et al. (2016) list is that forecasters might adjust the statistical forecast simply to fit the budget or to adhere to politically choices decided by senior management. The third reason is that forecasters can see the statistical methods as a black box and in turn adjust the forecast judgementally due to missing confidence in the statistical forecast. The fourth reason for judgemental interventions can be the fact that “humans are liable to confuse the signal with the noise and introduce unnecessary judgemental adjustments as the result of perceived systematic changes that were not captured by the statistical methods” (Petropoulos et al. (2016, pp. 1–2)).

An example of this in a real world company was described by Hunschofsky et al. (2012). Here, the company used a dual approach, where the first step consisted of creating a forecast that implements the macroeconomic environment and as a second step this forecast was adjusted by judgement of top level management, customer feedback and internal know-how. Additionally, they used their macroeconomic forecast to determine which phase of the cycle should be anticipated and subsequently defined action plans for all parts of the company.

3. Objectives of GNOSIS

Prior to any design and development steps, it is necessary to define the objectives of the software tool based on research gaps and real life needs identified in the previous chapters. The tool itself will be called GNOSIS (short for “**G**enerating **N**ew **O**pportunities for **S**election of **I**ndicators for **S**ales Forecasting”).

The first section of this chapter will deal with an overview concerning the different requirements that GNOSIS should fulfil. The second section deals with the construction of the sales forecast and which methods the user of GNOSIS should be able to choose.

The third section will define the calibration methods GNOSIS should have implemented and the fourth section will define the validation methods necessary for predictive accuracy analysis. The final section will deal with the requirements regarding usability, as it is key, that actual users can construct, calibrate and validate sales forecasts with GNOSIS.

3.1. General Requirements

To formulate the problems which are supposed to be solved by GNOSIS, the foremost step is to understand the system with regard to the problems arising from such a system and how these problems could be solved. GNOSIS is specified as a predictive accuracy analysis tool for sales forecasting utilising business cycles. First and foremost, this system should be discussed.

The relevance of business cycles for sales budgeting and planning is discussed in chapter 2.1. Chapter 2.4 described the benefits of macroeconomic indicators, containing information about business cycles, for the construction of sales forecasts. For this reason, GNOSIS will use macroeconomic indicators as an input variable. The selection of the necessary indicator data has to be made by the user.

The user will be modelled as an expert and we use the available expert knowledge of the user to load the essential and required indicators from external databases. This is consistent with chapter 2.5, where it was noted that pre-filtering by experts' judgement can improve the predictive accuracy of a sales forecast. As these indicators are mostly published once a month, GNOSIS should perform the calculations through monthly data. This has to be considered by the user when selecting the indicator. As a further requirement of GNOSIS, the indicator data must be loadable by the user in a predefined data format as an input to the program.

The next point GNOSIS must fulfil is sales forecasting. This sales forecast created by GNOSIS serves the analyst to predict future sales developments and thus adaptively intervene in the planning. In order to generate a sales forecast as an output, the input data of historical sales and historical macroeconomic indicators must be loaded. After the sales forecast has been calculated the forecast data has to be made available to the user. This should be in the form of a report and in the form of a downloadable Microsoft Excel spreadsheet containing all relevant data. The output of the sales forecast should be created in such a way that it can be used by a subsequent forecasting program, if such a program is in use or should be developed.

After the generation of a forecast this forecast has to be validated. Thus, another prob-

lem is the implementation of the predictive accuracy analysis. GNOSIS must enable in-sample validation and out-of-sample validation. These tests are needed to check how the forecast has been mapped to the real historical sales data. In addition to the validation methods that validate the forecast, univariate forecasts should also be implemented using widely known and established methods found in the literature enabling the user to benchmark the generated forecast and validate it against the benchmark forecasts.

As a last point, GNOSIS should not autonomously create a sales forecast in order to prevent the users from seeing the construction of the forecast as a black box and subsequently adjusting the forecast after its construction judgementally, as described in chapter 2.5 and chapter 2.6. The user must be able to intervene in the calculation at any time and be able to vary various parameters. Amongst other things these parameters should be the selection of a calibration period, the selection of indicator-specific properties, the selection of validation methods, etc. This ensures that expert knowledge is brought into the sales forecast in order to optimize it as described in chapter 2.5.

3.2. Construction Methods

In this section the different construction methods that are required of GNOSIS and must be available to the user are described.

Forecast Horizon - Time Lag

As already mentioned in chapter 2.5, the forecast horizon is a key parameter for the construction of a sales forecast using lagged indicators. This is due to the fact that only indicators with a time lag greater or equal to the forecast horizon can be used for the construction of the sales forecast. Thus, this parameter has to be settable by the user and the forecast horizon should range from 1 month to at least 12 months.

Rate of Change (ROC)

The use of the rate of change (ROC) for cost forecasting was already discussed in chapter 2.2. An example for applying this concept to sales forecasting would be Hunschofsky et al. (2012). Using the ROC for the construction of a sales forecast has the advantage of eliminating the need for seasonality dummy variables, thus making the model simpler and more easy to understand for the user. Additionally, there are different types of the ROC.

One type would be the $ROC_{1/12}$, which is analogue to the calculation of yearly returns in the context of stock market returns. The $ROC_{1/12}$ of the sales can tell the user the percentage change of the sales of one month compared to the sales in the same month of the previous year. The calculation of the $ROC_{1/12}$ is shown in formula 3.1, where t denotes the month, j denotes the time series that is being analysed with the ROC (for example the sales or an indicator) and TS is a matrix containing the different values for the different indicators or sales and for the different months available in the data. The $ROC_{1/12}$ is given in percent.

$$ROC_{1/12,t,j} = \left[\frac{TS_{t,j}}{TS_{(t-12),j}} - 1 \right] * 100\% \quad (3.1)$$

Another type would be the $ROC_{3/12}$, which is calculated by aggregating the last three monthly values and dividing them by the aggregated value of the same months of the previous year. The $ROC_{3/12}$ for the months January, April, July and October are broadly used in companies, however it is not usually called $ROC_{3/12, April}$ but quarterly percent changes of the first quarter. The calculation of the $ROC_{3/12}$ is shown in formula 3.2 and is calculated analogue to the $ROC_{1/12}$, but with the addition of aggregating the values of three consecutive months by using a sum.

$$ROC_{3/12,t,j} = \left[\frac{\sum_{i=t}^{t-2} TS_{i,j}}{\sum_{i=(t-12)}^{t-14} TS_{i,j}} - 1 \right] * 100\% \quad (3.2)$$

The $ROC_{12/12}$ is another type, which is calculated by summing up all the values from a whole year and dividing them by the values of the previous year. This has the advantage of containing all the months and thus all the seasonal differences and therefore smoothing out the data. The calculation of the $ROC_{12/12}$ is shown in formula 3.3 and is analogue to the calculation of the $ROC_{3/12}$, however, aggregating the values over 12 months instead of just 3.

$$ROC_{12/12,t,j} = \left[\frac{\sum_{i=t}^{t-11} TS_{i,j}}{\sum_{i=(t-12)}^{t-23} TS_{i,j}} - 1 \right] * 100\% \quad (3.3)$$

There are of course also different possibilities like a $ROC_{10/12}$ or $ROC_{2/12}$, but due to their missing equivalent in the typical planning processes of companies, a necessity to implement those does not seem warranted.

Correlation

The user should have different choices regarding the chosen correlation method. Most often this will be the Pearson's correlation coefficient, but there could also be a need for the use of the Spearman's rho or the Kendall's tau. Additionally, the beta factor described in the Capital Asset Pricing Model could also be an interesting choice.

There are different measures for the correlation of variables. The most often used measure of dependence is the Pearson's correlation coefficient according to Sukcharoen and Leatham (2016). The Pearson's correlation coefficient examines linear correlation. It can be calculated with formula 3.4, where X denotes the first variable and Y denotes the second one. Cov denotes the covariance and Var the variance.

$$\rho = \frac{Cov(X, Y)}{\sqrt{Var(X) * Var(Y)}} \quad (3.4)$$

The Spearman's rho is a non-linear measure that examines the dependence of the rank correlation (Sukcharoen and Leatham (2016)). In other words, it measures the directional correlation between two factors, by inserting the rank value of X and Y into the formula of the Pearson's correlation coefficient, as shown in formula 3.5.

$$\rho_s = \frac{Cov(Rank(X), Rank(Y))}{\sqrt{Var(Rank(X)) * Var(Rank(Y))}} \quad (3.5)$$

The Kendall's tau is non-linear concordance measure that “measures the difference between the probability of concordance and probability of discordance” Sukcharoen and Leatham (2016, p. 12). And it can be expressed according to Sukcharoen and Leatham (2016) with formula 3.6, where X and Y are ordered statistics.

$$\tau = Pr((X_1 - X_2) * (Y_1 - Y_2) > 0) - Pr((X_1 - X_2)(Y_1 - Y_2) < 0) \quad (3.6)$$

Another measure for the dependence of variables compared to one specific variable is the β found in the Capital Asset Pricing Model. This beta is normally applied to measure the dependence of the value changes of a stock relative to the value changes of the market portfolio and can be calculated with formula 3.7 according to Fama and French (2003). Here R_i denotes the returns of an individual stock and R_M the market returns.

$$\beta_{i,M} = \frac{Cov(R_i, R_M)}{Var(R_M)} \quad (3.7)$$

The formula 3.7 can be adapted for the construction of a sales forecast by changing R_i from the returns of a stock to the ROC of an indicator and using the ROC of the Sales instead of R_M .

Cross-Correlation

The importance of examining the cross-correlation (also called collinearity) of indicators or factors used to construct a composite index was already discussed in chapter 2.5. It is important to eliminate variables that are highly cross-correlated in order to make the assigned weights stable, when using a method like LASSO or stepwise regression. (Stock and Watson (2012)). Also, Mol et al. (2008) explain, that it is beneficial to exclude the highly cross-correlated variables, since the remaining variables “should capture the essence of the covariation of the data” (Mol et al. (2008, p. 322)).

The calculation of the cross-correlation can be performed by pairwise calculation of the Pearson’s correlation coefficient and determining a value above which the variable is to be excluded. However, this value must be adjustable by the user, since it can vary from case to case what constitutes a too high cross-correlation.

Composite Index (CI) Construction

The construction of a Composite Index using a set of indicators can be done with a broad variety of methods. One of those methods that should be available to the user consists of assigning a weighting factor to the different time lagged indicators. The approach called weighted arithmetic mean (WAM) seems to be most suited (Tarabusi and Guarini (2010)). Another method that should be selectable by the user is the “least absolute shrinkage and selection operator” (LASSO) which is a penalized regression method for selection and weighting of for instance indicators (Tibshirani (1996) and Sagaert et al. (2018)). The last method that should be selectable by the user is stepwise regression, as this is a widely used method according to Sagaert et al. (2018).

Using Weighting Factors

The framework for using weighting factors for the construction of a composite index that should be implemented is the WAM. This method basically assigns the weights in a way that the sum of all the indicators’ weights equals one.

In case of an equal weight assigned to each indicator that is selected, which is a very commonly used method according to Becker et al. (2015) and Pareto (2013), this becomes the arithmetic mean of all the indicators and the calculation of the individual weighting factor w_j can be seen in formula 3.8, where n is equal to the number of included indicators. This method should be available to the user.

$$w_j = \frac{1}{n} \quad (3.8)$$

However, this method is often intended to make the impact of all the indicators on the composite index equal. In general it can be said that the weights are often falsely seen as a direct measure of importance. However, Becker et al. (2015) point out that the nominal weights are often only by coincidence also the importance of the indicator and one has to consider the correlations of the indicators to the value that is supposed to be forecasted. This can lead to improved forecasts by assigning weights to the data by their properties (Boivin and Ng (2006)). Hence, the user should be able to select a method, where higher weights are assigned to indicators with a higher correlation. An approach to realize this can be seen in formula 3.9.

$$w_j = \frac{Corr_j}{\sum_{i=1}^n Corr_i} \quad (3.9)$$

An approach which increases the difference in assigned weights relative to the correlation coefficient was described by Hunschofsky et al. (2012) and can be seen in formula 3.10.

$$w_j = \frac{\max\{(Corr_j - 0.5), 0\}}{\sum_{i=1}^n \max\{(Corr_i - 0.5), 0\}} \quad (3.10)$$

The median of the values of the indicator's can be seen as a special form of weighting outside the framework of the WAM. Here, the composite index consists of just one indicator at one point in time, with the indicator representing the median getting assigned a weighting factor of one and the indicators not representing the median getting assigned a weighting factor of zero. These weights must then be calculated for each point in time.

Using LASSO

One of the benefits of LASSO is that “it tends to produce some coefficients that are exactly 0 and hence gives interpretable models” (Tibshirani (1996, p. 267)). Especially for dealing with large sets of indicators LASSO can be helpful since it automatically deals with variable selection. However, one of the shortcomings of LASSO is that “once another observation has been added into the estimation window, the estimated regression coefficients and the subset of important predictors may change dramatically” (Li and W. Chen (2014, p. 997)).

The mathematical foundation of LASSO is the following cost function (according to Sagaert et al. (2018)) shown in formula 3.11, where y_i denotes the sales values, x_{ip} are the values of an individual indicator, β_0 denotes the intercept term, β_p denotes the indicator’s specific regression coefficient and λ denotes the penalty which is responsible for the shrinkage. The goal of LASSO is to minimise this cost function.

$$\sum_{i=1}^n \left(y_i - \beta_0 - \sum_{p=1}^p \beta_p x_{ip} \right)^2 + \lambda \sum_{p=1}^p |\beta_p| \quad (3.11)$$

Using Stepwise Regression

As already noted, the stepwise regression should also be a selectable construction method for the composite index. This is due to the fact that it is one of the most commonly used regression methods for large sets of factors. The stepwise regression is a mix of backwards and forwards selection. While backwards selection uses all explanatory variables in the beginning and calculates the test statistic for individually excluding them, forward selection begins with zero explanatory variables and then adds the variables one by one, again calculating the test statistic. The stepwise regression is an iterative process performing both forward selection and backward selection, starting with the forward selection.

However, as G. Smith (2018) notes, the stepwise regression is actually not very well suited for big data sets as it may not select explanatory variables with causal relationships and instead weight nuisance factors too high. This is especially a critic on stepwise regression

using the coefficient of determination r^2 as the test statistic which is the determining factor whether a variable should be included or excluded from the model.

Another approach would be the Bayesian Information Criterion (BIC) and the Akaike Information Criterion (AIC) for determination whether a variable is included into the model. According to Burnham and Anderson (2004) the AIC has a target model that is specific to the sample size and minimizes the expected information loss whereas the BIC assumes that there is a true model independent from the sample size and has this true model as the target model. Hence, the stepwise regression should be performed with the AIC which is also in line with findings of Hurvich and Tsai (1991) who noted that the AIC produces good model selection.

Deriving Sales Forecast from CI

Since the composite index (CI) is calculated using the ROC, it is necessary to derive the sales values from the CI. The calculation of the sales derived from the $ROC_{1/12}$, $ROC_{3/12}$ and $ROC_{12/12}$ all differ.

First, the sales forecast from the CI using the $ROC_{1/12}$ will be derived. For a future point in time with a distance h from now, the ROC formula can be seen in formula 3.12, which is adapted from formula 3.1. It should be noted that due to the constraint on the time lag of the indicators, mentioned in chapter 2.5, the time horizon h into the future cannot exceed the forecast horizon.

$$ROC_{1/12,(t+h),Sales} = \left[\frac{TS_{(t+h),Sales}}{TS_{(t-12+h),Sales}} - 1 \right] * 100\% \quad (3.12)$$

Formula 3.13 shows the relationship between the CI and the $ROC_{1/12}$. Here it is shown, that the CI is the predicted value of the $ROC_{1/12}$.

$$CI_{1/12,(t+h)} = \widehat{ROC}_{1/12,(t+h),Sales} \quad (3.13)$$

Therefore it is possible to rearrange the formula 3.12 to calculate the sales value for a

future point in time, which is shown in formula 3.14.

$$\widehat{TS}_{t+h,Sales} = \left[\frac{\widehat{ROC}_{1/12,(t+h),Sales}}{100\%} + 1 \right] * TS_{(t-12+h),Sales} \quad (3.14)$$

Analogue to the formula for the $ROC_{1/12}$ for a future point in time, the formula for the $ROC_{3/12}$ can be adapted from formula 3.2 and is shown in formula 3.15.

$$ROC_{3/12,t+h,Sales} = \left[\frac{\sum_{i=t+h}^{t-2+h} TS_{i,Sales}}{\sum_{i=(t-12+h)}^{t-14+h} TS_{i,Sales}} - 1 \right] * 100\% \quad (3.15)$$

Similarly the CI is now the prediction for the $ROC_{3/12}$ at a future point in time and this is shown in formula 3.16.

$$CI_{3/12,(t+h)} = \widehat{ROC}_{3/12,(t+h),Sales} \quad (3.16)$$

Solving formula 3.15 with regard to the predicted $ROC_{3/12}$ to get the predicted sales value results in formula 3.17. It should be pointed out that in contrast to formula 3.14 for the $ROC_{1/12}$, now there is a subtractive term, which is necessary due to the fact that the sum of the current (or now future) sales values has to be split up. Additionally it should be noted that for the $ROC_{3/12}$ all the values between time point t and $t+h$ have to be calculated prior to calculating the value at $t+h$. This can be explained by looking at the subtractive term since those future sales values have to be inserted into the formula.

$$\widehat{TS}_{t+h,Sales} = \left[\frac{\widehat{ROC}_{3/12,(t+h),Sales}}{100\%} + 1 \right] * \sum_{i=t-12+h}^{t-14+h} TS_{i,Sales} - \sum_{i=t-1+h}^{t-2+h} TS_{i,Sales} \quad (3.17)$$

Since the calculation of the sales values derived from the CI calculated by using the $ROC_{12/12}$ is analogue to the derivation shown for the sales values derived from the CI calculated by using the $ROC_{3/12}$, the formulas are not explained in this chapter for brevity reasons. They can be found in appendix A.

Logistic Regression

Another method for constructing a forecast simply based on probabilities is the logistic regression. Due to the constraint of the forecast horizon by the time lag of the indicators, the problem could arise that the forecast horizon can not be set as high as the user needs it to be while keeping a high predictive accuracy. Therefore logistic regression can be used to exceed this limited forecast horizon by performing a univariate analysis of past ROCs.

The logistic regression requires a binomial distribution. Therefore events can be defined and their occurrence can be set to 1 while their missing occurrence can be set to 0. For example, the development of the ROC in past starting with a specific value x at time t , can be monitored for m months, and whenever it exceeded a value y in those h months, the event would be defined as having occurred.

With this approach it is possible to generate probabilities for the ROC exceeding or falling below a certain value during m months exceeding the forecast horizon. It should be noted that this comes with the problems of univariate forecasting methods already described in chapter 2.2, chapter 2.4 and chapter 2.5. Despite those shortcomings, it can be useful to some users and GNOSIS should therefore implement this method.

3.3. Calibration Methods

In this section the different calibration methods that are required of GNOSIS are described. These calibration methods must be available to the user of GNOSIS.

Selection of Indicators for ROC Calculation

One of the problems facing the construction of a sales forecast using indicators is the nature of those indicators. Since it was defined in chapter 3.2 that the sales forecast should be constructed using the ROC, the indicators have to be carefully calibrated. Some of the indicators already consist of rate of change values, thus it must be possible to exclude such indicators from the ROC calculation and insert them directly into the subsequent calculations.

In-sample Period

The in-sample period is one of the most important calibration parameters. Especially considering the already mentioned shortcomings of LASSO and stepwise regression where one extra observation can alter the weights of the individual indicators dramatically.

When setting the in-sample period the calculation of the ROC must be considered since the construction of the $ROC_{12/12}$ needs at least 24 months. Additionally, the forecast horizon must be considered as well since a time lag of h months increases the necessary in-sample period by h months.

Lastly, the user should be able to choose an in-sample period for the calibration that is smaller than the available data set, to enable backtesting.

Data Normalisation

Data normalisation is an important step in the calibration process. Its impact is not visible during the analysis of the correlation, since the most commonly used methods are

linear transformations and thus do not affect the correlation coefficients (Pareto (2013)) described in chapter 3.2.

One of the normalisation methods proposed by Pareto (2013) is the mean-variance normalisation shown in formula 3.18. Here i denotes the respective indicator and j denotes the index of the specific indicator's value. The expression $E(X_i)$ denotes the mean of the indicator's values and the expression $V(X_i)$ denotes the respective variance. It should be noted that this approach would center the indicators' values around zero and produce a standard deviation of 1 for all indicators. However, the indicators are supposed to represent the ROC values of the sales and thus this formula will have to be adapted.

$$x_{i,j} = \frac{X_{i,j} - E(X_i)}{\sqrt{V(X_i)}} \quad (3.18)$$

Another normalisation method mentioned by Pareto (2013) is the “min-max normalisation method” shown in formula 3.19. This would normalise the indicators to be between 0 and 1 and thus will also have to be adapted for the use in GNOSIS.

$$x_{i,j} = \frac{X_{i,j} - X_{min,i}}{X_{max,i} - X_{min,i}} \quad (3.19)$$

These methods should be changed in a way to enable the user to choose a method of normalisation of the amplitude and choose whether the user wants to set the mean of the indicators' values to the mean of the sales values.

Indicator Correlations

As described in chapter 3.2, the choice between the different correlation methods can also be seen as a calibration step, if the correlation method is picked according to the nature of the indicator.

The maximal allowed value for the cross-correlations between the indicators is an important step during the calibration. Picking a value that is too low could result in too few indicators remaining for the construction of a sales forecast. Picking a value that

is too high could result in a composite index constructed with indicators that mostly represent only one of the phenomena driving the sales development. However, defining a maximal value for all possible indicators and industries is not possible, and thus this has to be done by the user during the calibration process.

Composite Index Construction

When constructing the composite index there should be the methods available to the user already described in chapter 3.2. Additionally, it should be possible for the user to manually aggregate the indicators by setting the weights without the use of a method. This is then a manual calibration of the composite index.

Another calibration parameter that should be settable by the user is the number of indicators used by the composite index. This should range from only one indicator to the maximal possible set size that is possible depending on the constraints, i.e. possible elimination of highly cross-correlated indicators.

3.4. Validation Methods

This section describes the different validation methods that should be implemented in GNOSIS. These include different in-sample tests, out-of-sample tests, and benchmarking validation methods.

In-sample Tests

To test the fit model to the data an in-sample testing must be performed. One popular measure for in-sample testing is the coefficient of determination (R^2) used for example by Zhang et al. (2019). Another popular measure is the root mean squared error (RMSE) used by Guo et al. (2013). Therefore, both of these testing methods should be implemented and available to the user for validation of the fit of the model.

Coefficient of Determination (R^2)

The coefficient of determination can be calculated with formula 3.20. TS_i denotes the realized sales values at time point i , while \overline{TS} denotes the arithmetic mean of the realized sales values and \widehat{TS}_i denotes the predicted sales values at time point i which is the composite index at time point i .

$$R^2 = \frac{\sum(\widehat{TS}_i - \overline{TS})^2}{\sum(TS_i - \overline{TS})^2} \quad (3.20)$$

Root Mean Squared Error (RMSE)

The root mean squared error can be calculated with formula 3.21. Here, the notation is similar to formula 3.20 with t denoting the last month included in the in-sample period.

$$RMSE = \sqrt{\frac{\sum_{i=1}^t (\widehat{TS}_i - TS_i)^2}{t}} \quad (3.21)$$

Out-of-sample Tests

Out-of-sample testing is only possible in the context of backtesting where the forecasted values are already known and the forecast's predictive accuracy can be measured directly. Here, a multitude of methods exist and it is necessary to include more than one accuracy measure to avoid a possible bias (Guo et al. (2013)).

A very broadly used accuracy measure is the mean absolute percentage error (MAPE) which is for example used by H. L. Chen (2008), Guo et al. (2013), Homareau (2015), Sagaert et al. (2018), and Sanders and Manrodt (2003). Due to the broad application of this accuracy measure it must be available to the user. Other accuracy measures which are similar to the MAPE, but react differently to outliers are the mean absolute scaled error (MASE) and the geometric mean relative absolute error (GMRAE), for example used by Sagaert et al. (2018).

A different type of accuracy measurement is the mean directional accuracy which measures the relationship of the forecasts direction to the direction of the real sales values (for instance a positive slope). One example that has applied this accuracy measure would be Malone et al. (2016).

Mean Absolute Percentage Error (MAPE)

The formula for the calculation of the MAPE can be seen in formula 3.22. t denotes the last point in time of the in-sample period, fh denotes forecast horizon, TS_i denotes the values of the time series of the actual sales values, and \widehat{TS}_i denotes the predicted sales values.

$$MAPE = \frac{100\%}{fh} * \sum_{i=t+1}^{t+fh} \left| \frac{TS_i - \widehat{TS}_i}{TS_i} \right| \quad (3.22)$$

Mean Directional Accuracy (MDA)

The MDA can be calculated according to formula 3.23. Here $\mathbf{1}$ is the indicator function, which assigns the value 1 if the statement is true and the value 0 if it is false. $\text{sign}()$

denotes the signum function and the other notations are analogue to formula 3.22.

$$MDA = \frac{1}{fh} * \sum_{i=t+2}^{t+fh} \mathbf{1}_{\text{sign}(TS_i - TS_{i-1}) \equiv \text{sign}(\widehat{TS}_i - TS_{i-1})} \quad (3.23)$$

Mean Absolute Scaled Error (MASE)

The MASE can be calculated with formula 3.24 and the notations are analogue to formula 3.22.

$$MASE = \frac{(fh - 1) * \sum_{i=t+1}^{t+fh} |TS_i - \widehat{TS}_i|}{fh * \sum_{i=t+2}^{t+fh} |TS_i - TS_{i-1}|} \quad (3.24)$$

Geometric Mean Relative Absolute Error (GMRAE)

Formula 3.25 shows the calculation of the GMRAE and the notations are analogue to formula 3.22. The term \widehat{BTS}_i denotes the values of a benchmark forecast, for example this could be the naïve forecast or the random walk forecast.

$$GMRAE = \sqrt[fh]{\prod_{i=t+1}^{t+fh} \left| \frac{TS_i - \widehat{TS}_i}{TS_i - \widehat{BTS}_i} \right|} \quad (3.25)$$

Benchmarking

Whenever a forecast is constructed without available realisations for backtesting, the only commonly used out-of-sample testing method is benchmarking. Here the forecast is compared to benchmark forecasts. It should be noted that benchmarking can also be applied as a backtesting method.

A multitude of benchmark forecasts is available, however, it is reasonable to implement such benchmark forecasts which are broadly used in the literature. The naïve forecast and the seasonal naïve forecast are the most simple univariate methods (Sagaert et al. (2018)) and thus can be used to test the forecast for its predictive accuracy against

typically quite badly performing forecasts. Another univariate method would be the random walk with drift, which is very similar to the naïve forecast.

A more sophisticated univariate method is the auto-regressive integrated moving average model (ARIMA), which is also used as a benchmark forecast by Sagaert et al. (2018).

Naïve Forecast

The naïve forecast is a forecast generated by assuming that the current period's value is also the next period's value. This is also shown in formula 3.26, where \widehat{TS}_{i+1} denotes the prediction of the next period's value and TS_i denotes the actual value for the current period.

$$\widehat{TS}_{i+1} = TS_i \tag{3.26}$$

Seasonal Naïve Forecast

The seasonal naïve forecast is similar to the naïve forecast, however it assumes a seasonal pattern in the data and therefore does not take the current period's value for the prediction of the next period's value, but a past value that has a time shift of s months where s denotes the seasonality period. This is also shown in formula 3.27 with the other notations being analogue to formula 3.26. For most cases in sales forecasting a seasonality of 12 months is assumed and should therefore also be assumed for the implementation into GNOSIS.

$$\widehat{TS}_{i+1} = TS_{i+1-s} \tag{3.27}$$

Random Walk Forecast

The random walk forecast is similar to the naïve forecast with the difference of the possibility of a drift which gives the random walk a monotonic upwards or downwards trend. Additionally, it would be possible to include a random error term to create error boundaries when constructing multiple random walk forecasts, however, this does not

seem beneficial to visual comparisons. Therefore a random walk forecast should be implemented as a benchmark forecast without an error term but with a drift. This can be achieved by applying formula 3.28. Here, d denotes the constant drift factor and the other terms are analogue to formula 3.26.

$$\widehat{TS}_{i+1} = TS_i + d \quad (3.28)$$

ARIMA Forecast

The ARIMA (“Autoregressive Integrated Moving Average”) model for the construction of a forecast has an autoregressive term, a differencing term, and a moving average term. These three terms have to be estimated, in order to construct a forecast using ARIMA. However, this is a rather difficult step, hence GNOSIS should use an auto ARIMA function, in order to spare the user from constructing the ARIMA manually. The basic mathematical description of the ARIMA model analogue to Swain et al. (2018) is shown in formula 3.29. Here μ denotes the mean, ϕ denotes the p autoregressive terms and θ denotes the q moving average terms. The other notations are analogue to formula 3.26.

$$\widehat{TS}_{i+1} = \mu + \phi_1 * TS_i + \dots + \phi_p * TS_{i+1-p} - \theta_1 * TS_i - \dots - \theta_q * TS_{i+1-q} \quad (3.29)$$

3.5. Usability

As Klassen and Flores (2001) note, the use of sophisticated quantitative forecasting methods in real companies is very scarce. Sanders and Manrodt (2003) point out that the reason for this lack of use can be explained by the fact that many practitioners are dissatisfied with the current forecasting software.

According to Gumussoy (2016), the usability has to be considered during all stages of design of a software, however, it is often neglected, especially in early stages of the software design and development. The usability itself can be defined as “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specific context of use”(ISO 9241-11 (1993, p. 3)).

As pointed out by Marcus (2005), usability is important because it can help to prevent user errors, it requires less training for the user and it can increase the productivity of the user.

Therefore, it is a key requirement for GNOSIS to exhibit a very high usability. It is not sufficient to implement different forecasting construction, calibration, and validation methods, if those are not used by practitioners due to lacking usability.

4. Design and Development of GNOSIS

The following sections describe the implementation of the design and the development of GNOSIS. First the methodology is explained which serves as the basis for the development of the software tool. Then the overall structure of GNOSIS is described as a guideline for the design and development.

Then the program R and its package Shiny are described and it is discussed which advantages R offers. Based on the section “Decision for R” specific properties of R and Shiny will be demonstrated and how these properties can be used to develop an application. The section “Functions” explains how functions are written in R.

In the section “User Interface”, the Shiny application’s graphical user interface (UI) is explained. This discussion will consider on the one hand how the UI is programmed and on the other hand how the user can interact with it. The section “Server” describes how Shiny uses functions and the UI code to generate GNOSIS.

4.1. Methodology

Since the scope of this thesis is to develop a tool for the analysis of the predictive accuracy of sales forecasts based on business-cycles a methodology has to be found that is suitable for this type of problem. Many methodologies broadly used in business or finance research are not suited as they are not directly applicable to the design of a software tool.

There are different possible methodologies in the field of information science, one especially developed for the design of software tools, according to Peffers et al. (2007), is the “Design Science Research Methodology” (DSRM). One of the most comprehensive descriptions of the DSRM can be found in figure 4.1 created by Geerts (2011).

DSRM activities	Activity description	Knowledge base
Problem identification and motivation	<i>What is the problem?</i> Define the research problem and justify the value of a solution.	Understand the problem's relevance and its current solutions and their weaknesses.
Define the objectives of a solution	<i>How should the problem be solved?</i> In addition to general objectives such as feasibility and performance, what are the specific criteria that a solution for the problem defined in step one should meet?	Knowledge of what is possible and what is feasible. Knowledge of methods, technologies, and theories that can help with defining the objectives.
Design and development	<i>Create an artifact that solves the problem.</i> Create constructs, models, methods, or instantiations in which a research contribution is embedded.	Application of methods, technologies, and theories to create an artifact that solves the problem.
Demonstration	<i>Demonstrate the use of the artifact.</i> Prove that the artifact works by solving one or more instances of the problem.	Knowledge of how to use the artifact to solve the problem.
Evaluation	<i>How well does the artifact work?</i> Observe and measure how well the artifact supports a solution to the problem by comparing the objectives with observed results.	Knowledge of relevant metrics and evaluation techniques.
Communication	Communicate the problem, its solution, and the utility, novelty, and effectiveness of the solution to researchers and other relevant audiences.	Knowledge of the disciplinary culture.

Figure 4.1.: Design Science Research Methodology (Geerts (2011))

The DSRM is chosen for the creation of GNOSIS and its communication in this thesis due to its frequent application in software engineering. Additionally, its iterative nature seems very well suited to the process of solving the problem of the creation of GNOSIS.

However, the iterations of the DSRM are kept very general and the design and development is not further detailed. Due to this a methodology for specifically the step of the design and development of the DSRM is required. According to Awad (2005) there are so called “heavyweight methodologies” and “lightweight methodologies” in use.

The heavyweight methodologies rely on sophisticated and long planning prior to the generation of a software. However, the lightweight methodologies which are also called agile software development methods rely on customer feedback and include the customer into the development process.

Since it is required of GNOSIS to exhibit a high level of usability the satisfaction of the user is key. Therefore, the lightweight methodology of “Extreme Programming” (XP) is chosen. For brevity reasons, the different lightweight methodologies and the reasons for choosing XP are not further elaborated.

A small list of XP practices should suffice to explain its main characteristics. Awad (2005) lists, among others, these practices:

- development in small, frequently updated versions,
- definition of system between customer and programmers,
- the simplest possible solution is preferred,
- all code is written by two programmers on one computer,
- continuous integration and testing of new code, and
- customers must be available at all times for the developers.

Although it is not possible to adhere to the last point, a manager responsible for sales forecasting in a real company will be asked for frequent feedback during the design and development of GNOSIS.

4.2. Structure of GNOSIS

In order to describe GNOSIS a method must be found which makes it possible to abstract applications. Such a method can be found in the Unified Modeling LanguageTM (UML). The UML is an industry-wide modeling language which has also been standardized in *ISO 19505-2* (2012). UML can abstract a model of a software and describe models independently of the underlying methodology.

As already mentioned, the UML can be used for modelling applications. For this reason, a distinction must be made as to whether the entire system or only one subsystem should be displayed in a diagram. In order to enable a high abstraction of the application the UML introduces the Use Case Diagram (Klimek and Szwed (2010) and Object Management Group (2017)). A Use Case Diagram provides a general overview of a complete system. Here, the functionality is shown from the perspective of the user.

The user is referred to as an actor in this case. Every relationship between an actor and a use case usually needs a further definition of interfaces. The use case diagram should only be a graphical representation of use cases and their relationship between each other and with the environment. Subsequently, the necessary symbols of the UML are shown in figure 4.2 and it is further described how they can be used for modelling a software.

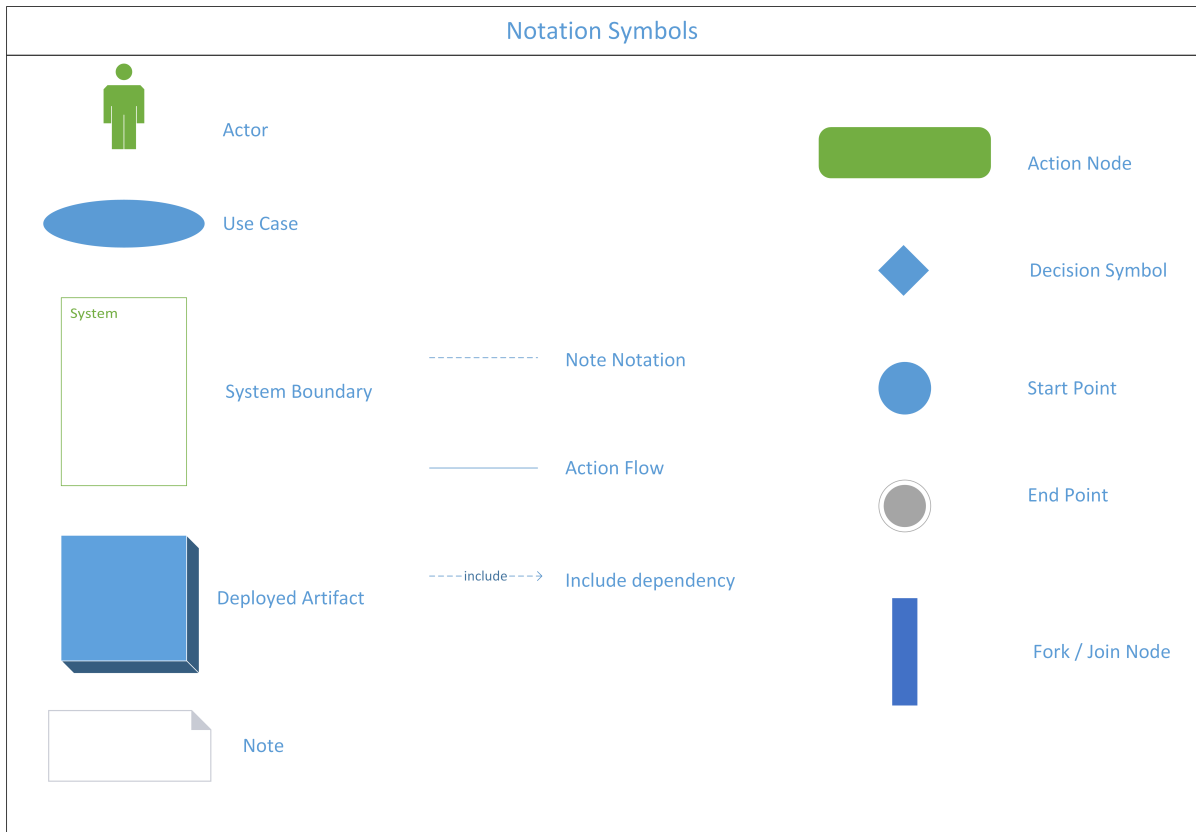


Figure 4.2.: Unified Modelling Language

The symbols that are used for a use case diagram can be seen in the left column of figure 4.2. The first symbol is used to describe the actor. Each use case is initiated by an actor. An actor can be either be a natural person, an institution, or an external system. The next symbol is the Use Case. This specifies a set of functions that a system performs and has a result. These use cases are mostly described by activity diagrams.

The next symbol describes the System Boundary. Within this limit the system carries out the actions of the Use Case. Actors interact with the system across the system boundary but are not encompassed within it. The deployed artefact icon is used to model third-party systems. The last icon of the first row shows notes. These notes are used to make charts more readable and understandable. (Object Management Group (2017) and Ambler (2004))

The middle column of figure 4.2 shows different connector symbols. The first symbol is used to combine notes with a use case. The second symbol, Action Flow, serves to

symbolize that one action passes into another. The last icon, Include Dependency, shows that one use case is needed by another use case and is included within another use case. (Object Management Group (2017))

After the Use Case Diagram has been created, the respective Use Cases are represented in Activity Diagrams. An activity diagram is intended to be a visual representation of a series of actions or to model the control flow in a system. Activities can be in serial or in parallel order. In general, an activity diagram has a start and an end point as shown in the last row of figure 4.2.

Figure 4.2 also shows an Action Node. This activity represents an uninterruptible action of an object which is completed before a subsequent activity can be started. The decision symbol makes it possible to symbolize decisions with different alternative paths. This is e.g. important if a decision has to be made before a subsequent activity. The last icon shows a fork or join node. This serves to divide an incoming activity flow into several flows or to reunite several flows. (Object Management Group (2017) and Dumas and ter Hofstede (2001))

Based on the previously defined system in combination with the UML notation, figure 4.3 shows the use case diagram of GNOSIS. It shows the necessary actors, the different use cases and how they interact with each other.

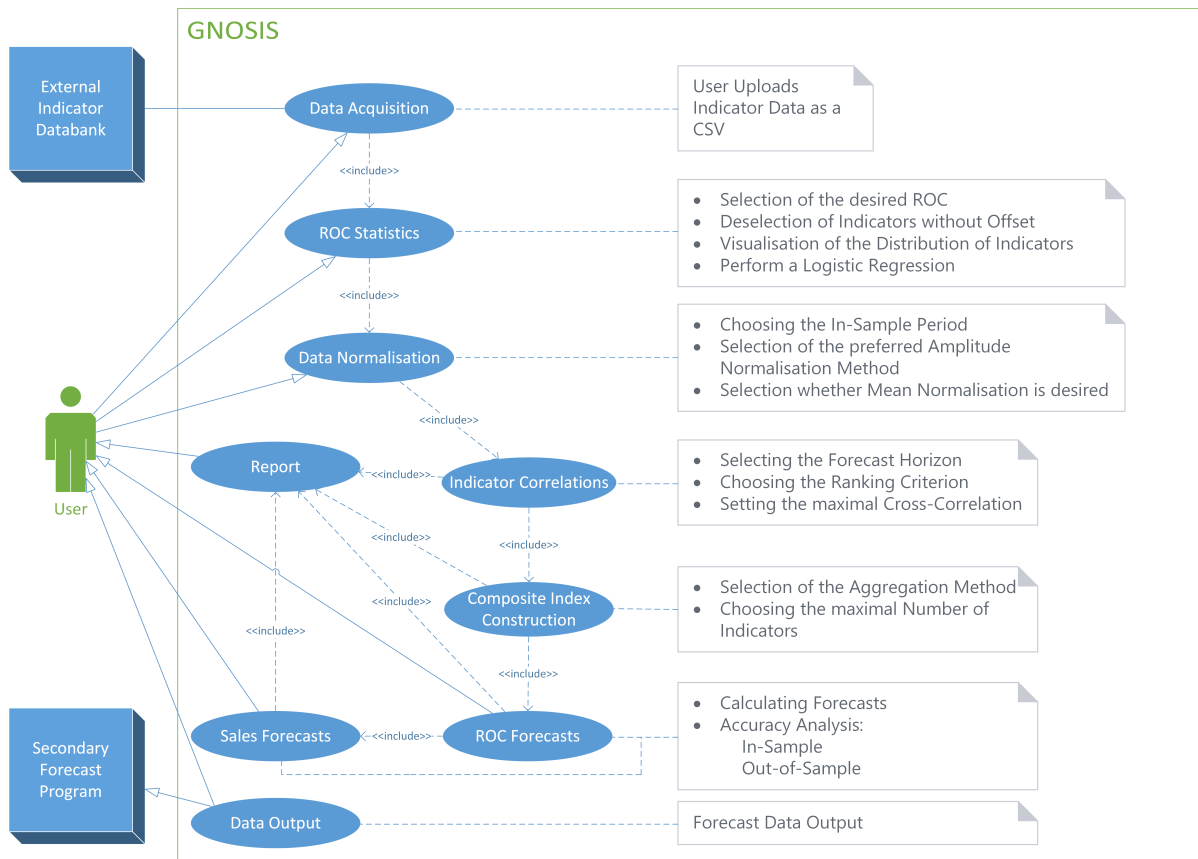


Figure 4.3.: Use Case

On the left-hand side of figure 4.3 the actors are introduced. In the middle of this column is the user who interacts with GNOSIS. It is shown that the user must first load data into GNOSIS. The data can be his own data, such as his own sales data, or external data, such as macroeconomic indicators. After the user has uploaded the datasets into GNOSIS the data can be inspected.

Subsequently to the Data Acquisition the ROC Statistics starts. It should be noted that the ROC Statistics needs the data from the Data Acquisition and therefore the connection between them is symbolised as include. During the ROC Statistics the user can select if some datasets should be excluded from the ROC calculation. This is necessary if some data have a predefined offset value. In this step the data will be visualised. This means that the historical sales data are displayed in a histogram to detect a possible distribution of the sales.

In addition, a boxplot shows the different indicators to check if the different indicators have the correct offset value. Finally, it is possible to make a logistic regression of the sales. After the ROC Statistics has been completed, the data is normalised. The first step is the definition of the in-sample period of the data. Subsequently, the desired amplitude normalization method is set and whether a mean normalization is desired by the user. In the next step the different indicators are ranked. First, the desired time lag must be specified. Then the desired ranking criterion must be selected and how high the cross-correlation between two indicators may be.

Once the indicators have been ranked, they need to be aggregated to create a composite indicator. In this step the desired aggregation method must be set and how many indicators should be used in the forecast. After the user has defined all parameters, the different forecasts are calculated. One forecast is the ROC forecast and the other calculated forecast is the sales forecast. It is important to mention that the accuracy of the forecasts is displayed to the user in order to get an estimate of the quality of the forecast.

After the forecasting step, it is possible to vary the previously selected parameters of the calculation and to create a new forecast. If the forecast is sufficient for the user, it can be downloaded as a report or as a file. This file could be used by the user for further analysis or could be used in a secondary forecast program.

4.3. Decision for R

The software which will be used to model GNOSIS must have implemented numerical and statistical calculation methods. In addition, this program must also be able to access previously calculated values, adapt input data values and change their data format, and allow self-written functions to be defined.

Another selection criterion for the software is that it can also be used to generate a graphical user interface. This is necessary so that a user with little IT knowledge can use GNOSIS. Finally, GNOSIS must be usable by companies. This results in another boundary condition that the chosen software has to be available as a freeware or have low usage fees. In summary, a software must meet the following requirements in order to be selected:

- implemented mathematical methods,
- implemented statistical methods,
- creation of a user interface is possible, and
- low costs.

These above-mentioned criteria have to be fulfilled by a software used for the generation of GNOSIS. Widely available software applications which satisfy some of these criteria are Microsoft Excel, MATLAB, Python or R. The software of choice is R. R was chosen because “R is an integrated suite of software facilities for data manipulation, calculation and graphical display” (Venables and D. M. Smith (2019, p. 2)).

One of the properties R exhibits is that it can effectively handle and save data. A very useful property of R is that it can perform calculations with arrays; the advantages of arrays will be discussed later. Another important point is that there is a large collection of integrated tools for rudimentary data analysis in a statistical context.

Since R was developed from the programming language S, this has the advantage that in R conditions, loops, functions, which can be developed by users, and input and output functions are available. Additionally, R is further developed by the individual software

developers.

This is reflected in the large number of packages which can be used by all software developers (Venables and D. M. Smith (2019)). The packages are a key feature that makes R very useful, since there is a big variety of packages with predefined functions enabling the user to use them without having to create the function manually.

The creation of the user interface can be done with the R package “Shiny”, which will be further explained in chapter 4.4. The last point is that R is released under a GNU license, which is a copyleft license. Therefore, R is open-source and free of charge for the user.

4.4. Shiny

Shiny is a package for building interactive web applications using R. It always consists of two elements, which have to be developed by the programmer. The first one being the user interface (UI) and the second one being the server. The UI will be further described in chapter 4.6 and the server will be further described in chapter 4.7. One of the main benefits of Shiny is that it allows to create a user interface that can access and interact with R-code.

Shiny applications are written in R code and do not necessarily require other programming language. However, it should be pointed out that for some special solutions for web applications JavaScript, CSS, and HTML can be used. Some benefits of using Shiny for the development of applications are the possible implementation of already existing widgets that can be modified to fit the specific needs, the use of default UI themes using Bootstrap, and its reactivity to user inputs which eliminates the need for reloading the browser. (Chang (2019))

Another important feature of Shiny is its ability to process data uploaded by the user while also allowing a manipulation of the data by the user and to enable the user to download data, i.e. as a Microsoft Excel spreadsheet which is especially important for further use of a constructed sales forecast.

The functionality of Shiny is further increased by its ability to access the package RMarkdown which can create reports of the construction, calibration and validation of the constructed forecast and enables the user to output these comprehensive reports as a PDF, Microsoft Word document or HTML code.

4.5. Functions

A function is an algorithm that is written in R code and can be called by writing the assigned name of the functions with two brackets, e.g. `mean(x)` would be a function already predefined in R which calculates the mean of the object `x` set inside the brackets.

R comes already equipped with a multitude of built-in functions. These include commonly used mathematical and numeric functions, e.g. `abs(x)` calculating the absolute value of `x` or `round(x, digits = n)` rounding the object `x` to `n` digits. Additionally, R has many character manipulating functions and many statistical functions which are not further listed for brevity reasons.

One of the key benefits, already described in chapter 4.3 is the option of using packages which are functions contained in libraries that are importable when needed. As an example, `ggplot2()` is used for the generation of graphical depictions of data, for example as a histogram.

However, it is often necessary to develop new functions for specific calculations needed to solve a specific problem. A user defined function always has the structure shown in listing 4.1. (Venables and D. M. Smith (2019))

```
1 # General Structure of a Function
2 name <- function(arg1, arg2, ...){expression}
```

Listing 4.1: General Structure of a Function in R

The expression is usually an algorithm that uses the arguments to perform some sort of calculation or transformation. The function is then used by calling it, following the example in listing 4.2 (Venables and D. M. Smith (2019))

```
1 # Calling a Function
2 name(val_1, val_2, ...)
```

Listing 4.2: Calling a Function in R

As an example for a more complex user defined function listing 4.3 shows a part of the code of GNOSIS. It shows the function for calculating the different correlations explained in chapter 3.2.

```

1 # Calculating the Correlations
2
3 PCOR_Cal <- function(x, selector , l, m , end, start) {
4   y <- c("pearson", "kendall", "spearman")
5   y1 <- (NA)
6   if (selector == 0) {
7     y1 <- y[1]
8     f1 <- c(1)
9   }
10  if (selector == 1) {
11    y1 <- y[2]
12    f1 <- c(0)
13  }
14  if (selector == 2) {
15    y1 <- y[3]
16    f1 <- c(0)
17  }
18  if (selector == 3) {
19    y1 <- y[4]
20    f1 <- c(0)
21  }
22
23  PCOR <- array(NA,
24                dim = c(25, 4, m),
25                dimnames = list(c(-24:0),
26                                c("Time Lag", "12/12", "3/12", "1/12"), names))
27
28  for (k in 1:m) {
29    for (j in 1:4) {
30      if (j == 1) {
31        PCOR[, 1, k] <- c(-24:0)
32      }
33      if (j > 1) {
34        for (z in -24:0) {
35          PCOR[(25 + z), j, k] <-
36            cor(x[start:(end + z), j - 1, k],
37              x[(start - z):(end), j - 1, 1],
38              method = y1,
39              use = "complete.obs") * sd(x[start:(end + z), j - 1, k], na.rm
40                                         TRUE) ^ f1 /
41            sd(x[(start - z):(end), j - 1, 1], na.rm = TRUE) ^ f1
42        }
43      }
44    }
45  }
46
47  return(PCOR)
48 }

```

Listing 4.3: Example for Functions from GNOSIS

Here the arguments inserted into the brackets of the PCOR_CAL in row 3 of listing 4.3 are necessary for the calculation. x is an array containing the ROC values of the sales and the indicators, the selector is dependent on the user's choice during the construction

of the sales forecast, l denotes the length of the time series found in the x array, m denotes the number of indicators plus 1 for the sales, $start$ denotes the first month of the in-sample period and end the last month of the in-sample period.

The reason for using arrays is that it simplifies the code significantly. For example, if the user chooses to construct three sales forecasts at once with different ROCs, an array with dimensions (i,j,k) can store the different values of one time series in the i dimension, the different ROCs of the same indicator can be stored in j dimension and the different time series (i.e. different indicators) can be stored in the k dimension.

The correlation values are also stored in an array, named PCOR, which is defined in lines 23 - 26 in listing 4.3. Each m dimension is for itself a matrix containing the different correlations of the indicators with the sales in dependence of the time lag of those indicators. It is preliminarily filled with NA, which stands for “not available” and gets overwritten in the next step which is the calculation of all the correlations between the differently time lagged indicators and the sales.

In order to adhere to the methodology of XP described in chapter 4.1, it is necessary to keep the code as simple as possible. Therefore a vector is introduced in line 4 of listing 4.3 and is filled with the different correlation methods in a form that makes them viable as an input for the built-in R function `cor()`. In lines 6 - 21 of listing 4.3, two variables, $y1$ and $f1$, are introduced, which get assigned different values according to the user’s choice of correlation method.

$y1$ is the variable to be used as the input to the `cor()` function, in order to select the correlation method and $f1$ is used as an exponential factor which is 0 for all correlations except the calculation of the β coefficient where the function `sd()`, which calculates the standard deviation, is used. The calculation of the different correlation coefficients and the storing into the PCOR array can be seen in lines 28 - 45 of listing 4.3. Finally, the line 47 shows the returning of the PCOR array which makes it available for further manipulation or calculations outside of the function.

4.6. User Interface

The UI code of Shiny creates the user interface and allows for the construction of a web application. It is possible to add text, images and additional HTML elements. The UI normally follows a structure similar to the code in listing 4.4. Here the `fluidPage()` function creates a “display that automatically adjusts to the dimensions of your user’s browser window” (RStudio (2019)).

The function `titlePanel()` is necessary to create a header for the respective page of a Shiny page. The function `sidebarLayout()` is used to create a panel as a sidebar and a main panel. The text in the brackets is printed in the displayed page as the title of the panel. The quotation marks are needed to tell Shiny that the input is to be seen as a character string.

```
1 # General Structure of a Shiny UI
2
3 ui <- fluidPage(
4   titlePanel("title panel"),
5
6   sidebarLayout(
7     sidebarPanel("sidebar panel"),
8     mainPanel("main panel")
9   )
10 )
```

Listing 4.4: General Structure of a Shiny UI

As an example for a more complex UI panel, listing 4.5 shows a part of the code of the UI of GNOSIS. It shows the UI code for the generation of the tab panel for the Data Normalisation step. The different options that are defined in listing 4.5 is a direct result from the requirements in the chapters 3.2, 3.3 and 3.4. The user interface resulting from the code in listing 4.5 can be seen in chapter 5.

```
1 ##### Panel Data Normalisation #####
2
3 tabPanel( "Data Normalisation",
4   # Sidebar Panel
5   sidebarPanel( h2("Normalisation"),
6     h4("In-sample period"),
7     p("Please select the time period of the data for the training set. It
        should contain at least 51 months of data."),
```

```

8 |     style = "font-family: 'times'; font-size: 16pt"
9 |   ),
10 | # Defining in-sample Period
11 | dateRangeInput(
12 |   "insample",
13 |   label = NULL,
14 |   format = "mm/yyyy",
15 |   language = "en",
16 |   start = "2000-01-01",
17 |   end = "2005-01-01",
18 |   startview = "year",
19 |   separator = " - "
20 | ),
21 | # Normalisation Methods
22 | h4("Amplitude Normalisation Method"),
23 | p("Please select the amplitude normalisation method.", style = "font-family
24 |   : 'times'; font-size: 16pt"),
25 | radioButtons(
26 |   "ampl.norm",
27 |   label = NULL,
28 |   choices = list(
29 |     "Volatility Scaling" = 2,
30 |     "Min/Max Scaling" = 3,
31 |     "Integral Scaling" = 4
32 |   ),
33 |   selected = 2
34 | ),
35 | h4("Mean Normalisation"),
36 | p("Please select whether a mean normalisation should be applied.",
37 |   style = "font-family: 'times'; font-size: 16pt"
38 | ),
39 | radioButtons(
40 |   "mean.norm",
41 |   label = NULL,
42 |   choices = list("Yes" = 1,
43 |                 "No" = 2),
44 |   selected = 1
45 | ),
46 | # Main Panel
47 | mainPanel(tabsetPanel(
48 |   tabPanel("Rate of Change",
49 |     br(),
50 |     conditionalPanel(condition = "input.rocselect.indexOf('A') > -1",
51 |       plotOutput("ROC_S12_12"), br()),
52 |     conditionalPanel(condition = "input.rocselect.indexOf('B') > -1",
53 |       plotOutput("ROC_S3_12"), br()),
54 |     conditionalPanel(condition = "input.rocselect.indexOf('C') > -1",
55 |       plotOutput("ROC_S1_12"))
56 |   ) ) )

```

Listing 4.5: Example of UI Code from GNOSIS

In listing 4.5 the code shows the creation of a tab panel, featuring a sidebar panel for the selection of parameters and a main panel for the display of reactive plots showing the impact of the chosen parameters on the data. The function `dateRangeInput()`, shown in lines 11 - 20 of listing 4.5, creates a Shiny widget that is used for the calibration of the in-sample period. Line 12 assigns a name to the vector storing the start and end date.

In lines 24 - 33 of listing 4.5 the code shows the creation of a Shiny widget with the function `radioButtons()`. This is used for the selection of the amplitude normalisation

method. The argument `choices` generates the different radio buttons and the argument `selected` defines a default selection.

The main panel uses the function `conditionalPanel()` to create adaptive tab panels depending on the selection of different ROC methods made by the user in another tab panel prior to this tab panel. The condition argument is a boolean function which checks the condition and only prints the plot output if the condition is fulfilled. It should be noted that the condition is not R-code but uses the function of R to implement JavaScript when the functionality of R does not suffice.

The `p()` function used multiple times allows for the creation of a text output with a custom font and font size. The `h4()` function creates a headline with the pre-set size and font.

The packages used for the UI of GNOSIS can be found in Appendix B.

4.7. Server

Any Shiny application needs a server file which serves as the back-end, while the UI is the front-end. The server performs all the calculations and transformations of the data and the objects. The server can access other files, for instance a file containing functions, to load those algorithms and perform the calculations. The basic structure of a server is shown in listing 4.6.

The function(input, output) term is necessary to access the inputs by the user performed in the graphical user interface and to export the output objects created in the server for the UI. Additionally, one could add an argument to make Shiny reactive to changed input values and even allow for changes in the UI defined in the server and dependent on the input values.

```
1 # General Structure of a Shiny Server
2
3 server <- function(input, output) {
4
5 }
```

Listing 4.6: General Structure of a Shiny Server

As an example for real server code listing 4.7 shows code from GNOSIS. The function called in this example is the same function which is explained in chapter 4.5.

```
1 # Calling the function PCOR_Cal
2
3 PCOR <-
4   reactive({
5     PCOR <-
6     PCOR_Cal(ROC_SCALED(), input$soi.cor , l() , m() , is.end() , is.start())
7   })
8
9 # Plotting the Correlation Coefficients
10
11 output$CM12 <-
12   renderPlot(ggcorrplot(PCOR()[, 2, ],
13                        title = "Correlation matrix for 12/12",
14                        lab = TRUE))
15 output$CM3 <-
16   renderPlot(ggcorrplot(PCOR()[, 3, ],
17                        title = "Correlation matrix for 3/12",
18                        lab = TRUE))
19 output$CM1 <-
```

```

20 | renderPlot(ggcorrplot(PCOR()[, 4, ],
21 |               title = "Correlation matrix for 1/12",
22 |               lab = TRUE))

```

Listing 4.7: Example of Server Code from GNOSIS

The code shown in line 6 of listing 4.7 calls the user defined function `PCOR_Cal()` which is used for the calculation of the indicators' correlation coefficients with the sales. The arguments inside the brackets are needed for this calculation. However, it should be noted that the brackets behind some of the variables used as arguments signify that these objects are reactive. This means that their value can change depending on the user's input.

The argument `input$soi.cor` is a vector containing the user's input directly from the UI. The lines 3 and 5 in listing 4.7 are assigning the return value of the function `PCOR_Cal` to an object called `PCOR` which is set to be reactive by the code shown in line 4 of listing 4.7.

The code shown in lines 11 to 22 of listing 4.7 shows the generation of plots, used in the UI for display to the user. The three plots are analogue to each other and only differ in the type of ROC on which the calculation is based. Therefore, only the code in line 11 to 14 of listing 4.7 will be explained.

This code creates a plot using the functions `renderPlot()` and `ggcorrplot()` to represent the correlation coefficients of the calculation performed with the $ROC_{12/12}$. The arguments in the square brackets after `PCOR()` prompt the server to only use the second column of the array. The argument `title` is self explanatory and the argument `lab` prompts the plot to feature the actual numerical values of the correlation coefficients and not only depict them in form of a heat map.

The functions `renderPlot()` and `ggcorrplot()` are additional functions from packages which have to be loaded as libraries prior to the calculation. The list of all the packages required for the server of GNOSIS can be found in Appendix B.

5. Demonstration of GNOSIS

In the following sections the software tool GNOSIS will be demonstrated using sales data from a real company (ARC). For confidentiality reasons, the sales data has been transformed numerically to only represent the overall trends of the real sales data.

The indicators used for the construction and calibration of the sales forecast were picked by experts from ARC. The sales data was provided as monthly sums ranging from January 2000 to December 2018.

The first section will demonstrate the construction, calibration and validation of a sales forecast based on the provided data from ARC. The point of the section is to show the design and functionality of GNOSIS.

The second section will perform a parameter variation, to examine the differences in predictive accuracy of the ROC forecasts and the sales forecasts dependent on the different construction and calibration methods.

5.1. Creating a Sales Forecast using GNOSIS

In this chapter a user defined sales forecast will be created. It will be discussed how a user can upload the required data needed for the forecast. After the user has uploaded the data, it will be shown how the different menus of GNOSIS are designed and in which way the user can customise GNOSIS for the forecasting task. Additionally, the different plots which GNOSIS creates will be shown and further discussed. Finally, after the forecast has been constructed, calibrated and validated by the user, it will be shown how the user can create and download a report and a Microsoft Excel spreadsheet containing the different data of the forecast.

It should be mentioned that the performed forecast should only illustrate the functionality of GNOSIS and is not a recommendation for parameter choice. Due to concerns for brevity only a sales forecast constructed using the $ROC_{12/12}$ will be shown.

For the performed forecast it will be assumed that the user has the necessary data available. This data should contain internal data from a company, i.e. sales, and macroeconomic indicators which have been selected by experts. This data must be saved in a .csv file format to be accessible by GNOSIS. For this reason, table 5.1 shows how the CSV file must be structured.

Table 5.1.: Exemplary CSV File

Dates	Internal Data	Indicator 1	Indicator 2	Indicator 3	...
YYYY-MM-DD	Value ID1	Value I1 1	Value I2 1	Value I3 1	...
YYYY-MM-DD	Value ID2	Value I1 2	Value I2 2	Value I3 2	...
⋮	⋮	⋮	⋮	⋮	⋮

5.1.1. Data Acquisition

The first tab “Data Acquisition” of GNOSIS is shown in figure 5.1 and deals with the data acquisition. The user can adjust the parameters featured in the sidebar panel, in order to adapt the read out function of R to fit the data structure. For example, the user can adjust the separator between values of the original file in order to make the file readable for Shiny.

It is important to choose these values prior to the upload in order to make the program run properly. The upload itself is solved by the implementation of an upload button which allows the user to browse the computer for the file and upload it simply by selecting it. Additionally, the user can visually inspect the data and its structure in the first tab panel of the main panel.

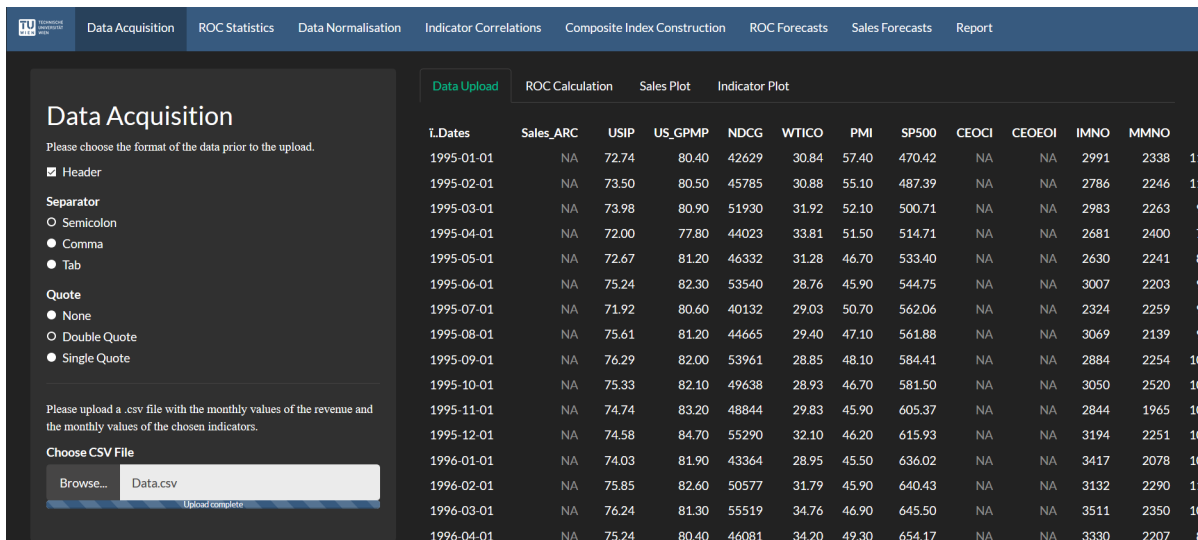


Figure 5.1.: GNOSIS Data Acquisition

The second tab panel of the main panel of “Data Acquisition” is called “ROC Calculation” and is intended to allow the user to deselect indicators. It is shown in figure 5.2. The deselected indicators will not be processed like the rest, meaning that their rate of change will not be calculated. This is an important functionality, since some indicators are not depicting real values but rather rate of changes or yearly trends. Other than skipping the rate of change calculation, the deselected indicators are processed like the

rest of the indicators.

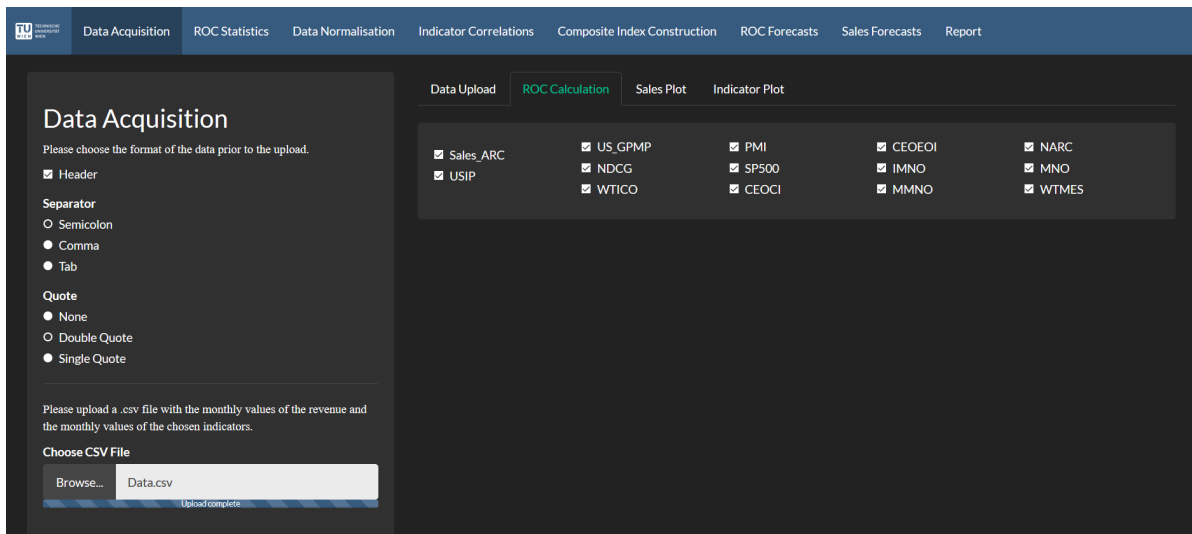


Figure 5.2.: GNOSIS Data Acquisition - ROC

The other two tab panels of the main panel, namely “Sales Plot” and “Indicator Plot”, are merely plots depicting the uploaded sales data and the uploaded indicators without any changes to their values. Hence, for brevity reasons those tab panels will not be shown or further explained.

5.1.2. ROC Statistics

The second tab of GNOSIS is the “ROC Statistics” tab which is shown in figure 5.3. The sidebar allows the user to choose which types of ROC calculations are desired. If the user chooses more than one type of ROC, more than one forecast would be constructed. GNOSIS would then depict the plots vertically aligned and the tables would be expanded horizontally.

The first tab panel of the main panel “Distribution Plot” features a historical distribution plot of the ROCs of the sales values over all the data that was uploaded. The second tab panel “Box Plot” depicts a box plot of the historical ROC values of the indicators and is not depicted for brevity concerns.

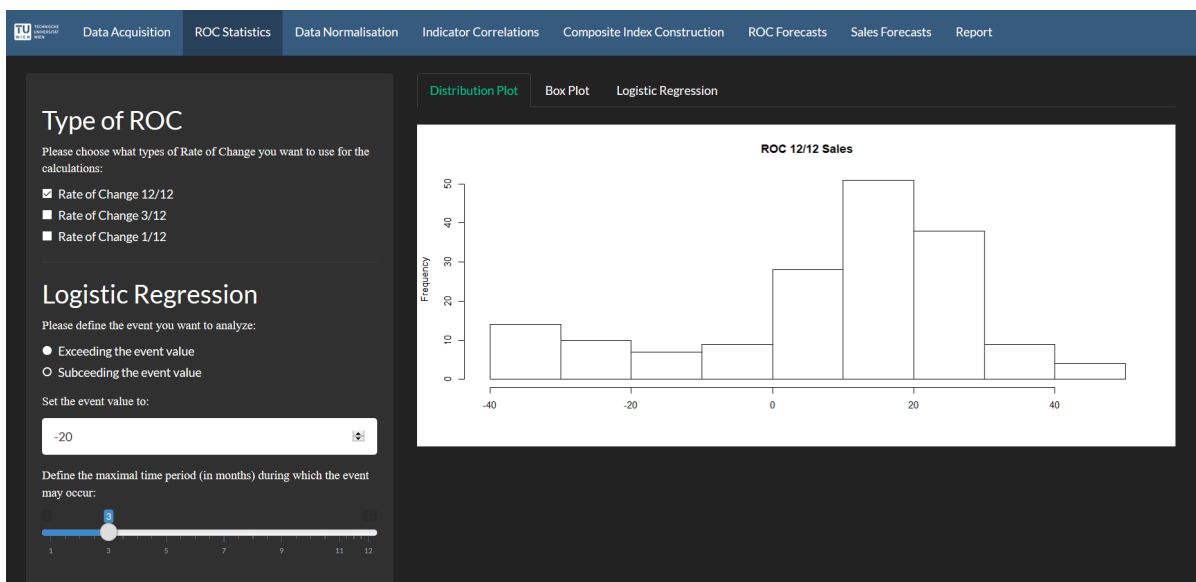


Figure 5.3.: GNOSIS ROC Statistics - Sales

The third tab panel of the main panel is the “Logistic Regression” panel and is shown in figure 5.4. It features a plot depicting the probability function of a logistic regression of the ROCs of the sales. The logistic regression requires a discrete and defined event, that can occur or not occur. This event can be set by the user in the sidebar panel.

The user can adjust the logistic regression to depict whether a certain value was exceeded or subceeded in a certain number of months following a certain ROC at the start period. If the user sets the value of months and the value to be ex- or subceeded, the user can click on the probability function in the plot to generate the probability for the different possible ROCs at the start period.

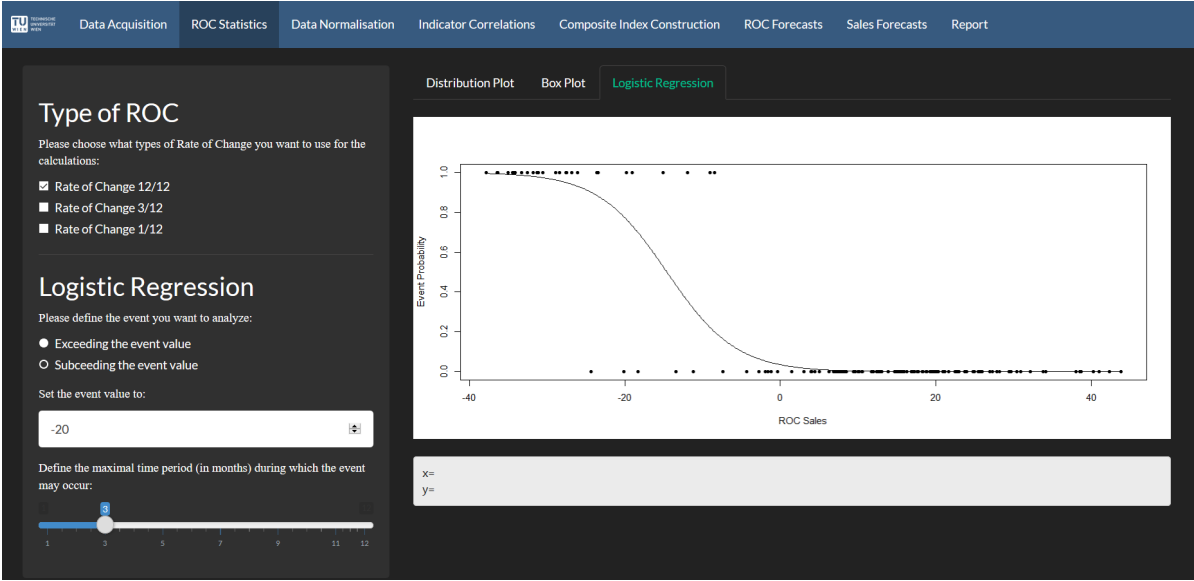


Figure 5.4.: GNOSIS ROC Statistics - Logistic Regression

5.1.3. Data Normalisation

The third tab of GNOSIS is the tab “Data Normalisation” shown in figure 5.5. It has only one tab panel in the main panel, depicting the time series of the sales ROC in black and the time series of the normalised indicators’ ROCs in green. In the sidebar panel, the user can choose the in-sample period, by clicking on the date and choosing a new date with a calendar widget. The amplitude normalisation methods can be selected by the user and the mean normalisation could be deactivated. The plot in the main panel is reactive and shows the user the resulting changes in normalisation in real time.

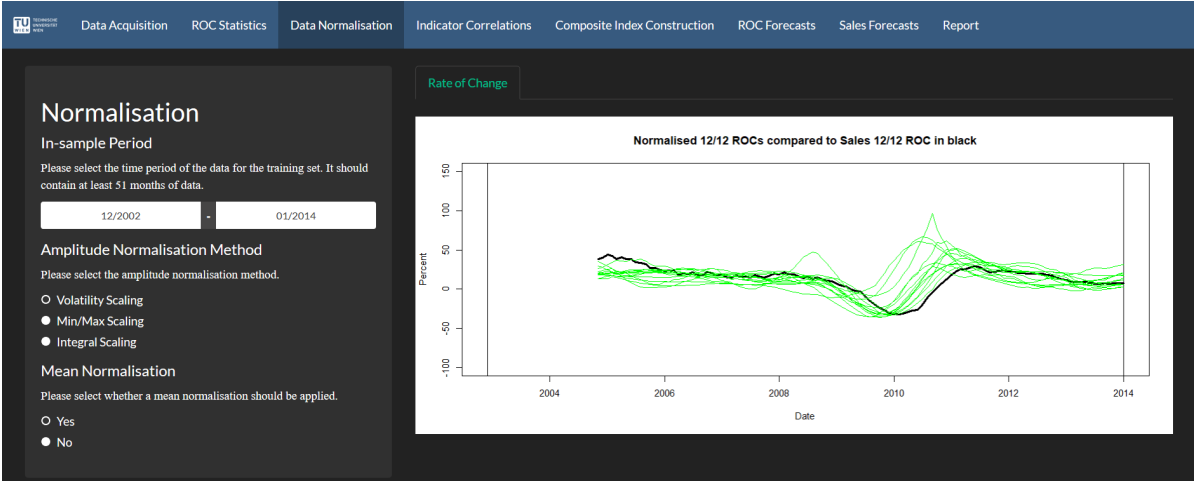


Figure 5.5.: GNOSIS Data Normalisation

5.1.4. Indicator Correlations

The fourth tab of GNOSIS is the tab “Indicator Correlations” shown in figure 5.6. Here, the user can select the forecast horizon in the sidebar panel. In the sidebar panel the user can select the correlation analysis method for determination of the level of correlation between the sales and the different indicators with the different time lags. Additionally, the user can set a maximal value for the allowed cross correlation between the indicators. If the cross-correlation between two indicators exceeds this value one of the indicators will be excluded from the further construction steps.

The first tab of the main panel “Table” depicts the different correlations with their time lag at the biggest correlation coefficient with the boundary condition of the time lag being at least as large as the set forecast horizon for the indicator to be included in the subsequent steps.

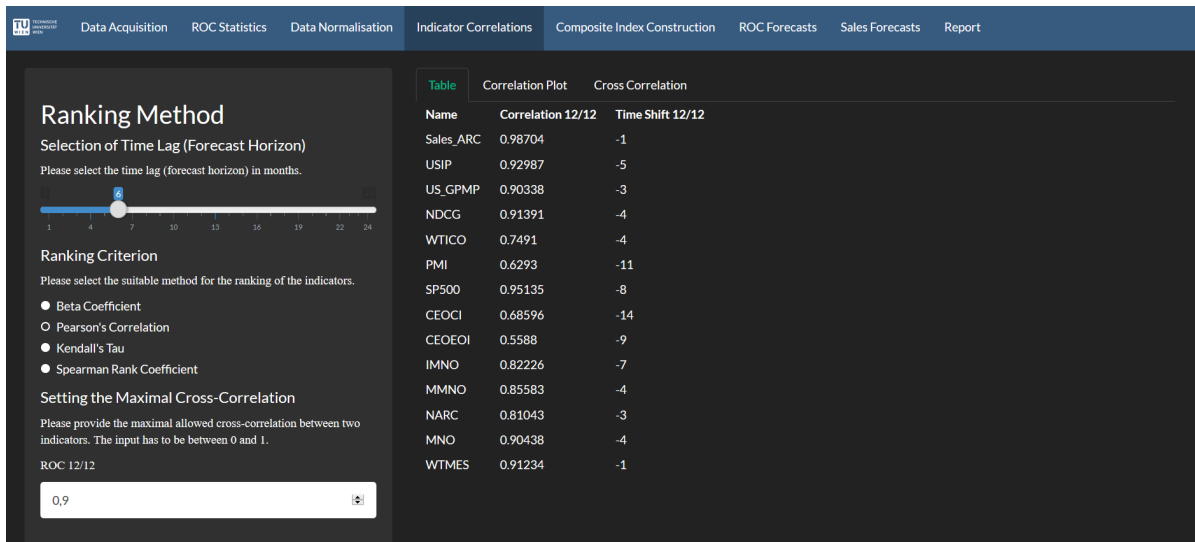


Figure 5.6.: GNOSIS Indicator Correlations - Table

The second tab of the main panel is shown in figure 5.7. It depicts a heat map with values of the different correlation coefficients of the indicators with the sales. The x-axis features the different time lags, ranging from 0 months to 24.

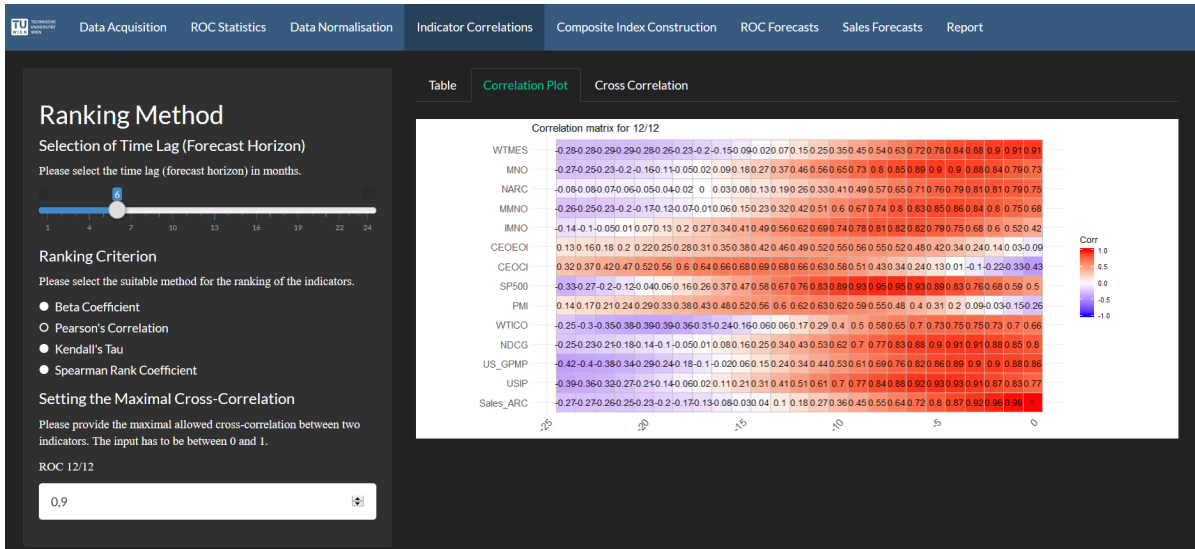


Figure 5.7.: GNOSIS Indicator Correlations - Correlation Plot

Figure 5.8 shows the third panel of the main panel. Here, a heat map with the values of the different indicators' cross correlation coefficients is shown. It can be seen as a depiction of the cross correlation matrix over a heat map. The different time lags are not shown, since the cross correlation is only important for the time lags shown in the tab panel "Table" of the main panel. Hence, the indicators have the time lag shown in the tab panel "Table".

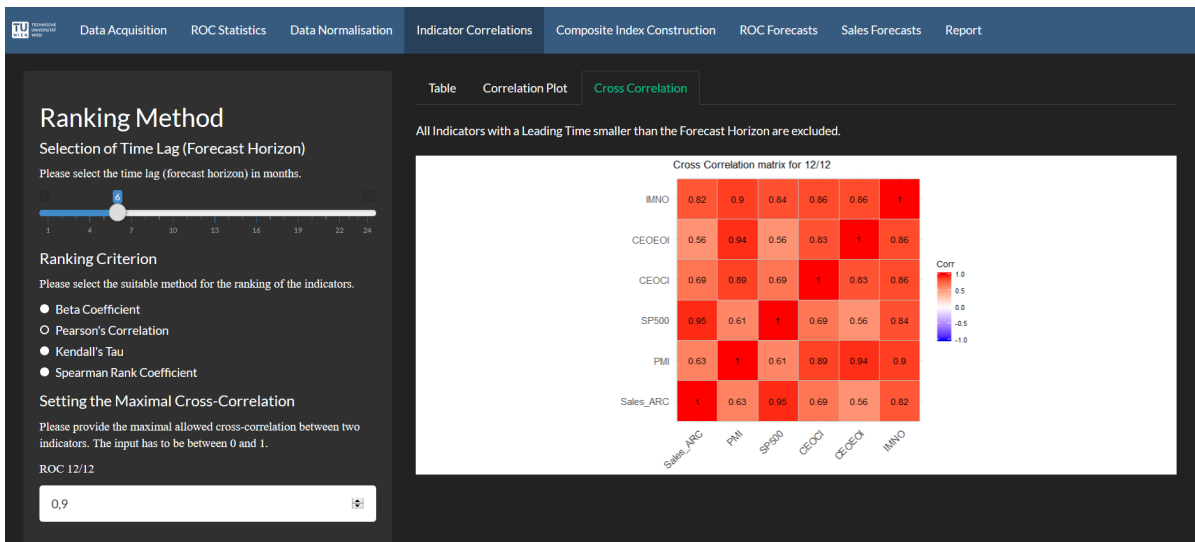


Figure 5.8.: GNOSIS Indicator Correlations - Cross Correlation

5.1.5. Composite Index Construction

Figure 5.9 shows the fifth tab of GNOSIS, named “Composite Index Construction”. In the side bar the user can choose between a method-based aggregation of indicators and a custom weight assignment. Should the user choose the custom selection and weight assignment, the table in the main panel would change to represent a Microsoft Excel like spreadsheet where the user can manually insert the weights of the indicators of his choosing.

Otherwise, if the user chooses a method-based aggregation of indicators, the user sees the weights depicted in the first tab panel “Table” in the main panel as shown in figure 5.9. Additionally, the user can choose the aggregation method in the sidebar panel as well as the maximal number of indicators to be included in the composite index.

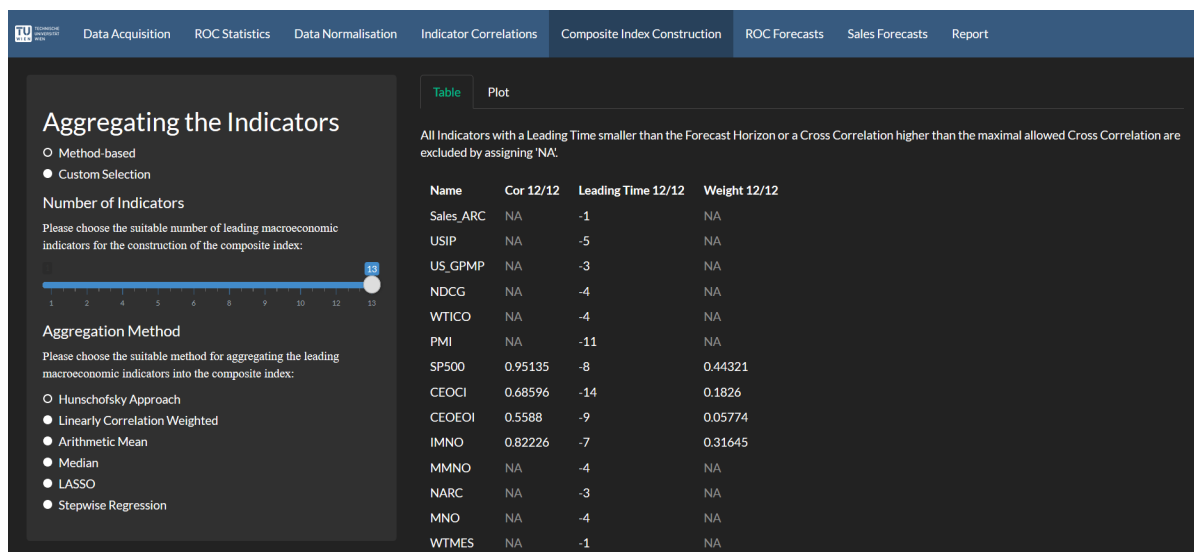


Figure 5.9.: GNOSIS Composite Index Construction - Table

The ‘NA’ values in the table in figure 5.9 mean that the indicator is not eligible for a composite index due to a too high cross-correlation or a too small time lag. However, if the LASSO method or the stepwise regression is chosen, the cross-correlation will not function as an exclusion criterion since those methods deal with cross-correlated indicators themselves.

In figure 5.10 the second tab panel of the main panel, called “Plot”, is shown. A plot depicting the resulting composite index and the sales ROC is shown to allow for the user to validate whether the chosen aggregation method or custom set weights are resulting in a composite index which is resembling the sales during the in-sample period.

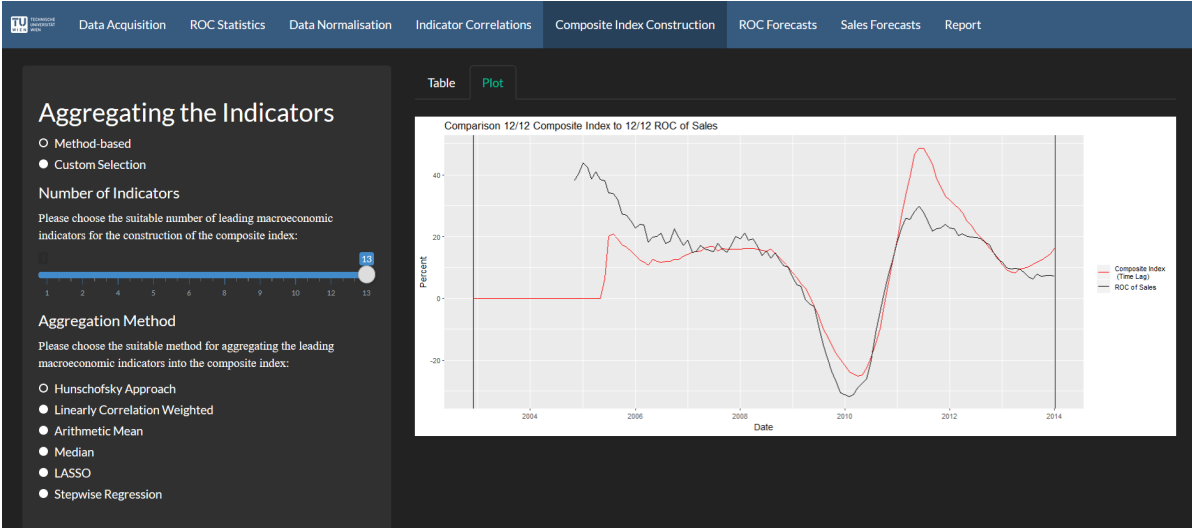


Figure 5.10.: GNOSIS Composite Index Construction - Plot

5.1.6. ROC Forecasts

The sixth tab of GNOSIS shows the constructed and calibrated ROC forecast as shown in figure 5.11. The first tab panel of the main panel features a plot depicting the sales ROC, the composite index during the in-sample period and the ROC forecast which is the composite index in the out-of-sample period.

The sidebar panel allows the user to select the desired benchmark forecasts which are then also depicted in the plot. Additionally, the user can choose from a list of different accuracy measures, both, for the in-sample as well as the out-of-sample period. If the user deselects one of these measures, it will not be displayed in the tab panel “Accuracy Analysis” of the main panel. Additionally, it will also be excluded from the report.

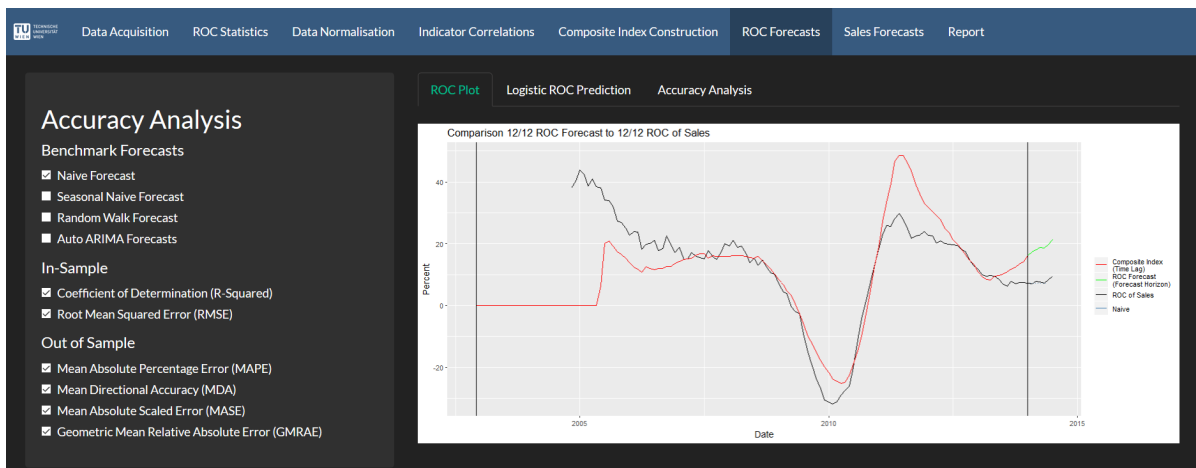


Figure 5.11.: GNOSIS ROC Forecasts - Plot

The tab panel “Logistic ROC Prediction” of the main panel is depicting an extended forecast, going further into the future than the forecast horizon and is shown in figure 5.12. This is achieved by defining two events. Namely, the event that the ROC falls below -20% or goes above 20% in a given period of time. Those periods are given with a probability derived from the logistic regression of the sales values and are compared to the real sales ROCs whenever they are known, i.e. if backtesting is performed.

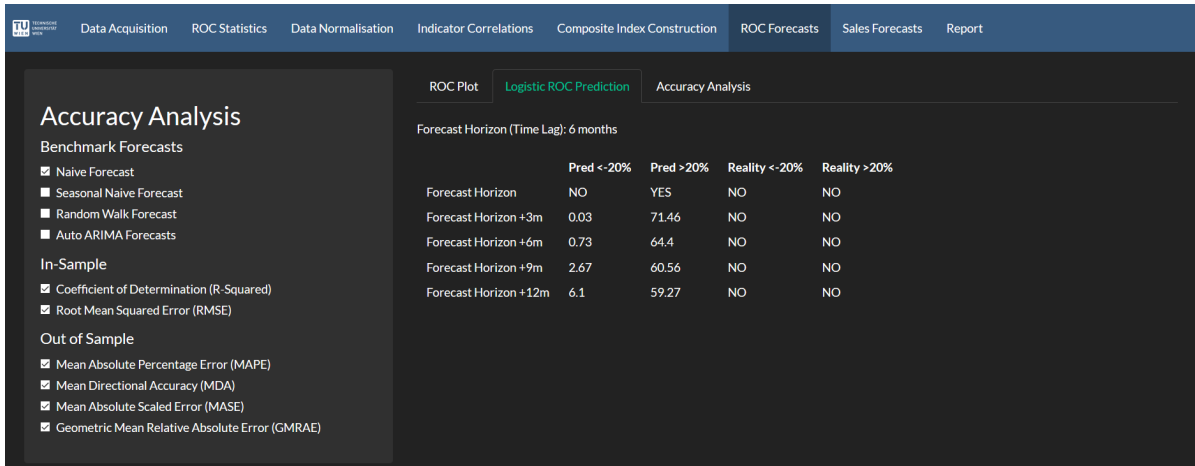


Figure 5.12.: GNOSIS ROC Forecasts - Logistic Prediction

The last tab panel of the main panel shows the selected accuracy measures and their values for the ROC prediction as shown in figure 5.13.



Figure 5.13.: GNOSIS ROC Forecasts - Accuracy Analysis

5.1.7. Sales Forecasts

The seventh tab of GNOSIS is called “Sales Forecast” and is shown in figure 5.14. The sidebar panel is analogue to the sidebar panel of the tab “ROC Forecast” with the difference being that the in-sample accuracy measures are not featured. This is due to the fact that the sales forecast is derived from the composite index using only the values of the out-of-sample period. Therefore, the user can only choose between different benchmark forecasts and the different out-of-sample accuracy measures. The tab panel “SF Plot” of the main panel shows a plot of the sales in black and the sales forecast in green.

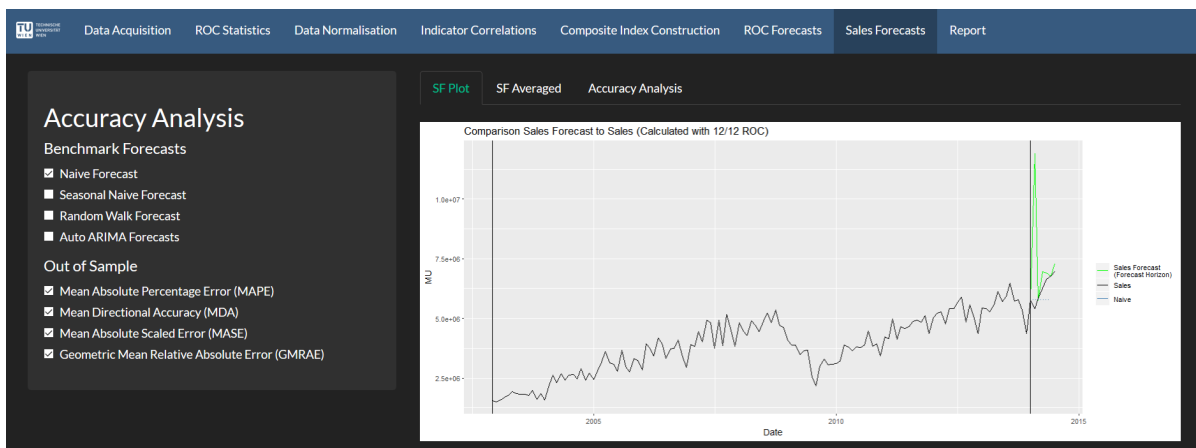


Figure 5.14.: GNOSIS Sales Forecasts - Plot

The second tab panel of the main panel is called “SF Averaged” and is shown in figure 5.15. Here the sum of the sales forecast and the sum of the sales is calculated for the out-of-sample period and an average monthly value is also depicted. Additionally, the deviation of the average value of the sales forecast to the sales and optionally the deviation of any benchmark forecast to the sales is shown in percent.

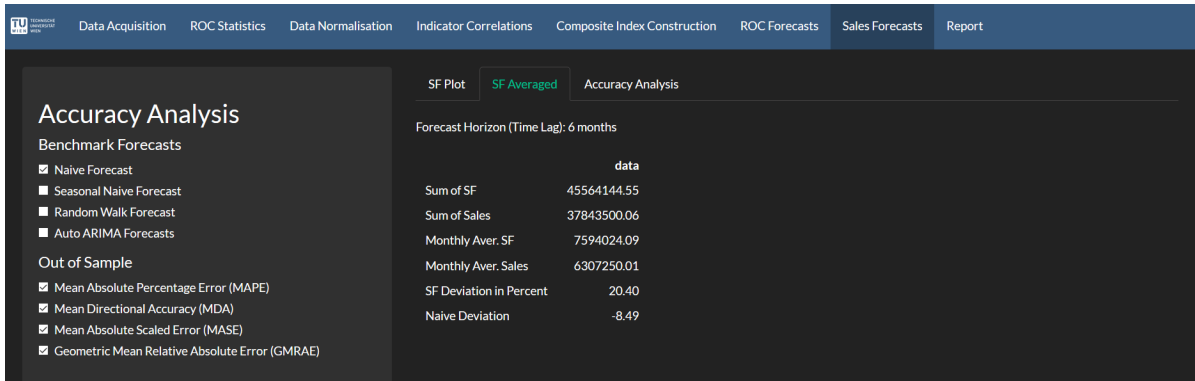


Figure 5.15.: GNOSIS Sales Forecasts - Averaged

The last tab panel “Accuracy Analysis” of the main panel can be seen in figure 5.16. It depicts the selected accuracy measures and their respective values.

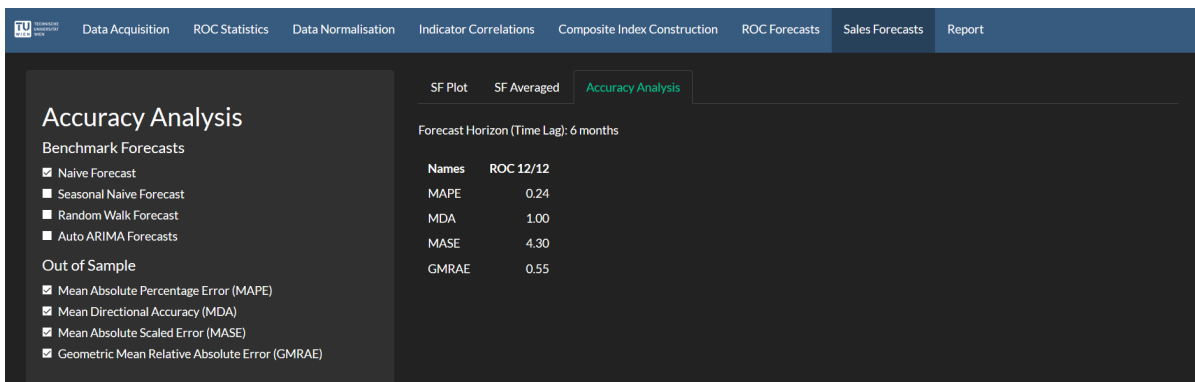


Figure 5.16.: GNOSIS Sales Forecasts - Accuracy Analysis

5.1.8. Report

The eighth and last tab of GNOSIS is the tab “Report” which is shown in figure 5.17. Here the user can download a report featuring all the chosen parameters, the depicted plots during the construction and calibration, as well as all the plots and results from the ROC and sales forecasts, including their accuracy analysis. The report can be either downloaded as a PDF file, as an HTML file or as a Microsoft Word document.

The download button seen under the headline “Download Data” in the main panel creates a Microsoft Excel file which features the values of the ROC and sales forecasts as a time series as well as the results of the accuracy analysis as a table.

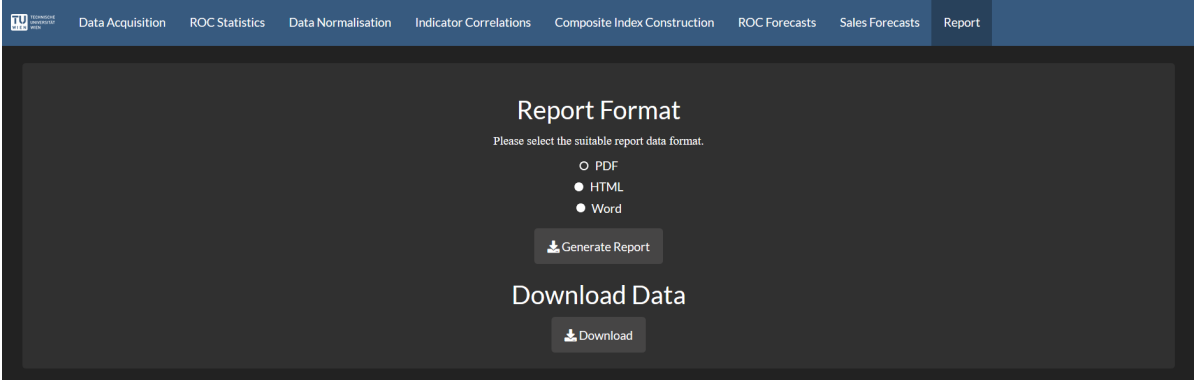


Figure 5.17.: GNOSIS Report Download

5.2. Analysis of Differently Constructed Sales Forecasts

This chapter will perform a parameter variation and will examine the impact on the MAPE of the ROC forecast and the MAPE of the sales forecast. The percentage deviation of the average value of the sales forecast compared to the sales can be found in appendix D and is excluded due to brevity concerns. The sales data is the sales of ARC and the indicators are picked by expert judgement and are provided by ARC as well. For a list of the different indicators the reader is referred to appendix C.

The parameter variation will vary the in-sample period, however, it will always end with December 2017 in order to use the same time period, i.e. 2018 sales data, for validation and backtesting. All the parameters varied are:

- the length of the in-sample period,
- the length of the forecast horizon,
- the normalisation method,
- the correlation method,
- the cross-correlation value, and
- the aggregation method.

The parameter variation only varies one parameter at a time keeping the rest always constant at the default values defined in table 5.2.

Table 5.2.: Standard Parameters for Analysis

Standard Parameters	
Method	Parameter
In-Sample Period Start	2005-01-01
In-Sample Period End	2017-12-01
Forecast Horizon	6 Months
Normalisation	Volatility-Scaling
Mean Normalisation	Yes
Correlation	Pearson-Correlation
Cross-Correlation	0.9
Aggregation	Linear Correlation Weighted
Number of Indicators	Maximal Number of Indicators

The reason for choosing the constant in-sample period end date at December 2017 is the availability of the sales data of ARC until December 2018, thus enabling backtesting for a forecast horizon of 12 months. The standard value for the forecasting horizon was set to 6 months, since this seems to be the smallest value for useful forecasts. The volatility scaling was used as the standard normalisation method as it is a commonly used normalisation method found in the literature.

The Pearson's correlation coefficient is used as it seems to fit the nature of the indicators best. The linear correlation weights are set as the standard aggregation method to use a WAM - method with a causal weighting according to the degree of correlation of the indicators to the sales. Finally, the maximal number of indicators is chosen to be able to differentiate the methods that exclude many indicators, i.e. the parameter variation concerning the cross-correlation.

5.2.1. Length of In-sample Period

In figure 5.18 the ROC forecast for different in-sample periods is shown. The $ROC_{1/12}$ has the smallest MAPE for short in-sample periods, however, five years seem to be the worst in-sample period. For longer in-sample periods than nine years the MAPE gets higher again. The $ROC_{3/12}$ is best for an in-sample period of ten to 14 years and the $ROC_{12/12}$ is best for 16 to 18 years.

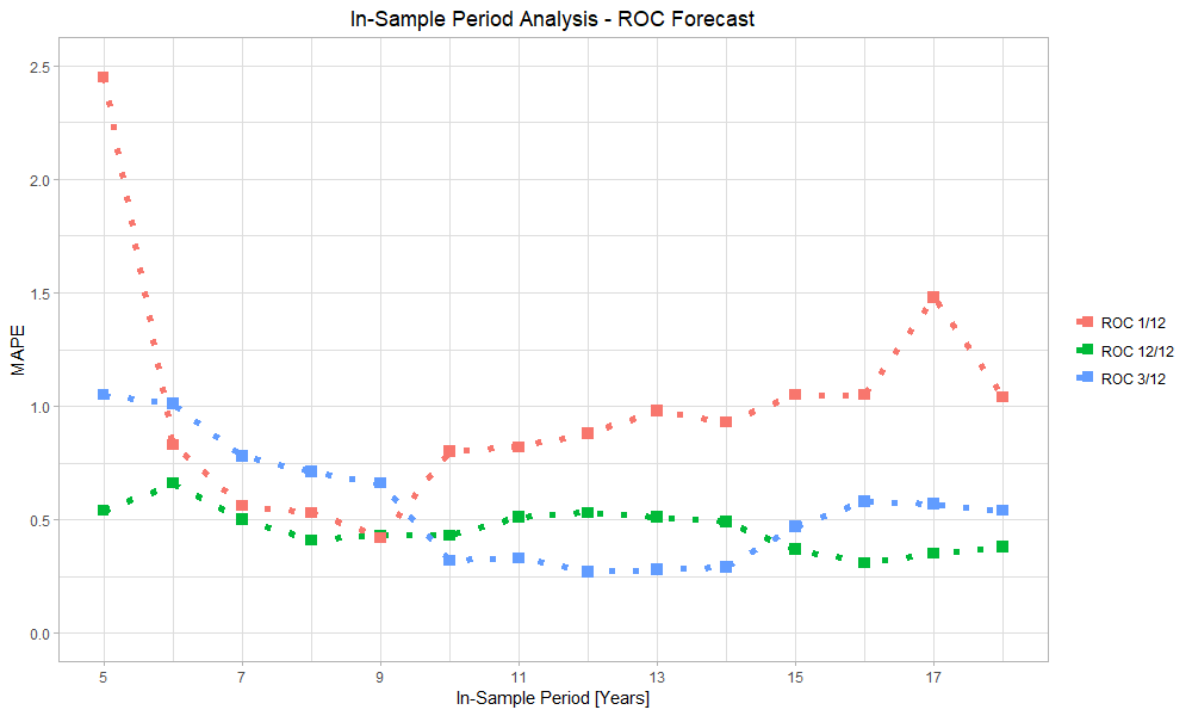


Figure 5.18.: ROC Forecast with Varied In-sample Period

In figure 5.19 the sales forecast for different in-sample periods is shown. It should be noted that the $ROC_{1/12}$ has the lowest MAPE in contrast to the ROC forecast where the $ROC_{1/12}$ had the highest MAPE. Another interesting point is that the MAPE for the $ROC_{3/12}$ and $ROC_{12/12}$ remained fairly constant, with the exception of the minimal MAPE at 16 years for the $ROC_{12/12}$ and 17 years for the $ROC_{3/12}$. The MAPE for the $ROC_{1/12}$ is maximal at 5 years and then remains very constant. This behaviour is

different from the ROC forecast where the $ROC_{1/12}$ worsened significantly with longer in-sample periods.

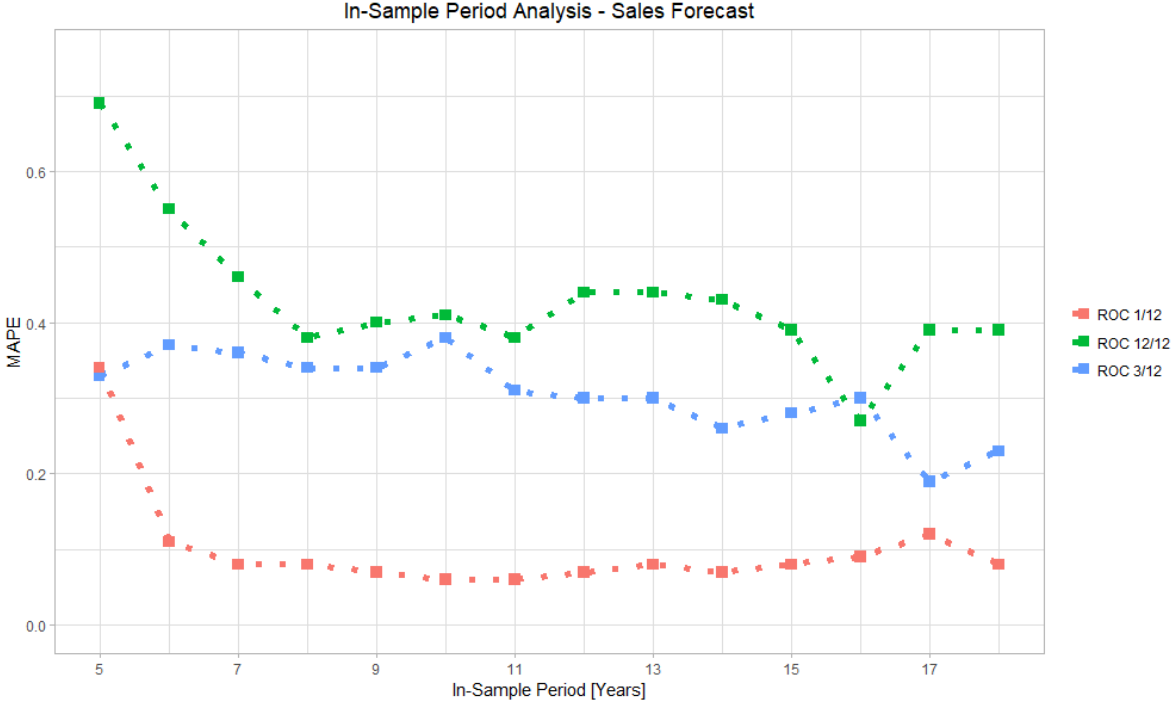


Figure 5.19.: Sales Forecast with Varied In-sample Period

5.2.2. Length of Forecast Horizon

The analysis of the ROC forecast for the variation of the forecast horizon is shown in figure 5.20. The $ROC_{1/12}$ has the lowest MAPE for a short forecast horizon but the MAPE becomes drastically higher if the forecast horizon increased in duration. The forecast for the $ROC_{3/12}$ and the $ROC_{12/12}$ is almost identical for a short forecast horizon but the forecast of the $ROC_{12/12}$ gets better with a longer forecast horizon. The longer forecast horizons exceeding 7 months can not be assessed for the $ROC_{1/12}$ and $ROC_{3/12}$ because there are no indicators left resulting the exclusion for too high cross-correlations and a too small time lag. Therefore the graph does not continue for those forecasts. The lowest MAPE for the ROC forecast is 2 months for the $ROC_{1/12}$, 5 months for the $ROC_{3/12}$ and 11 months for the $ROC_{12/12}$.

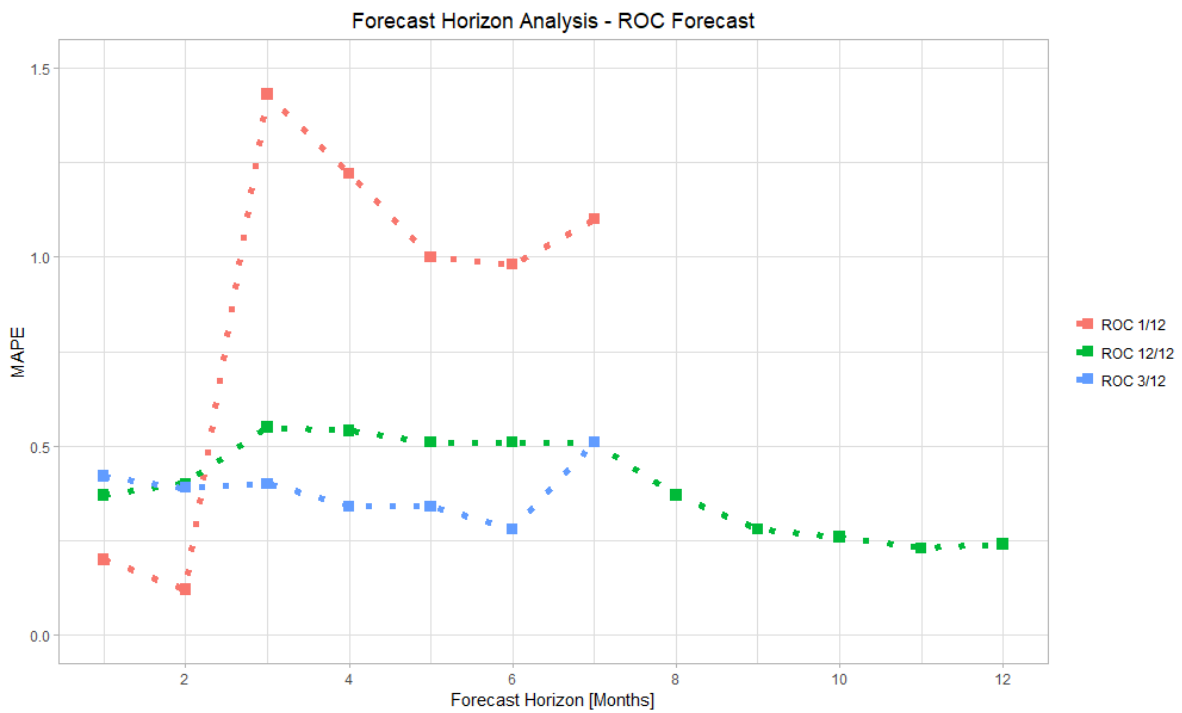


Figure 5.20.: ROC Forecast with Varied Forecast Horizon

Figure 5.21 shows the sales forecast. Similarly to figure 5.20 the forecast with the longer forecast horizons exceeding 7 months can not be assessed for the forecasts constructed using the $ROC_{1/12}$ and $ROC_{3/12}$. However, both are pretty constant with the exception

of reaching the minimum at 2 months which can be explained by the subsequent exclusion of indicators that only have a time lag of 2 months. The forecast constructed using the $ROC_{12/12}$ has the highest MAPE. However, the MAPE decreases with increasing forecast horizon. Interestingly, there is a very high outlier value for 8 months, which performed surprisingly badly. For 9 to 12 months the MAPE remains quite constant and relatively low.

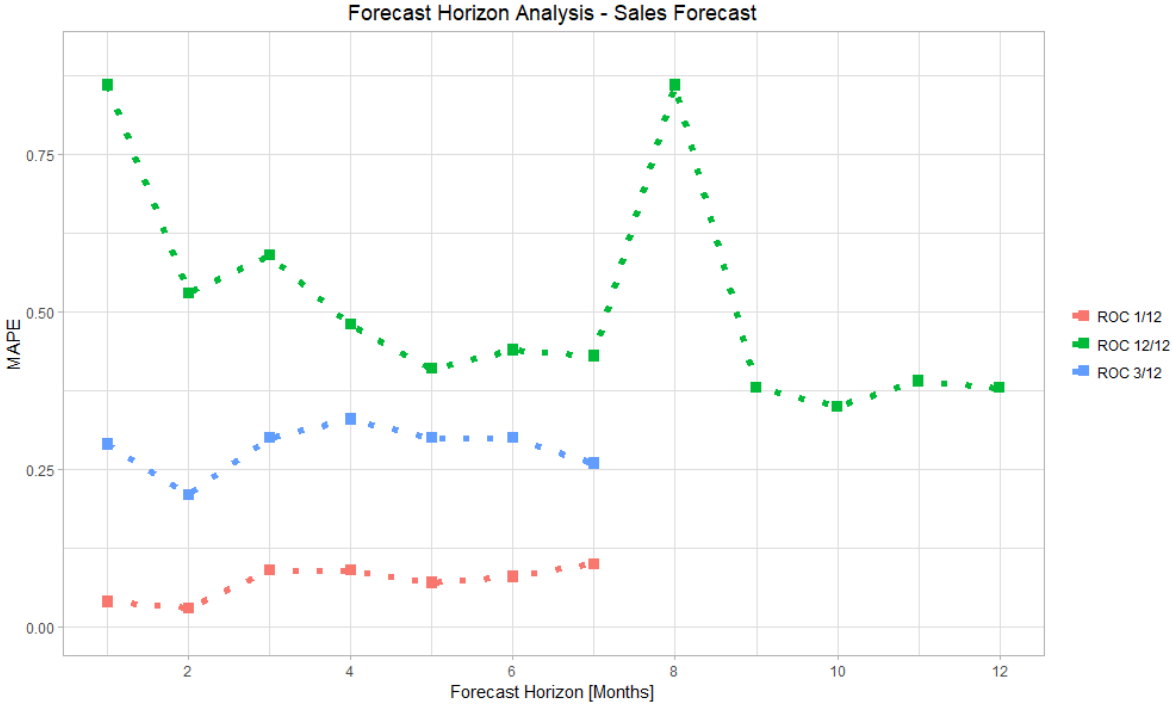


Figure 5.21.: Sales Forecast with Varied Forecast Horizon

5.2.3. Normalisation Method

Figure 5.22 shows the ROC forecasts' MAPE for different normalisation methods. The $ROC_{1/12}$ and the $ROC_{3/12}$ forecast has the MAPE minimum with the minimum / maximum normalization method. The $ROC_{12/12}$ forecast has the MAPE minimum with the integral normalization method. Interestingly, the min/max normalisation method would be the worst method for the $ROC_{12/12}$ forecast.

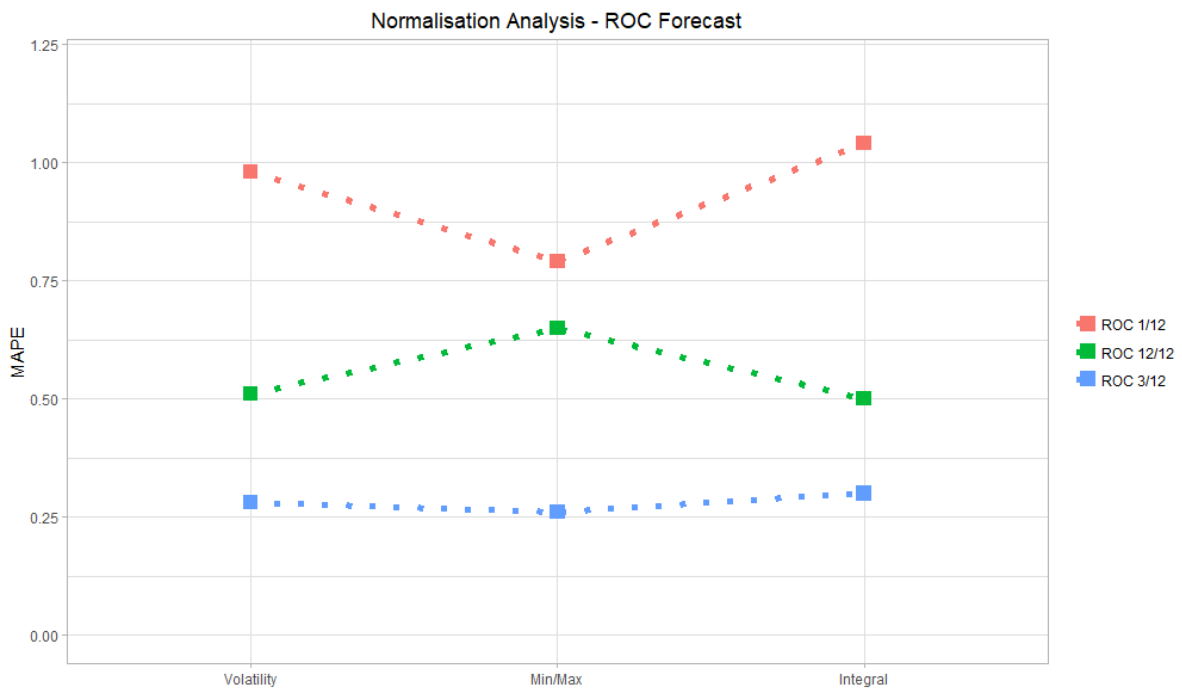


Figure 5.22.: ROC Forecast with Varied Normalisation Method

In figure 5.23 the sales forecasts are shown which were created with different ROCs. As it can be seen in figure 5.23 the MAPE values are almost identical for the different normalisation methods.

As shown earlier the MAPE for the $ROC_{1/12}$ is lower for the sales forecast than for the ROC forecast. The lowest MAPE values for the forecast constructed using the

$ROC_{1/12}$ were achieved with the minimum / maximum scaling, for the $ROC_{3/12}$ sales forecast there was no difference between the different normalisation methods and for the $ROC_{12/12}$ sales forecast the min/max method yielded again the best MAPE.

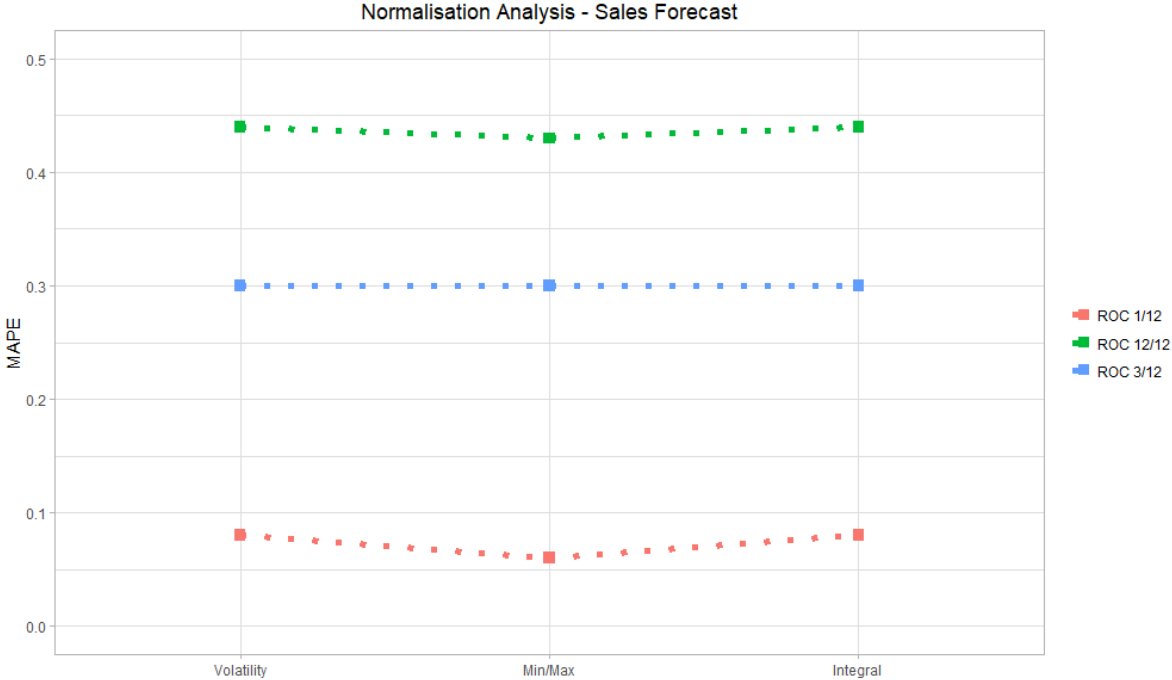


Figure 5.23.: Sales Forecast with Varied Normalisation Method

5.2.4. Correlation Method

Figure 5.24 shows the ROC forecasts created with different correlation methods. The $ROC_{1/12}$ forecast has very high MAPE values for the Pearson's correlation method and the Beta method, however, for the Kendall and the Spearman methods the MAPE is similar to the $ROC_{12/12}$. The MAPE values for the $ROC_{3/12}$ forecast and the $ROC_{12/12}$ forecast are nearly constant with the $ROC_{3/12}$ forecast having the lowest MAPE values of all the forecasts.

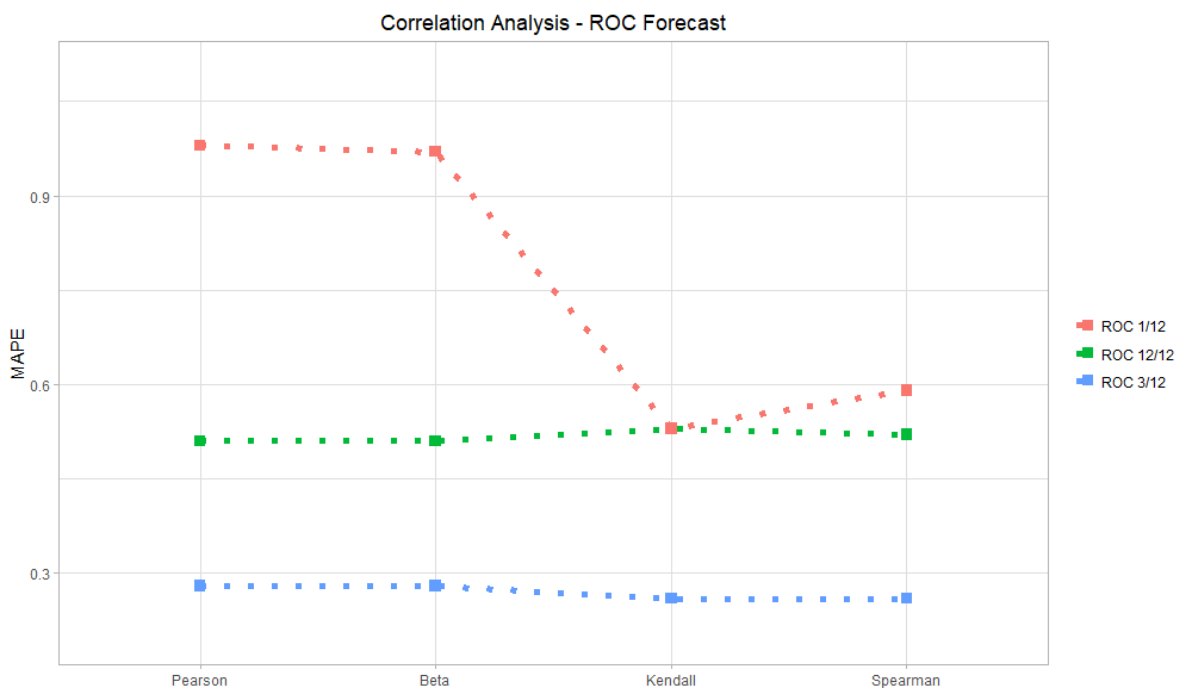


Figure 5.24.: ROC Forecast with Varied Correlation Method

Figure 5.25 shows the sales forecast. As it can be seen the sales forecast constructed with the $ROC_{1/12}$ has the lowest MAPE and remains quite constant. the sales forecast constructed with the $ROC_{3/12}$ has the best MAPE when using the Beta method and the $ROC_{12/12}$ sales forecast when using the Spearman method.

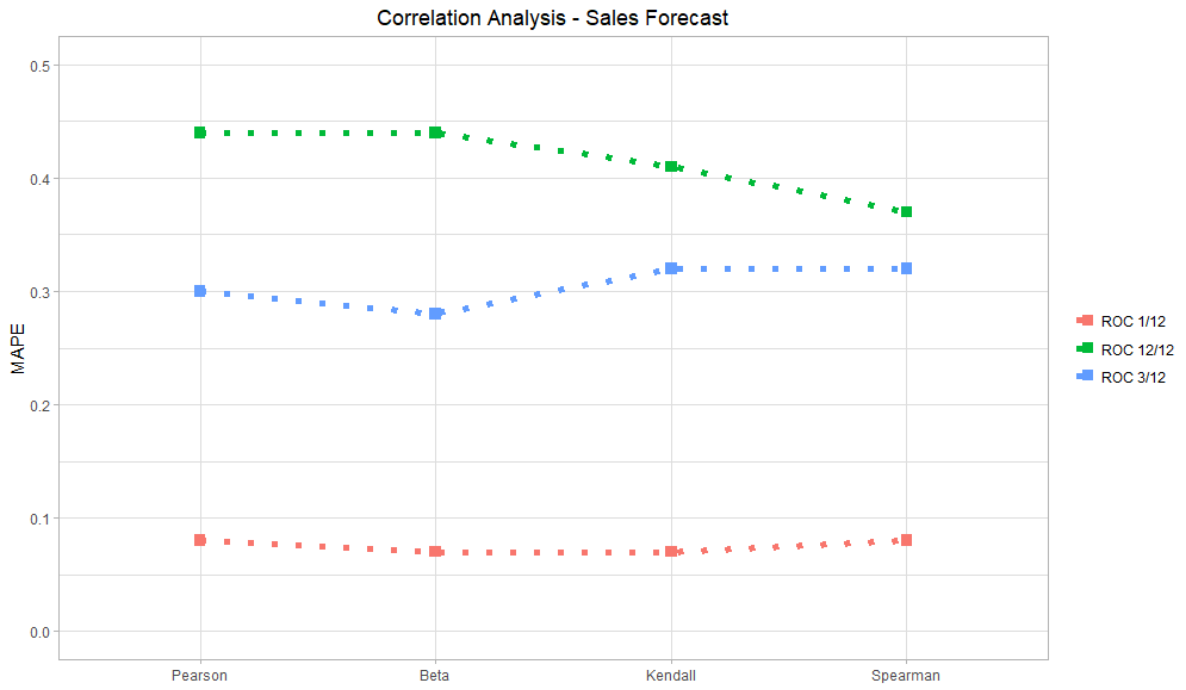


Figure 5.25.: Sales Forecast with Varied Correlation Method

5.2.5. Cross Correlation Value

The MAPE values for the different ROC forecasts with different maximal allowed cross-correlation values are shown in figure 5.26. The $ROC_{1/12}$ has a high MAPE at high maximal allowed cross-correlation which is something that is expected due to the problem of collinearity. However, the $ROC_{3/12}$ forecast and the $ROC_{12/12}$ forecast achieve a lower MAPE when the maximal allowed cross-correlation is increased. This surprising result can be explained by realising that the additional data smoothing results in higher observed cross-correlations without actually meaning that the indicators measure the same phenomenon. Therefore useful indicators would be excluded.

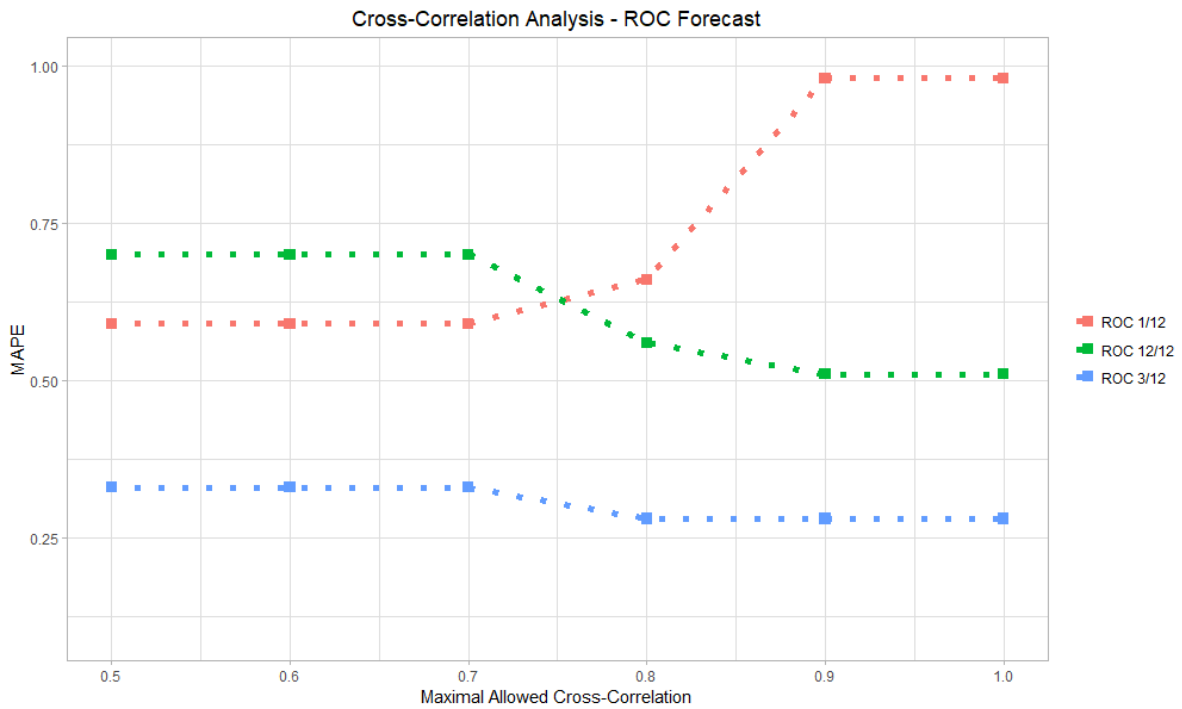


Figure 5.26.: ROC Forecast with Varied Cross-Correlation

Figure 5.27 shows the sales forecast for the different maximum allowed cross-correlations. Here it can be observed that the MAPE stays constant for the sales forecasts constructed with the $ROC_{3/12}$ and $ROC_{1/12}$. The sales forecast constructed with the $ROC_{12/12}$ is also quite constant with the exception of a peak at a maximal allowed value of 0.8 for the cross-correlation.

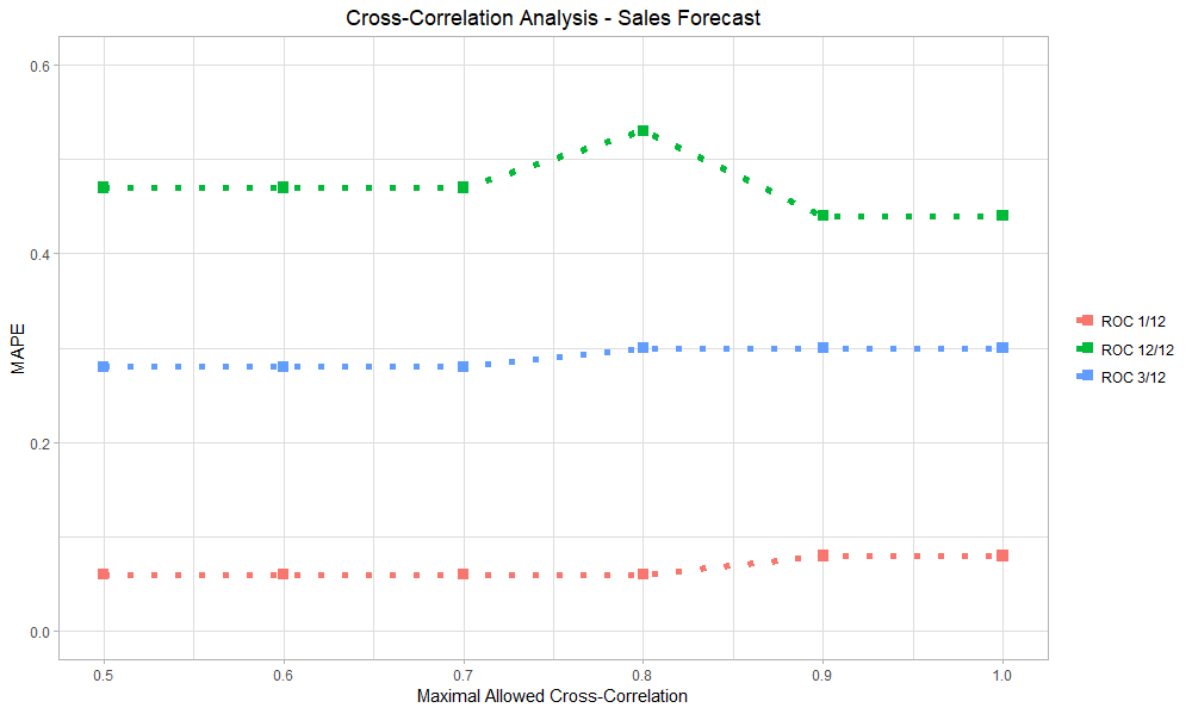


Figure 5.27.: Sales Forecast with Varied Cross-Correlation

5.2.6. Aggregation Method

Figure 5.28 shows the aggregation analysis for the ROC forecast. It can be seen that the MAPE remains fairly constant for the $ROC_{3/12}$ forecast. The $ROC_{1/12}$ forecast was best with the LASSO method, followed by the stepwise regression and the Hunschofsky method. The other methods performed quite poorly for the $ROC_{1/12}$ forecast. The $ROC_{12/12}$ forecast was best with the stepwise regression, followed closely by the LASSO method.

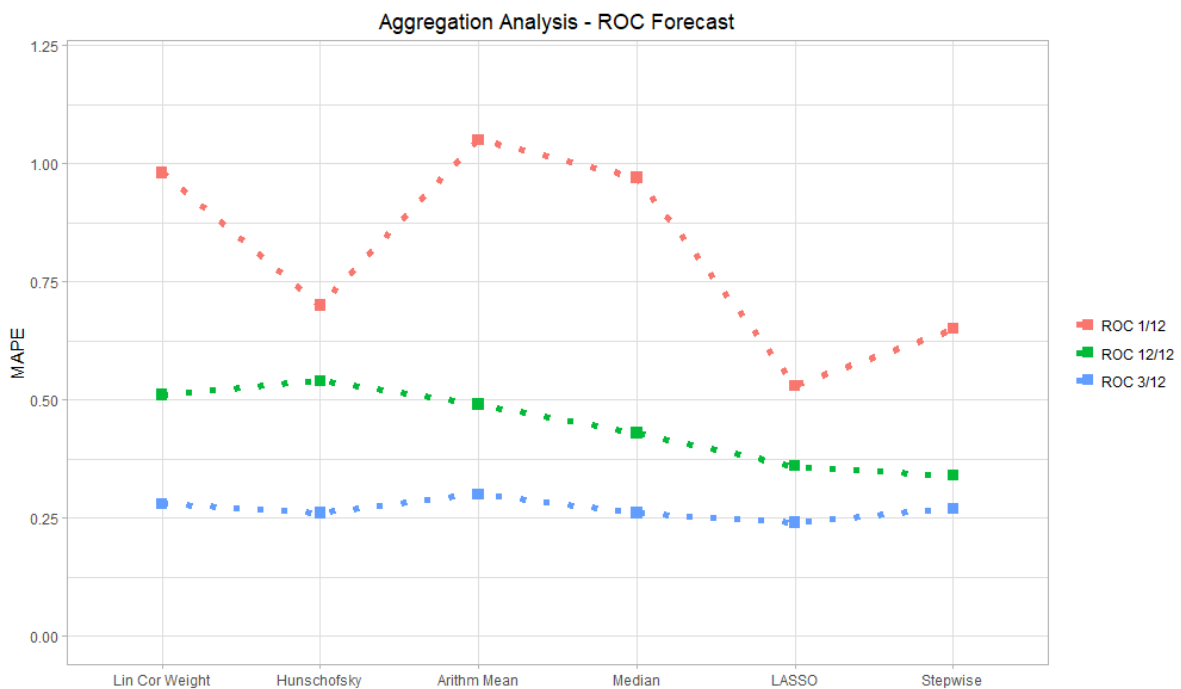


Figure 5.28.: ROC Forecast with Varied Aggregation Method

Figure 5.29 shows the sales forecast for varied aggregation methods. It can be seen that the different MAPE values vary greatly due to the different aggregation methods for the sales forecasts constructed using the $ROC_{3/12}$ or the $ROC_{12/12}$. Both show the same pattern with the LASSO method and the stepwise regression being the best methods by a quite large margin.

Interestingly, the MAPE of the sales forecast constructed using the $ROC_{1/12}$ was quite

constant which is very different to the MAPE of the $ROC_{1/12}$ forecast in figure 5.28.

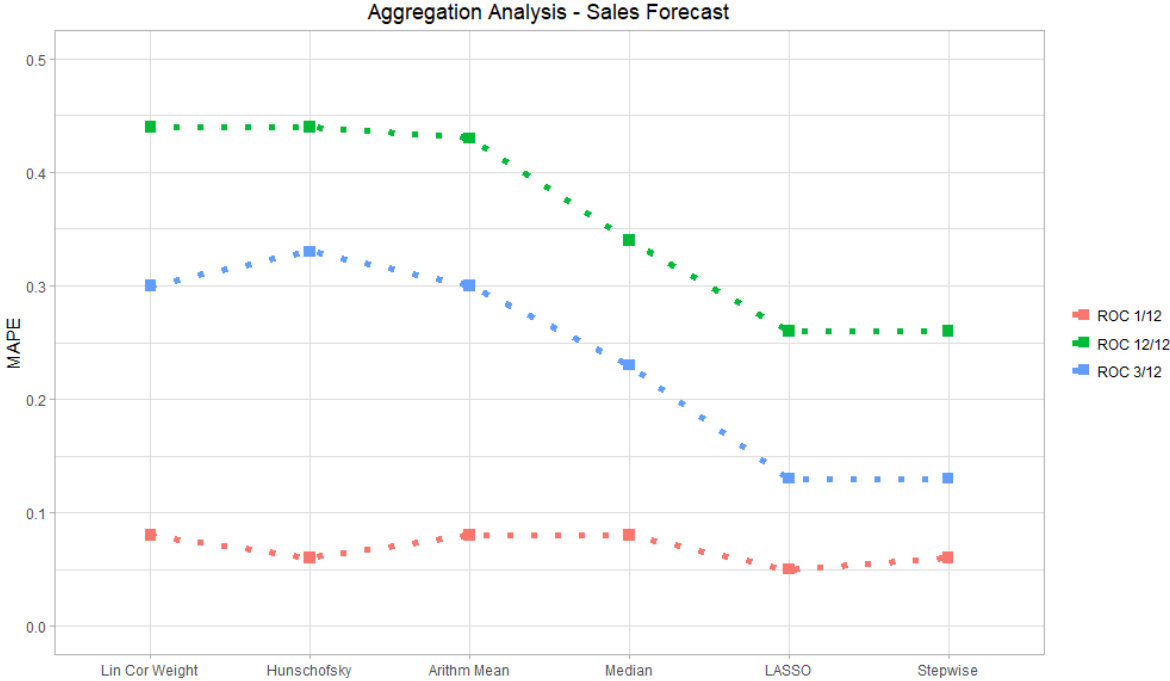


Figure 5.29.: Sales Forecast with Varied Aggregation Method

5.2.7. Summary Parameter Variation

In table 5.3 the best parameters for the ROC forecasts, differentiated into the $ROC_{12/12}$ forecast, the $ROC_{3/12}$ forecast and the $ROC_{1/12}$ forecast are shown.

Table 5.3.: Best Parameters for ROC Forecast

Best Parameters for ROC Forecast			
Method	Parameter ROC 12/12	Parameter ROC 3/12	Parameter ROC 1/12
In-Sample Period Start	2002-01-01	2006-01-01	2009-01-01
In-Sample Period End	2017-12-01	2017-12-01	2017-12-01
Forecast Horizon	11	6	2
Normalisation	Integral	Min/Max	Min/Max
Mean Normalisation	Yes	Yes	Yes
Correlation	Pearson/Beta	Kendall/Spearman	Kendall
Cross-Correlation	0.9-1	0.8-1	0.5-0.7
Aggregation	Stepwise	LASSO	LASSO

Table 5.4 shows the best parameters for the construction of sales forecasts using the $ROC_{12/12}$, the $ROC_{3/12}$ and the $ROC_{1/12}$.

Table 5.4.: Best Parameters for Sales Forecast

Best Parameters for Sales Forecast			
Method	Parameter ROC 12/12	Parameter ROC 3/12	Parameter ROC 1/12
In-Sample Period Start	2002-01-01	2001-01-01	2007-01-01
In-Sample Period End	2017-12-01	2017-12-01	2017-12-01
Forecast Horizon	10	2	2
Normalisation	Min/Max	All equal	Min/Max
Mean Normalisation	Yes	Yes	Yes
Correlation	Spearman	Beta	Beta/Kendall
Cross-Correlation	0.9-1	0.5-0.7	0.5-0.8
Aggregation	LASSO/Stepwise	LASSO/Stepwise	LASSO

The ideal length of the in-sample period varies quite substantially from 9 to 16 years. However, it is visible that the forecasts constructed using the $ROC_{12/12}$ prefer a longer in-sample period while the $ROC_{1/12}$ prefers a smaller period. The forecast horizon is exhibiting the same preference with the $ROC_{12/12}$ preferring a longer forecast horizon and the $ROC_{1/12}$ preferring a shorter forecast horizon.

Regarding the normalisation method the min/max normalisation is with one exception always the best method. The ideal correlation method varies substantially and a clear trend can not be observed. Similarly, the cross-correlation varies a lot, however the

$ROC_{12/12}$ seems to prefer a higher allowed cross-correlation. This could be due to a stronger smoothing effect on the data thus increasing the correlation between the individual indicators.

The aggregation method has a clear best method, and it is LASSO. With one exception all the best forecasts used LASSO. However, in half of the forecasts the best result would have also been obtained using stepwise regression. It should be pointed out, however, that for some parameters (not featured in this parameter variation) the LASSO and stepwise regression method can lead to overfitting of the composite index to the in-sample sales data and therefore create extremely implausible out-of-sample predictions.

6. Analysis and Evaluation of GNOSIS

The content of this chapter will serve to evaluate the developed software tool GNOSIS. For the proper evaluation it is necessary to first analyse the logical structure of GNOSIS from the perspective of the user. After visualising and describing this logical structure the evaluation can be done.

For this evaluation it is necessary to compare the realisation of the requirements posed in chapter 3. These requirements are structured into the different steps necessary while constructing any forecast. Namely, these are the construction, the calibration and the validation of a forecast. For this, the different construction methods, the different calibration methods and the different validation methods of GNOSIS will be aggregated from the different tabs of GNOSIS shown in chapter 5.1 and compared to the requirements in chapter 3.

Additionally, the requirements also contain the aspects of proper usability. Hence, the usability of GNOSIS will also be assessed and evaluated in this chapter.

6.1. Analysing the Demonstration

For the evaluation of GNOSIS, it is necessary to perform a comprehensive analysis of the software tool. For this task, a suitable modelling tool has to be picked to make the structure and logical flow of the forecast construction, calibration and validation with GNOSIS visible.

For this task, the UML already explained in chapter 4.2 will be used to depict the inner workings of GNOSIS. The structure of this chapter will follow the structure of chapter 5.1, as it is necessary to depict the logical structure of GNOSIS from the perspective of the user and not the structure of the code.

Only if the logical structure of GNOSIS makes sense from the perspective of a user, it can be considered a successful artefact. If the structure is not intuitive or does not allow the user to change the different parameters at the right point in time, it will fail due to missing functionality and usability.

6.1.1. Data Acquisition

Figure 6.1 shows the data acquisition structure of GNOSIS. After GNOSIS has been started, the user must load data into GNOSIS. These data are on the one hand the individual sales data and on the other hand the external macroeconomic data. The uploaded data is in a CSV format. For this reason, it must be possible to set the correct CSV format in GNOSIS. After the correct CSV format is set there is a first branch.

The user has to know which indicators can represent a ROC and which not. The indicators that represent a ROC must be selected by the user. Then the user can either start the next step, this is the ROC statistics, or the sales data plot can be viewed and also the various indicator plots can be seen. After the user has made all the settings for the data acquisition, the data for the ROC statistics is made available and the ROC statistics step follows.

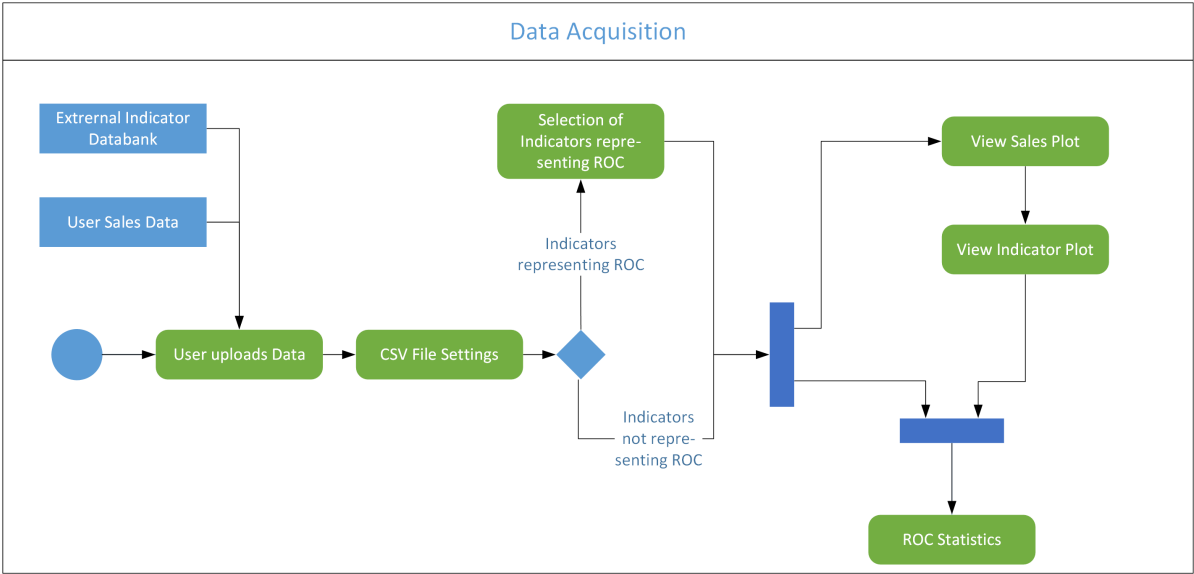


Figure 6.1.: GNOSIS Structure - Data Acquisition

6.1.2. ROC Statistics

In order to make GNOSIS concise, different data statistics are shown as graphs in the ROC statistics step (Figure 6.2). The user must first determine the various ROCs with which GNOSIS will subsequently perform the calculations. The user is also offered the opportunity to understand the sales more precisely with a logistic regression of the sales values.

In addition, the user can see with the help of the distribution plot how many times a particular ROC has been reached and how the sales ROCs are distributed. As the last important step, the user can again look at a box plot of the indicators. This is important to check once again whether the respective indicators have been set to represent a ROC or not. The selected ROC methods are finally passed to GNOSIS and the user can continue with the data normalisation step.

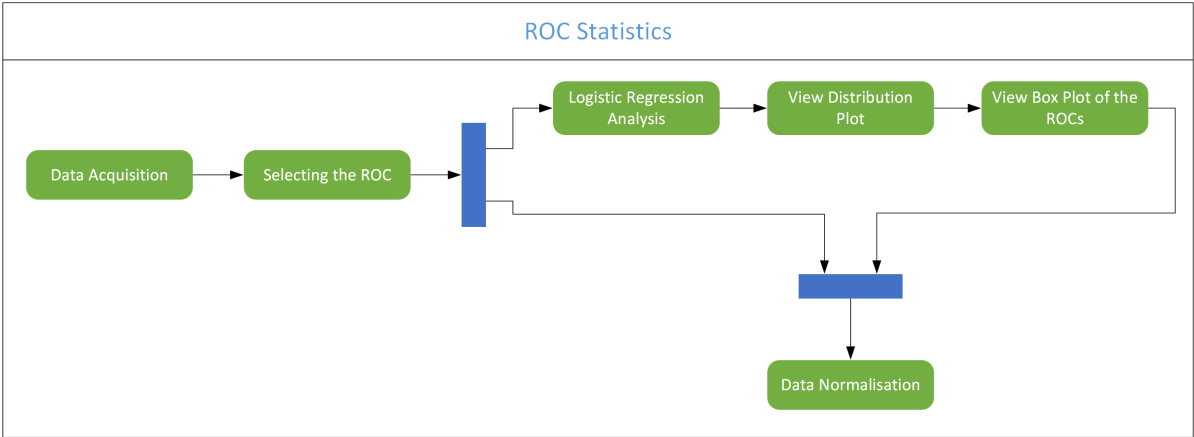


Figure 6.2.: GNOSIS Structure - ROC Statistics

6.1.3. Data Normalisation

In figure 6.3 the structure of the data normalisation step is shown. In this step, the user has the opportunity to first define a proper in-sample period. Afterwards, the user can select with which amplitude normalisation method the indicators should be normalised to fit the sales data.

In addition, the possibility to deselect the mean normalisation was integrated in GNOSIS. This could provide further information to the user for some indicators and forecasts. With each selection, the user can directly see a plot showing the ROCs diagram which interactively varies according to the user's selection. Finally, the settings are passed to GNOSIS and the user can proceed with the indicator correlations.

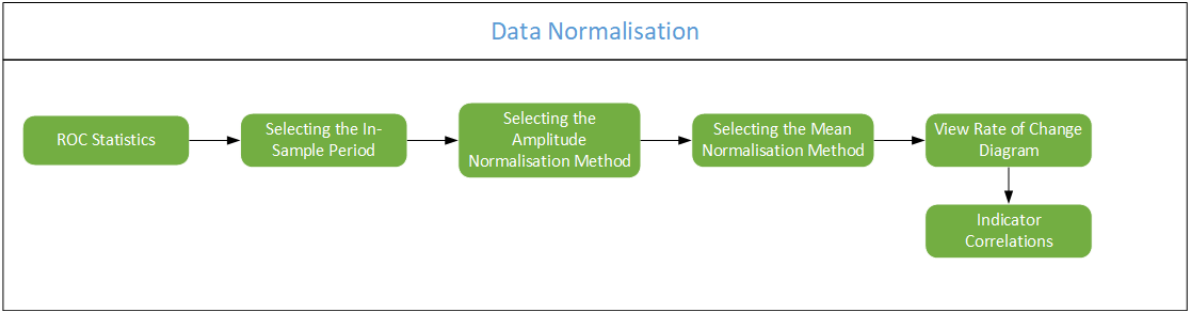


Figure 6.3.: GNOSIS Structure - Data Normalisation

6.1.4. Indicator Correlations

The indicator correlations can be seen in figure 6.4. As a first step, the user can set the forecast horizon with which the forecast will be calculated. By default, the forecast horizon is set to 6 months. Afterwards the correlation analysis method has to be chosen. In the next step, the maximum allowed cross-correlation can be set. The default for the maximal allowed cross-correlation is 0.9. After these steps, the user can see a table and the correlation plots.

Here GNOSIS offers the user the opportunity to decide for himself whether all parameters have been set correctly. If this is not the case, the user can iteratively change the forecast horizon, use a new correlation method or vary the maximum allowed cross-correlation. If, in a later step, the forecast has too many errors, these settings can also be adjusted later. After the user has adjusted the settings, the composite index construction can be started.

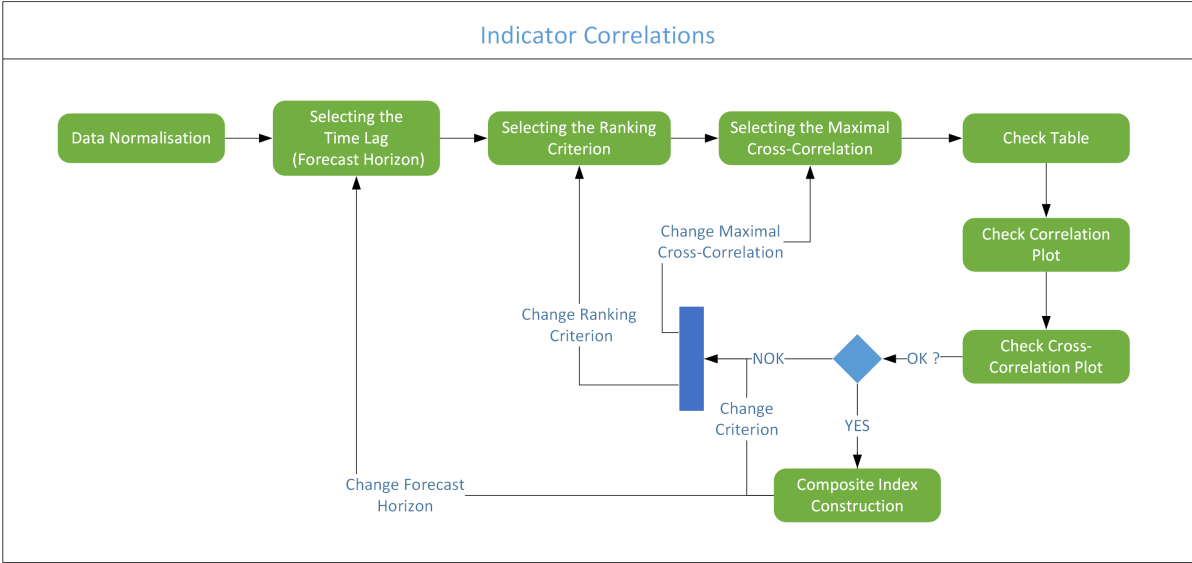


Figure 6.4.: GNOSIS Structure - Indicator Correlations

6.1.5. Composite Index Construction

Figure 6.5 shows the composite index construction. As a first important step, the user can choose whether the composite index should be made with a heuristic custom selection of the indicators and weights, or whether it should be calculated using a mathematically based method. If the forecast is method based, the user can select the number of indicators, which is by default the maximal number of indicators. In this step, the user must also select the aggregation method with which the composite index will be created.

No matter whether the user creates the composite index method-based or custom-based, there is the possibility to see a table with all indicators and their weights. In addition, the user is shown the in-sample period with the composite index. After this step, the user can decide if the created composite index should be used in the further calculation. If the composite index is not precise enough, a new custom selection can be performed. Likewise, the number of indicators and the aggregation method can also be re-selected. If the forecast has too many errors in the later calculation, these settings can also be adjusted later. If the composite index is correct, the ROC forecast is calculated.

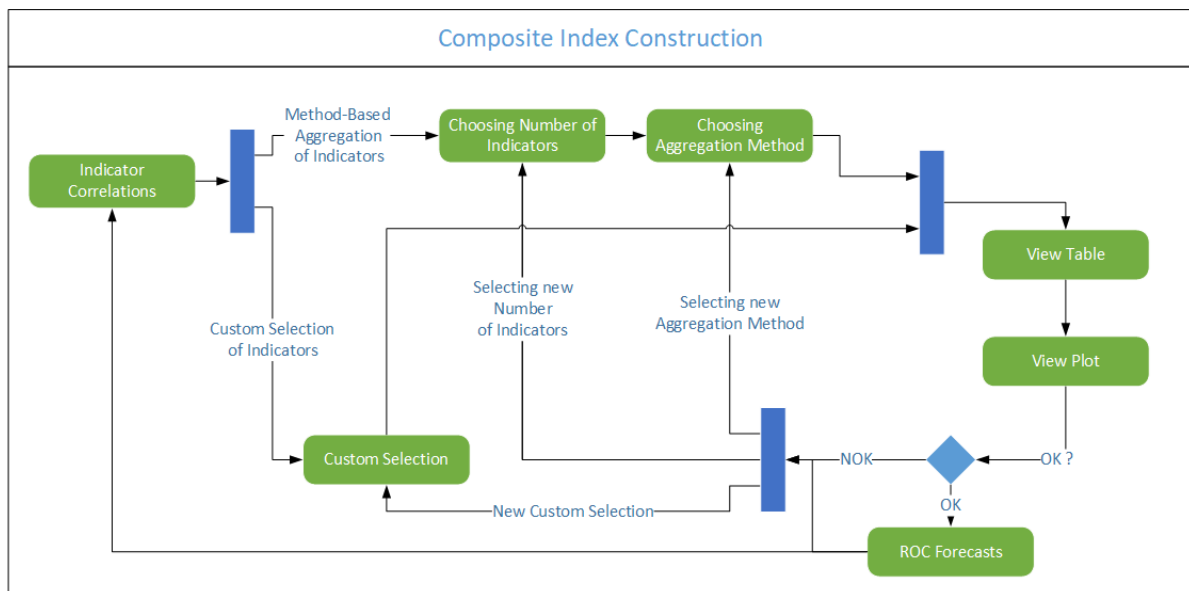


Figure 6.5.: GNOSIS Structure - Composite Index Construction

6.1.6. ROC Forecasts

Figure 6.6 shows the ROC forecast. Here it is important that the user has the opportunity to determine the accuracy of the forecasts. In this case there is the possibility to perform in-sample and out-of-sample tests. The user can set which in-sample and out-of-sample tests should be used to calculate the accuracy tests. These tests are displayed in a chart combined with the ROC forecast. In addition, the values of the accuracy tests are displayed in a table. Furthermore, a logistic univariate ROC prediction is made. If the ROC forecast has too many errors, parameters can be adapted in the previous steps. If the ROC forecast is right, the sales forecast can be started.

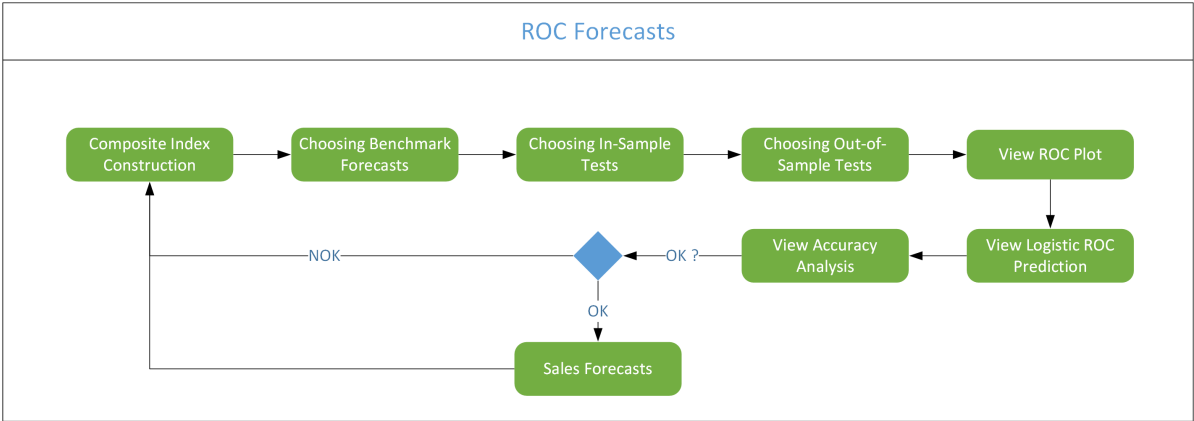


Figure 6.6.: GNOSIS Structure - ROC Forecasts

6.1.7. Sales Forecasts

The sales forecast can be seen in figure 6.7. The user must be able to perform out-of-sample accuracy tests for the sales forecast. Also, univariate benchmark forecasts are created. These are necessary for the user, so that it can be judged, if the forecast is better or worse than a benchmark forecast.

The accuracy of the forecasts and the averaged sales forecast is also shown to the user in separate tables. Then the user can decide whether the forecast seems suitable or whether parameters have to be varied in the previous calculations. Finally, the user can download reports.

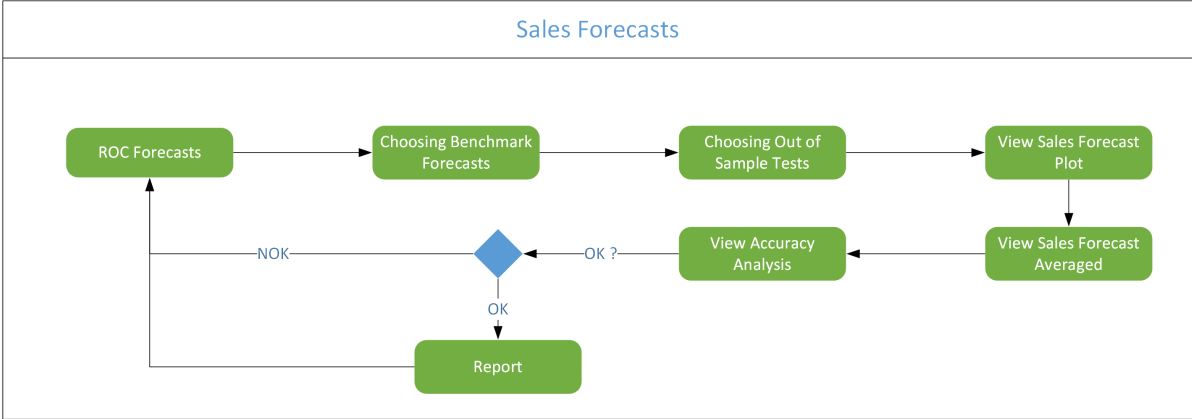


Figure 6.7.: GNOSIS Structure - Sales Forecasts

6.1.8. Report

Figure 6.8 shows the different possibilities which are available in GNOSIS to download data. The user can decide if a report should be downloaded or not. If the user wants to download a report, it can be chosen to download the report in a Microsoft Word, PDF or HTML format. The user can also decide whether the values of the forecasts and the accuracy tests should be downloaded. This download is made in a Microsoft Excel file format. After this, there are no additional steps of GNOSIS available to the user.

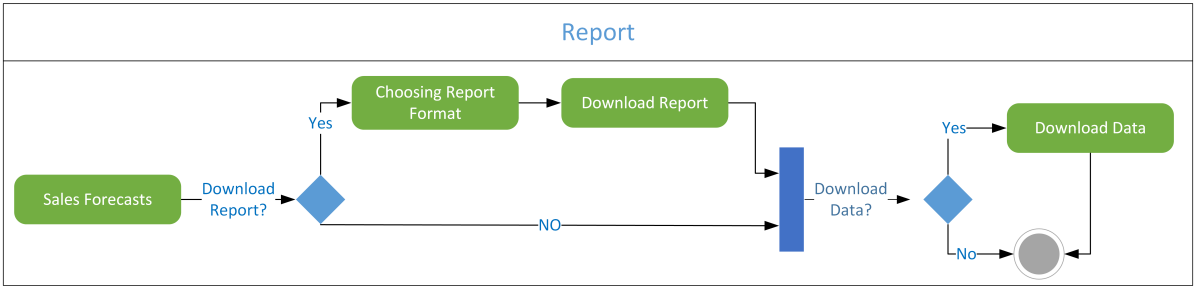


Figure 6.8.: GNOSIS Structure - Report

6.2. Fulfilment of Requirements

To summarise the analysis of GNOSIS in section 6.1, this section will compare the requirements of GNOSIS established in chapter 3 with the realisations seen in section 5 and the analysis of chapter 6.1. For brevity reasons this will mostly be done with the help of tables comparing the requirements with the realisations.

The result of the comparison will be depicted with checkmarks or X-marks. One checkmark signifies that the requirement is fulfilled, two checkmarks signify that the requirement is fulfilled plus additional functionality is implemented in GNOSIS and an X-mark would mean that the requirement is not fully fulfilled.

This chapter will be structured analogue to chapter 3 to make the comparison more intuitive. Hence, this chapter will be structured in the three main activities for forecasting:

- construction,
- calibration, and
- validation.

Additionally, the usability will also be addressed, since it is useless to design and develop a software tool, if it is not used due to missing usability.

6.2.1. Construction

In table 6.1 it is clearly visible that all the requirements of section 3.2 concerning the construction phase are fulfilled by GNOSIS. The forecast horizon has even more functionality as it was required, with a maximal settable forecast horizon in GNOSIS of 24 months instead of the 12 months specified in the requirements.

Table 6.1.: Evaluation of GNOSIS -Construction

Type of Requirement	Requirement	Realisation
Forecast Horizon	1 - 12 Months	✓✓
Rate of Change	ROC 12/12	✓
Rate of Change	ROC 3/12	✓
Rate of Change	ROC 1/12	✓
Correlation Analysis	Pearson's Correlation Coefficient	✓
Correlation Analysis	Spearman's Rho	✓
Correlation Analysis	Kendall's Tau	✓
Correlation Analysis	Beta	✓
Cross Correlation	Settable Maximal Value	✓
Composite Index Construction	Arithmetic Mean	✓
Composite Index Construction	Weights According to Correlation	✓
Composite Index Construction	Weights Described by Hunschofsky	✓
Composite Index Construction	Median	✓
Composite Index Construction	LASSO	✓
Composite Index Construction	Stepwise Regression	✓
Sales Forecast	Calculated with ROC 12/12 forecast	✓
Sales Forecast	Calculated with ROC 3/12 forecast	✓
Sales Forecast	Calculated with ROC 1/12 forecast	✓
Logistic Regression	Univariate Forecast Exceeding Forecast Horizon	✓

6.2.2. Calibration

The evaluation of the implementation of the calibration method requirements from section 3.3 shown in table 6.2 shows that all the requirements were implemented into GNOSIS. The normalisation methods have more functionality than required.

Table 6.2.: Evaluation of GNOSIS - Calibration

Type of Requirement	Requirement	Realisation
Selection of Indicators for ROC Calculation	Excluding certain Indicators from the ROC Calculation	✓
In-sample Period	Selectable by the User	✓
Normalisation Method	Mean-Variance Normalisation	✓✓
Normalisation Method	Min-Max Normalisation	✓✓
Correlation	User Can Pick the Type of Correlation	✓
Cross-correlation	User Can Set the Maximal Cross-Correlation	✓
Composite Index Construction	User Can Set the Number of Indicators	✓
Composite Index Construction	User Can Assign Weights Manually	✓

It was required, that the normalisation method could be split into a mean and variance normalisation. Additionally, the normalisation should not normalise all indicators and the sales to have a mean of zero. Therefore, the normalisation methods had to be slightly adapted, to normalise the indicators to exhibit the same characteristics or magnitudes as the sales.

All the normalisation methods in GNOSIS can be coupled with a mean normalisation, where the mean of the indicators will be set to the mean of the sales. For the normalisation of the amplitude three normalisation methods were implemented. The mean-variance normalisation was adapted to normalise the indicators to have the same variance as the sales.

The min/max normalisation was adapted to adjust the amplitude of the indicators to fit into the range of the sales, meaning that the largest absolute value of an indicator is not larger than the respective counterpart of the sales. Finally, the integral scaling in GNOSIS is a normalisation to adjust the sum of the total values of the indicators to be equal to the sum of the total values of the sales.

6.2.3. Validation

The evaluation of the validation methods described in section 3.4 can be seen in table 6.3. Here it is clearly visible that GNOSIS has fulfilled all the requirements concerning validation methods. However, it should also be noted that there is no additional functionality regarding the accuracy measures, other than required a priori, since the implemented ones seem to suffice.

Table 6.3.: Evaluation of GNOSIS - Validation

Type of Requirement	Requirement	Realisation
In-sample Testing	Coefficient of Determination	✓
In-sample Testing	Root Mean Squared Error	✓
Out-of-sample Testing	Mean Average Percentage Error	✓
Out-of-sample Testing	Mean Directional Accuracy	✓
Out-of-sample Testing	Mean Absolute Scaled Error	✓
Out-of-sample Testing	Geometric Mean Relative Absolute Error	✓
Benchmark Forecasts	Naïve	✓
Benchmark Forecasts	Seasonal Naïve	✓
Benchmark Forecasts	Random Walk	✓
Benchmark Forecasts	Auto ARIMA	✓

6.2.4. Usability

The evaluation of the usability is not as simple, as a mere checklist that has to be fulfilled. The usability has many different aspects, that all have to be considered in order to create a software tool with a high usability. Gumussoy (2016) lists many different questions ordered by their level of impact. Those questions can be seen in table 6.4, table 6.5 and table 6.6. They address the usability catastrophes and major problems. For brevity concerns, the tables featuring the minor usability problems and cosmetic problems are in appendix E.

These questions will be used to assess the usability of GNOSIS by ranking it for each criterion from one to five, where a one means that GNOSIS fulfils the criterion completely, a two means that GNOSIS nearly fulfils the criterion during all the steps, a three means that it only fulfils the criterion partially or not during all the steps, a four that it only sometimes or to a small degree fulfils the criterion, and a five means that GNOSIS does not at all fulfil the criterion. The evaluation was performed by the authors of this thesis. The questions that were not applicable to GNOSIS were left out.

Table 6.4.: Evaluation of GNOSIS - Usability Catastrophe

Severity	Usability Criterion	GNOSIS
Usability Catastrophe	Are window operations easy to learn and use?	1
	If the system has deep (multilevel) menus, do users have the option of typing ahead?	1

In table 6.4 it is clearly shown that GNOSIS does not exhibit any usability catastrophe. In table 6.5 GNOSIS fully exhibits a high usability for 16 out of 18 questions, and has only two questions where it is neutral or slightly positive.

Table 6.5.: Evaluation of GNOSIS - Major Usability Problems

Severity	Usability Criterion	GNOSIS
Major Usability Problems	Are field level prompts provided for data entry screens?	3
	Do related and interdependent fields appear on the same screen?	1
	Do the selected colors correspond to common expectations about color codes?	2
	Are icons concrete and familiar?	1
	When prompts imply a necessary action, are the words in the message consistent with that action?	1
	Do menu choices fit logically into categories that have readily understood meanings?	1
	Are menu choices ordered in the most logical way, given the user, the item names, and the task variables?	1
	Does the system provide visibility: that is, by looking, can the user tell the state of the system and the alternatives for action?	1
	Is the current status of icon clearly indicated?	1
	Is there visual feedback in menus or dialog boxes about which choices are selectable?	1
	Is the menu naming terminology consistent with the user's task domain?	1
	Does every display begin with a title or header that describes screen contents?	1
	Is there a consistent icon design scheme and stylistic treatment across the system?	1
	Is a single, selected icon clearly visible when surrounded by unselected icons?	1
	Are icons labeled?	1
	Are there salient visual cues to identify the active window?	1
	Are field labels consistent from one data entry screen to another?	1
	Do abbreviations follow a simple primary rule and, if necessary, a simple secondary rule for abbreviations that otherwise would be duplicates?	1

Table 6.6 shows questions regarding help functions or error prevention for major usability problems and usability catastrophes. Here GNOSIS has three negative evaluations. First, the help function is not displayed since there is no help function in GNOSIS. Second, the error messages do not inform the user of an error's severity and third, an "UNDO" function is not implemented in GNOSIS. Those are definitely features that

would be sensible to implement, when updating GNOSIS. However, since GNOSIS does not exhibit any major usability problems or catastrophes, this in turn also not a major usability problem. Overall, it can be said that GNOSIS exhibits a very high usability, although with some small room left for improvement.

Table 6.6.: Evaluation of GNOSIS - Help

Severity	Usability Criterion	GNOSIS
For Usability Catastrophes and Major Usability Problems	Are data entry screens and dialog boxes supported by navigation and completion instructions?	1
	Is the help function visible; for example, a key labeled HELP or a special menu?	5
	If menu choices are ambiguous, does the system provide additional explanatory information when an item is selected?	1
	Do error messages inform the user of the error's severity?	5
	Does the system prevent users from making errors whenever possible?	1
	Are menu choices logical, distinctive, and mutually exclusive?	1
	If the database includes groups of data, can users enter more than one group on a single screen?	1
	Is the menu choice name on a higher-level menu used as the menu title of the lower-level menu?	1
	Can users easily reverse their actions?	2
	Is there an "undo" function at the level of a single action, a data entry, and a complete group of actions?	5
	Are menus broad (many items on a menu) rather than deep (many menu levels)?	1

7. Discussion

In this chapter the software tool GNOSIS will be discussed. The first section will be a short summary and discussion of the structure and functionality of GNOSIS. The second section will be a discussion concerning the forecasts that can be constructed, calibrated and validated using GNOSIS. Also, the different possible parameters will be mentioned as well as their influence on the predictive accuracy of the forecast. The last section will feature interesting problems for further research, which would have gone beyond the scope of this thesis.

7.1. The Artefact GNOSIS

GNOSIS is designed as a tool for sales forecasting. To improve the accuracy of the sales forecasts, GNOSIS uses macroeconomic indicators which represent business cycles. Additionally, GNOSIS has implemented tools for predictive accuracy analysis. To make such a program useful for a user, GNOSIS was designed in such a way that a user with average IT knowledge can use the full functionality.

The structure of GNOSIS is designed to be logical and intuitive to a user. The different menu points are therefore as follows:

- Data Acquisition,
- ROC Statistics,
- Data Normalisation,
- Indicator Correlations,
- Composite Index Construction,
- ROC Forecast,
- Sales Forecast, and
- Report.

The functionality requirements regarding the different methods during construction, calibration and validation were all met, as chapter 6.2.1,6.2.2 and 6.2.3 show. The usability of GNOSIS is also very high, as shown in chapter 6.2.4. Whenever the user's choices have an impact on the data, GNOSIS shows the impact using plots, tables or heat maps. This is a key feature since it allows the user to judge the impact of a parameter choice on the data right away, as opposed to many different already available tools which just show the finished sales forecast to the user.

Additionally to the requirements, GNOSIS has implemented two different download functions. The first download function is a report either downloadable as a PDF file, a

Microsoft Word file or as an HTML file. This report lists all the parameter choices of the user and all the resulting plots and forecasts. It can be seen as a complete summary of all the actions the user performed and all the outputs. The second download function is a Microsoft Excel file which features the time series values of the composite index and ROC forecast, the time series values of the sales forecast, the predictive accuracy analysis results, and the logistic ROC prediction.

7.2. Forecasts Generated Using GNOSIS

The different ROCs used for the construction of the ROC forecasts and the sales forecasts are as follows:

- $ROC_{12/12}$,
- $ROC_{3/12}$, and
- $ROC_{1/12}$.

The parameter variation and analysis of the MAPE of the ROC forecasts and the sales forecasts in chapter 5.2 showed that the choice of ROC used for the construction of the forecast is an important parameter that changes the predictive accuracy drastically. The best ROC for the construction of a ROC forecast is the $ROC_{3/12}$, closely followed by the $ROC_{12/12}$. The $ROC_{1/12}$ produces poorly performing ROC forecasts, however, it is the best ROC for the construction of a sales forecast.

Hence, it is important that the user chooses the ROC used for the construction of a forecast depending on the type of forecast the user wants to generate. If the user wants to generate a ROC forecast, the user should choose the $ROC_{3/12}$ or the $ROC_{12/12}$, but if the user wants to construct a sales forecast the $ROC_{1/12}$ would be the best choice.

The parameter variation revealed that the ideal correlation method is not constant and a clear recommendation can not be made. However, it could sometimes be possible for an expert user to choose the best correlation method by examining the nature of the indicators. The normalisation methods did not have a very big impact on the predictive accuracy, but in most cases the min/max normalisation outperformed the rest. The forecast horizon up to 12 months was tested and the findings suggest that it is possible to construct a forecast using the $ROC_{12/12}$ with a forecast horizon of 12 months, which still exhibits a high predictive accuracy. The forecasts using the $ROC_{3/12}$ and $ROC_{1/12}$ could not be constructed for a forecast horizon bigger than seven months due to missing indicators with this time lag in their respective ROC values.

The ideal cross-correlation varied depending on the ROC used for the forecast. For a forecast using the $ROC_{12/12}$ the best cross-correlation value would be between 0.9 and

1, whereas the other forecasts often performed better with a cross-correlation of 0.5-0.8. Additionally, the aggregation method was tested and the findings suggest that the LASSO method is the most suitable for achieving a high predictive accuracy. This is in line with the approaches found in the state-of-the-art literature which also suggest that the LASSO method is the best aggregation method for constructing a composite index.

The ideal length of the in-sample period is also different for the different ROCs. Forecasts using or predicting the $ROC_{12/12}$ performed better with a long in-sample period of 16 years, whereas forecasts using or predicting the $ROC_{1/12}$ performed best with in-sample periods of nine to eleven years. The $ROC_{3/12}$ forecast was also performing well with a rather small in-sample period of 12 years, however, the sales forecast using the $ROC_{3/12}$ had the smallest MAPE with an in-sample period of 17 years.

The findings of the parameter variation did deliver the best parameters for a high predictive accuracy, however, those findings are only applicable to the sales data of ARC and the indicators used for the prediction. If one wants to construct a forecast using GNOSIS, it is highly recommended to perform a separate parameter variation.

Additionally, it was observed that the sales forecast has an interesting property concerning its monthly values. Due to the calculation of the sales forecast via the ROC forecast, the sales forecast becomes indirectly predicted and exhibits an oscillating behaviour. The monthly values are thus not exact with the exception of the forecast constructed using the $ROC_{1/12}$ which does not exhibit such a behaviour. However, all the sales forecasts produce a quite accurate total value of the sales in the following months.

One feature that GNOSIS has implemented, not commonly found in sales forecasting, is a prediction based on logistic regression. Here, an event can be defined and historically calibrated to generate probabilities for this event to occur. GNOSIS defines these events to be a ROC below -20% or a ROC above 20% and delivers the probabilities for those events to occur after the forecast horizon of the actual sales forecast. Hence, the user gets additional information, that the sales forecast alone would not have been able to provide.

7.3. Outlook

For further research it would be interesting to test GNOSIS with the sales data of other companies and with other macroeconomic indicators. It could be possible, that there are industry and possibly firm size dependent macroeconomic indicators which can be used for sales forecasts with a high predictive accuracy.

If this is the case, it would be possible to create sets of macroeconomic indicators directly implemented into GNOSIS, so that the user just has to upload the sales data and select the industry and region of the company to automatically get a set of suitable macroeconomic indicators for the construction of a sales forecast.

An interesting problem for further research would be the influence of supply chain forecasting on the predictive accuracy of the sales forecasts and how to implement it into GNOSIS. It could be possible to implement the forecasts of the supplier and customer as two additional indicators, however, this would have to be tested for its influence on the predictive accuracy of the sales forecast.

Additionally, it would be interesting to test whether forecasts generated with GNOSIS suffer from the same problems as forecasts not generated with GNOSIS, namely the high judgemental changes after the construction and calibration of a forecast. This is probably not the case, since GNOSIS is built with the intent to give the user many options and implement experts' knowledge in the selection of indicators in order to make judgemental adaptations after the construction and calibration undesirable. However, this would have to be tested in a study.

8. Conclusion

Following the Design Science Research Methodology, this thesis has performed an in-depth literature review on sales forecasts using business cycles. This is an important step to make sure that the resulting artefact is solving a problem that is not solved yet. In this case a software tool for the predictive analysis of sales forecasts constructed using macroeconomic indicators was defined as the desired artefact.

Based on the gaps found in the literature, a list of requirements was created which the artefact has to adhere to in order to be considered a successful artefact. These requirements were different construction methods, different calibration methods, and different validation methods. Additionally, there were also requirements concerning the usability of the artefact, since it is useless if nobody uses it due to it lacking usability.

For the realisation of the requirements, it was necessary to choose a software. Here, R and its package “Shiny” were deemed as the most suitable choices and were used for the development of the software tool GNOSIS. R is free of charge and proved to be a good choice due to the vast amount of free packages which can be used for statistical analysis of time series data.

GNOSIS was then demonstrated and tested using the real sales data of a real company and using expert-picked macroeconomic indicators. The evaluation showed that GNOSIS fulfils all the requirements created prior to the design and development, with the only constraint that there is a small room for improvements in the help function of GNOSIS, which is only concerning the usability.

A parameter variation testing all the different functionalities of GNOSIS was also performed and discussed. Hence, GNOSIS is a valuable tool for the construction, calibration and validation of sales forecasts using time-lagged macroeconomic indicators.

However, some gaps found in the literature were not solved. Namely, the selection of indicators from big databases would be improved if there was a way of pooling them for different industries and regions to allow a user to just use a predefined set of indicators right away. Also, GNOSIS does not address the topic of collaborative forecasting in a supply chain, which would be an interesting extension of the functionality of GNOSIS. A study examining the use of the forecasts generated with GNOSIS would also be a valuable contribution, especially to examine whether the forecasts that are constructed, calibrated and validated with GNOSIS are less subject to judgemental adaptations due to the high level of user involvement during the construction, calibration and validation phase.

Abbreviations and Acronyms

AIC	Akaike Information Criterion
ARC	A Real Company
BCM	Business Cycle Management
BIC	Bayesian Information Criterion
CI	Composite Index
DSRM	Design Science Research Methodology
e.g.	exempli gratia, for example
et al.	et alii (<i>m</i>), et aliae (<i>f</i>) or et alia (<i>n</i>), among others
GMRAE	Geometric Mean Relative Absolute Error
GNOSIS	Generating New Opportunities for Selection of Indicators for Sales Forecasting
GPL	GNU General Public License
i.e.	id est, that is
LASSO	Least Absolute Shrinkage and Selection Operator
MDA	Mean Directional Accuracy
MAPE	Mean Absolute Percentage Error
MASE	Mean Absolute Scaled Error
RMSE	Root Mean Squared Error
ROC	Rate Of Change
UI	User Interface
UML	Unified Modeling Language
WAM	Weighted Arithmetic Mean
XP	Extreme Programming

List of Tables

5.1. Exemplary CSV File	69
5.2. Standard Parameters for Analysis	85
5.3. Best Parameters for ROC Forecast	98
5.4. Best Parameters for Sales Forecast	98
6.1. Evaluation of GNOSIS -Construction	111
6.2. Evaluation of GNOSIS - Calibration	112
6.3. Evaluation of GNOSIS - Validation	113
6.4. Evaluation of GNOSIS - Usability Catastrophe	114
6.5. Evaluation of GNOSIS - Major Usability Problems	115
6.6. Evaluation of GNOSIS - Help	116
B.1. Packages for Server	138
B.2. Packages for UI	138
C.1. Used Macroeconomic Indicators for the Sales Forecast	139
D.1. Best Parameters for Averaged Sales Forecast Deviation	143
E.1. Evaluation of GNOSIS - Minor Usability Problems	144
E.2. Evaluation of GNOSIS - Cosmetic Problems Only	144

List of Figures

2.1. Lagged Indicators (Sagaert et al. (2018))	24
4.1. Design Science Research Methodology (Geerts (2011))	50
4.2. Unified Modelling Language	53
4.3. Use Case	55
5.1. GNOSIS Data Acquisition	70
5.2. GNOSIS Data Acquisition - ROC	71
5.3. GNOSIS ROC Statistics - Sales	72
5.4. GNOSIS ROC Statistics - Logistic Regression	73
5.5. GNOSIS Data Normalisation	74
5.6. GNOSIS Indicator Correlations - Table	75
5.7. GNOSIS Indicator Correlations - Correlation Plot	76
5.8. GNOSIS Indicator Correlations - Cross Correlation	76
5.9. GNOSIS Composite Index Construction - Table	77
5.10. GNOSIS Composite Index Construction - Plot	78
5.11. GNOSIS ROC Forecasts - Plot	79
5.12. GNOSIS ROC Forecasts - Logistic Prediction	80
5.13. GNOSIS ROC Forecasts - Accuracy Analysis	80
5.14. GNOSIS Sales Forecasts - Plot	81
5.15. GNOSIS Sales Forecasts - Averaged	82
5.16. GNOSIS Sales Forecasts - Accuracy Analysis	82
5.17. GNOSIS Report Download	83
5.18. ROC Forecast with Varied In-sample Period	86
5.19. Sales Forecast with Varied In-sample Period	87
5.20. ROC Forecast with Varied Forecast Horizon	88
5.21. Sales Forecast with Varied Forecast Horizon	89
5.22. ROC Forecast with Varied Normalisation Method	90

5.23. Sales Forecast with Varied Normalisation Method	91
5.24. ROC Forecast with Varied Correlation Method	92
5.25. Sales Forecast with Varied Correlation Method	93
5.26. ROC Forecast with Varied Cross-Correlation	94
5.27. Sales Forecast with Varied Cross-Correlation	95
5.28. ROC Forecast with Varied Aggregation Method	96
5.29. Sales Forecast with Varied Aggregation Method	97
6.1. GNOSIS Structure - Data Acquisition	102
6.2. GNOSIS Structure - ROC Statistics	103
6.3. GNOSIS Structure - Data Normalisation	104
6.4. GNOSIS Structure - Indicator Correlations	105
6.5. GNOSIS Structure - Composite Index Construction	106
6.6. GNOSIS Structure - ROC Forecasts	107
6.7. GNOSIS Structure - Sales Forecasts	108
6.8. GNOSIS Structure - Report	109
D.1. GNOSIS In-sample Period	140
D.2. GNOSIS Forecast Horizon	141
D.3. GNOSIS Normalisation Method	141
D.4. GNOSIS Correlation Method	142
D.5. GNOSIS Cross Correlation	142
D.6. GNOSIS Aggregation Method	143

Listings

4.1. General Structure of a Function in R	60
4.2. Calling a Function in R	60
4.3. Example for Functions from GNOSIS	61
4.4. General Structure of a Shiny UI	63
4.5. Example of UI Code from GNOSIS	63
4.6. General Structure of a Shiny Server	66
4.7. Example of Server Code from GNOSIS	66

References

- Ali, Ö. G., Sayın, S., van Woensel, T., & Fransoo, J. (2009). SKU demand forecasting in the presence of promotions. *Expert Systems with Applications*, *36*(10), 12340–12348. doi:10.1016/j.eswa.2009.04.052
- Ambler, S. W. (2004). *The object primer*. Cambridge University Pr.
- Andreev, A., Kanto, A., & Malo, P. (2007). Computational Examples of a New Method for Distribution Selection in the Pearson System. *Journal of Applied Statistics*, *34*(4), 487–506. doi:10.1080/02664760701231922
- Aviv, Y. (2001). The Effect of Collaborative Forecasting on Supply Chain Performance. *Management Science*, *47*(10), 1326–1343. doi:10.1287/mnsc.47.10.1326.10260
- Awad, M. A. (2005). A Comparison between Agile and Traditional Software Development Methodologies.
- Becker, W., Paruolo, P., Saisana, M., & Saltelli, A. (2015). Weights and Importance in Composite Indicators: Mind the Gap. In *Handbook of uncertainty quantification* (pp. 1–30). doi:10.1007/978-3-319-11259-6_40-1
- Boivin, J., & Ng, S. (2006). Are more data always better for factor analysis? *Journal of Econometrics*, *132*(1), 169–194. doi:10.1016/j.jeconom.2005.01.027
- Bulligan, G., Marcellino, M., & Venditti, F. (2015). Forecasting economic activity with targeted predictors. *International Journal of Forecasting*, *31*(1), 188–206. doi:10.1016/j.ijforecast.2014.03.004
- Burnham, K. P., & Anderson, D. R. (2004). Multimodel Inference. *Sociological Methods & Research*, *33*(2), 261–304. doi:10.1177/0049124104268644
- Chang, W. (2019). Shiny v1.3.2. Accessed: 2019-03-29. Retrieved from <https://www.rdocumentation.org/packages/shiny/versions/1.3.2>
- Chen, H. L. (2008). Using Financial and Macroeconomic Indicators to Forecast Sales of Large Development and Construction Firms. *The Journal of Real Estate Finance and Economics*, *40*(3), 310–331. doi:10.1007/s11146-008-9158-7

- Currie, C. S. M., & Rowley, I. T. (2010). Consumer behaviour and sales forecast accuracy: What's going on and how should revenue managers respond? *Journal of Revenue and Pricing Management*, *9*(4), 374–376. doi:10.1057/rpm.2010.22
- Dalrymple, D. J. (1987). Sales forecasting practices. *International Journal of Forecasting*, *3*(3-4), 379–391. doi:10.1016/0169-2070(87)90031-8
- Dekimpe, M. G., & Deleersnyder, B. (2017). Business cycle research in marketing: A review and research agenda. *Journal of the Academy of Marketing Science*, *46*(1), 31–58. doi:10.1007/s11747-017-0542-9
- Deleersnyder, B., Dekimpe, M. G., Sarvary, M., & Parker, P. M. (2004). Weathering Tight Economic Times: The Sales Evolution of Consumer Durables Over the Business Cycle. *Quantitative Marketing and Economics*, *2*(4), 347–383. doi:10.1007/s11129-004-0137-x
- Deschamps, B., & Ioannidis, C. (2014). The efficiency of multivariate macroeconomic forecasts. *The Manchester School*, *82*(5), 509–523. doi:10.1111/manc.12016
- Dumas, M., & ter Hofstede, A. H. M. (2001). UML Activity Diagrams as a Workflow Specification Language. In *UML 2001: The unified modeling language. modeling languages, concepts, and tools* (pp. 76–90). doi:10.1007/3-540-45441-1_7
- Elliott, G., Gargano, A., & Timmermann, A. (2013). Complete subset regressions. *Journal of Econometrics*, *177*(2), 357–373. doi:10.1016/j.jeconom.2013.04.017
- Fama, E. F., & French, K. R. (2003). The Capital Asset Pricing Model: Theory and Evidence. *SSRN Electronic Journal*. doi:10.2139/ssrn.440920
- Fildes, R., Goodwin, P., Lawrence, M., & Nikolopoulos, K. (2009). Effective forecasting and judgmental adjustments: An empirical evaluation and strategies for improvement in supply-chain planning. *International Journal of Forecasting*, *25*(1), 3–23. doi:10.1016/j.ijforecast.2008.11.010
- Forni, M., & Reichlin, L. (1998). Let's Get Real: A Factor Analytical Approach to Disaggregated Business Cycle Dynamics. *Review of Economic Studies*, *65*(3), 453–473.
- Franses, P. H., McAleer, M., & Legerstee, R. (2012). Evaluating Macroeconomic Forecasts: A Concise Review Of Some Recent Developments. *Journal of Economic Surveys*, *28*(2), 195–208. doi:10.1111/joes.12000
- Gao, J., Xie, Y., Cui, X., Yu, H., & Gu, F. (2018). Chinese automobile sales forecasting using economic indicators and typical domestic brand automobile sales data: A method based on econometric model. *Advances in Mechanical Engineering*, *10*(2), 168781401774932. doi:10.1177/1687814017749325

- Geerts, G. L. (2011). A design science research methodology and its application to accounting information systems research. *International Journal of Accounting Information Systems*, 12(2), 142–151. doi:10.1016/j.accinf.2011.02.004
- Gumussoy, C. A. (2016). Usability guideline for banking software design. *Computers in Human Behavior*, 62, 277–285. doi:10.1016/j.chb.2016.04.001
- Guo, Z., Wong, W., & Li, M. (2013). A multivariate intelligent decision-making model for retail sales forecasting. *Decision Support Systems*, 55(1), 247–255. doi:10.1016/j.dss.2013.01.026
- Halling, M., Yu, J., & Zechner, J. (2016). Leverage dynamics over the business cycle. *Journal of Financial Economics*, 122(1), 21–41. doi:10.1016/j.jfineco.2016.07.001
- Homareau, J. (2015). Forecasting sales volumes with economic indicators. *Journal of Business Forecasting*, 34(3).
- Hunschofsky, H., Woschank, M., & Magnet, C. (2012). Verbesserung der Unternehmensplanung durch makroökonomische Indikatoren. *WINGbusiness*, 3.
- Hurvich, C., & Tsai, C. L. (1991). Bias of the corrected AIC criterion for underfitted regression and time series models. *Biometrika*, 78(3), 499–509. doi:10.1093/biomet/78.3.499
- Iso 19505-2. (2012). International Organization for Standardization.
- Iso 9241-11. (1993). International Organization for Standardization.
- Jiang, H., Habib, A., & Gong, R. (2015). Business Cycle and Management Earnings Forecasts. *Abacus*, 51(2), 279–310. doi:10.1111/abac.12047
- Klassen, R. D., & Flores, B. E. (2001). Forecasting practices of Canadian firms: Survey results and comparisons. *International Journal of Production Economics*, 70(2), 163–174. doi:10.1016/s0925-5273(00)00063-3
- Klimek, R., & Szwed, P. (2010). Formal analysis of use case diagrams. *Computer Science*, 11, 115–131. doi:10.7494/csci.2010.11.0.115
- Kufenko, V., & Geiger, N. (2016). Business cycles in the economy and in economics: An econometric analysis. *Scientometrics*, 107(1), 43–69. doi:10.1007/s11192-016-1866-9
- Lamont, O. A. (2002). Macroeconomic forecasts and microeconomic forecasters. *Journal of Economic Behavior & Organization*, 48(3), 265–280. doi:10.1016/s0167-2681(01)00219-0
- Lan, C.-h., Lan, T.-s., & Yeh, L.-j. (2001). Forecast cost budget of material ordering and inventory through the price change rate. *Journal of Statistics and Management Systems*, 4(2), 147–160. doi:10.1080/09720510.2001.10701034

- Lawrence, M., O'Connor, M., & Edmundson, B. (2000). A field study of sales forecasting accuracy and processes. *European Journal of Operational Research*, *122*(1), 151–160. doi:10.1016/s0377-2217(99)00085-5
- Leitner, J., & Leopold-Wildburger, U. (2011). Experiments on forecasting behavior with several sources of information – A review of the literature. *European Journal of Operational Research*, *213*(3), 459–469. doi:10.1016/j.ejor.2011.01.006
- Li, J., & Chen, W. (2014). Forecasting macroeconomic time series: LASSO-based approaches and their forecast combinations with dynamic factor models. *International Journal of Forecasting*, *30*(4), 996–1015. doi:10.1016/j.ijforecast.2014.03.016
- Lopez-Perez, V. (2016). Macroeconomic forecast uncertainty in the euro area. *Equilibrium*, *11*(1), 9. doi:10.12775/equil.2016.001
- Malone, S., Gramacy, R., & ter Horst, E. (2016). Timing Foreign Exchange Markets. *Econometrics*, *4*(4), 15. doi:10.3390/econometrics4010015
- Maranon, M., & Kumral, M. (2019). Kondratiev long cycles in metal commodity prices. *Resources Policy*, *61*, 21–28. doi:10.1016/j.resourpol.2019.01.008
- Marcus, A. (2005). 2 chapter - User Interface Design's Return on Investment: Examples and Statistics. In R. G. Bias & D. J. Mayhew (Eds.), *Cost-justifying usability (second edition)* (Second Edition, pp. 17–39). Interactive Technologies. doi:10.1016/B978-012095811-5/50002-X
- Mills, F. C. (1926). An Hypothesis concerning the Duration of Business Cycles. *Journal of the American Statistical Association*, *21*(156), 447–457. doi:10.1080/01621459.1926.10502200
- Mol, C. D., Giannone, D., & Reichlin, L. (2008). Forecasting using a large number of predictors: Is Bayesian shrinkage a valid alternative to principal components? *Journal of Econometrics*, *146*(2), 318–328. doi:10.1016/j.jeconom.2008.08.011
- Navarro, P., Bromiley, P., & Sottile, P. (2010). Business cycle management and firm performance. *Journal of Strategy and Management*, *3*(1), 50–71. doi:10.1108/17554251011019413
- Object Management Group. (2017). *Omg unified modeling language (omg uml)*. 2.5.1. Accessed: 2019-03-20. Object Management Group. 09 Highland Ave Needham, MA 02494 USA. Retrieved from <https://www.omg.org/spec/UML/>
- Pareto, A. (2013). Methods for Constructing Composite Indices: One for All or All for One? *Rivista italiana di economia, demografia e statistica*, *67*, 67–80.

- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. doi:10.2753/mis0742-1222240302
- Petropoulos, F., Fildes, R., & Goodwin, P. (2016). Do ‘big losses’ in judgmental adjustments to statistical forecasts affect experts’ behaviour? *European Journal of Operational Research*, *249*(3), 842–852. doi:10.1016/j.ejor.2015.06.002
- Rötheli, T. F. (2017). Should business rely on business cycle forecasting? *Central European Journal of Operations Research*, *26*(1), 121–133. doi:10.1007/s10100-017-0477-8
- RStudio. (2019). Lesson 2 build a user interface. Accessed: 2019-03-29. Retrieved from <https://shiny.rstudio.com/tutorial/written-tutorial/lesson2/>
- Sagaert, Y. R., Aghezzaf, E.-H., Kourentzes, N., & Desmet, B. (2018). Tactical sales forecasting using a very large set of macroeconomic indicators. *European Journal of Operational Research*, *264*(2), 558–569. doi:10.1016/j.ejor.2017.06.054
- Sanders, N. R., & Manrodt, K. B. (2003). The efficacy of using judgmental versus quantitative forecasting methods in practice. *Omega*, *31*(6), 511–522. doi:10.1016/j.omega.2003.08.007
- Schreiber, S., & Soldatenkova, N. (2016). Anticipating business-cycle turning points in real time using density forecasts from a VAR. *Journal of Macroeconomics*, *47*, 166–187. doi:10.1016/j.jmacro.2015.12.002
- Smith, G. (2018). Step away from stepwise. *Journal of Big Data*, *5*(1). doi:10.1186/s40537-018-0143-6
- Stock, J. H., & Watson, M. W. (2002). Forecasting Using Principal Components From a Large Number of Predictors. *Journal of the American Statistical Association*, *97*(460), 1167–1179. doi:10.1198/016214502388618960
- Stock, J. H., & Watson, M. W. (2012). Generalized Shrinkage Methods for Forecasting Using Many Predictors. *Journal of Business & Economic Statistics*, *30*(4), 481–493. doi:10.1080/07350015.2012.715956
- Sukcharoen, K., & Leatham, D. J. (2016). Dependence and extreme correlation among US industry sectors. *Studies in Economics and Finance*, *33*(1), 26–49. doi:10.1108/sef-01-2015-0021
- Swain, S., Nandi, S., & Patel, P. (2018). Development of an ARIMA Model for Monthly Rainfall Forecasting over Khordha District, Odisha, India. In *Advances in intelligent systems and computing* (pp. 325–331). doi:10.1007/978-981-10-8636-6_34

- Tarabusi, E. C., & Guarini, G. (2010). An aggregation method for composite indicators with unbalance adjustment: An application to the index of african governance. *Working Paper DULBEA*.
- Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1), 267–288.
- Trapero, J. R., Kourentzes, N., & Fildes, R. (2015). On the identification of sales forecasting models in the presence of promotions. *Journal of the Operational Research Society*, 66(2), 299–307. doi:10.1057/jors.2013.174
- Venables, W. N., & Smith, D. M. (2019). *An Introduction to R - Notes on R: A Programming Environment for Data Analysis and Graphics*. 3.6.0. Accessed: 2019-03-10. R-Project. Retrieved from <https://cran.r-project.org/manuals.html>
- Zhang, C., Tian, Y.-X., Fan, Z.-P., Liu, Y., & Fan, L.-W. (2019). Product sales forecasting using macroeconomic indicators and online reviews: A method combining prospect theory and sentiment analysis. *Soft Computing*. doi:10.1007/s00500-018-03742-1

Appendices

A. Deriving Sales Forecast from CI Calculated with $ROC_{12/12}$

The explanation of the derivation of the sales values from the $ROC_{3/12}$ and $ROC_{1/12}$ can be found in chapter 3.2. The derivation for the $ROC_{12/12}$ is analogue to the derivation of the sales values from the $ROC_{3/12}$.

$$ROC_{12/12,t+h,Sales} = \left[\frac{\sum_{i=t+h}^{t-11+h} TS_{i,Sales}}{\sum_{i=(t-12+h)}^{t-23+h} TS_{i,Sales}} - 1 \right] * 100\% \quad (A.1)$$

$$CI_{12/12,(t+h)} = \widehat{ROC}_{12/12,(t+h),Sales} \quad (A.2)$$

$$\widehat{TS}_{t+h,Sales} = \left[\frac{\widehat{ROC}_{12/12,(t+h),Sales}}{100\%} + 1 \right] * \sum_{i=t-12+h}^{t-23+h} TS_{i,Sales} - \sum_{i=t-1+h}^{t-11+h} TS_{i,Sales} \quad (A.3)$$

B. Packages Needed for Server and UI of GNOSIS

The packages used for the server of GNOSIS are shown in table B.1 and the packages used for the UI of GNOSIS are shown in table B.2. The packages used for the demonstration and evaluation of GNOSIS were available at the 10th June 2019.

Table B.1.: Packages for Server

Package	Author	Reasons for Use
shiny	Winston Chang	Generating an interactive web application
xts	Joshua Ulrich	Extending the functionality of time series
ggplot2	Hadley Wickham	Creating more options for visual representation of data
ggcorrplot	Alboukadel Kassambara	Visualisation of correlation matrices
zoo	Achim Zeileis	Extending the functionality of time series
DT	Yihui Xie	Improving the rendering of tables
shinyjs	Dean Attali	Implementing JavaScript code for additional functionality
shinyWidgets	Victor Perrier	Custom input controls and UI components
plotly	Carson Sievert	Making plots interactive
caret	Max Kuhn	Used for regression models
forecast	Rob Hyndman	Calculation of univariate benchmark forecasts
rminer	Paulo Cortez	Calculation of error measurements
writexl	Jeroen Ooms	Exporting data as a Microsoft Excel file
Metrics	Michael Frasco	Evaluation metrics for regression analysis
MASS	Brian Ripley	Model building for logistic regression
dplyr	Hadley Wickham	Used for data manipulation and visualisation
tidyr	Hadley Wickham	Helpful for data manipulation
glmnet	Trevor Hastie	Model building for LASSO

Table B.2.: Packages for UI

Package	Author	Reasons for Use
shinythemes	Winston Chang	Using predefined themes for UI
shinyWidgets	Victor Perrier	Custom input controls and UI components
shinyjs	Dean Attali	Implementing JavaScript code for additional functionality
rhandsontable	Jonathan Owen	Enabling a Microsoft Excel-like table manipulation in the UI

C. List of Indicators for Demonstration

Table C.1 lists the different macroeconomic indicators used for the demonstration of GNOSIS. The URLs were accessed on the 25th May 2019.

Table C.1.: Used Macroeconomic Indicators for the Sales Forecast

Macroeconomic Indicators		
Abbreviation	Name	Description
CEOCI ¹	CEO Confidence Index	Monthly survey, which asks chief executives. the confidence in current and future business conditions and in the revenue and investment forecast for the next year.
CEOEI ²	CEO Economic Outlook Index	A monthly composite index. Consisting of expectations for sales, capital spending and hiring for half a year.
IMNO ³	US Industrial Machinery New Orders	Index containing all new US industrial machinery orders.
MMNO ⁴	US Metalworking Machinery New Orders	Index containing all new US metalworking machinery orders.
MNO ⁵	US Machinery New Orders	Index containing all new US machinery orders.
NARC ⁶	North American Rig Count	Index containing the number of north American rigs.
NDCG ⁷	US Nondefense Capital Goods New Orders w/o Aircraft	Index containing new orders for US nondefense capital goods.
PMI ⁸	Purchasing Manager Index	A survey index that analyse, if purchasing managers think the market conditions are expanding, staying the same, or are contracting.
SP500 ⁹	Standard & Poors: 500	Index containing the 500 most valuable US firms.
US_GPMP ¹⁰	US General Purpose Machinery Production	The Index measures the production of all relevant business located in the US.
USIP ¹¹	US Industrial Production	The Index measures the production of all manufacturing, mining, electric, and gas utilities in the US.
WTICO ¹²	WTI Crude Oil Prices	West Texas Intermediate (WTI or NYMEX) crude oil prices per barrel
WTMES ¹³	US Wholesales Trade of Machinery, Equipment, and Supplies	Index containing US wholesale trades for machinery, equipment, and supplies

¹ <https://www.vistage.com/research-center/vistage-ceo-survey/archive/>

² <https://www.businessroundtable.org/about-us/ceo-economic-outlook-index/ceo-economic-outlook-index-q4-2018>

³ <https://fred.stlouisfed.org/series/U33ENO>

⁴ <https://fred.stlouisfed.org/series/A33INO>

⁵ <https://fred.stlouisfed.org/series/A33SNO>

⁶ <https://bakerhughesrigcount.gcs-web.com/na-rig-count?c=79687&p=irol-reportsotter>

⁷ <https://fred.stlouisfed.org/series/NEWORDER>

⁸ https://ycharts.com/indicators/purchasing_managers_index

⁹ <https://finance.yahoo.com/quote/%5EGSPC/history?p=%5EGSPC>

¹⁰ <https://fred.stlouisfed.org/series/IPG333A9N>

¹¹ <https://fred.stlouisfed.org/series/INDPRO/>

¹² <https://www.macrotrends.net/1369/crude-oil-price-history-chart>

¹³ <https://fred.stlouisfed.org/series/WPU571>

D. Parameter Variation: Deviation of Averaged Sales Forecast

This appendix shows the parameter variation for the deviation of the averaged sales forecast constructed using the $ROC_{1/12}$, $ROC_{3/12}$ or $ROC_{12/12}$.

Length of In-sample Period

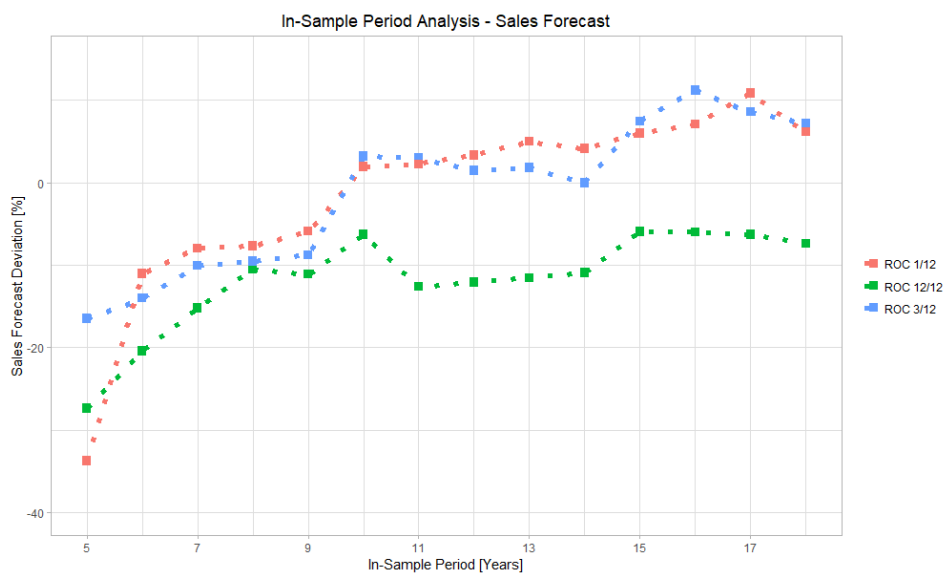


Figure D.1.: GNOSIS In-sample Period

Length of Forecast Horizon

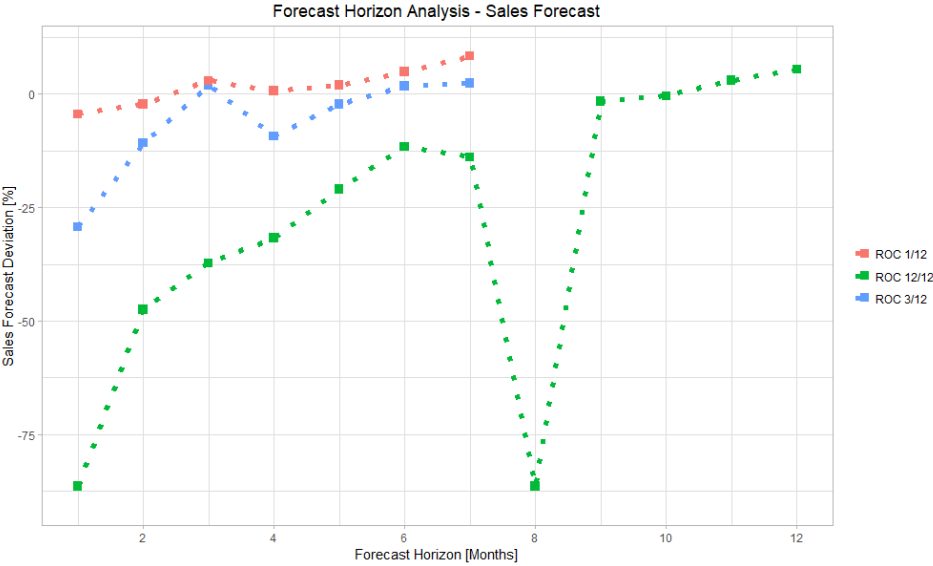


Figure D.2.: GNOSIS Forecast Horizon

Normalisation Method

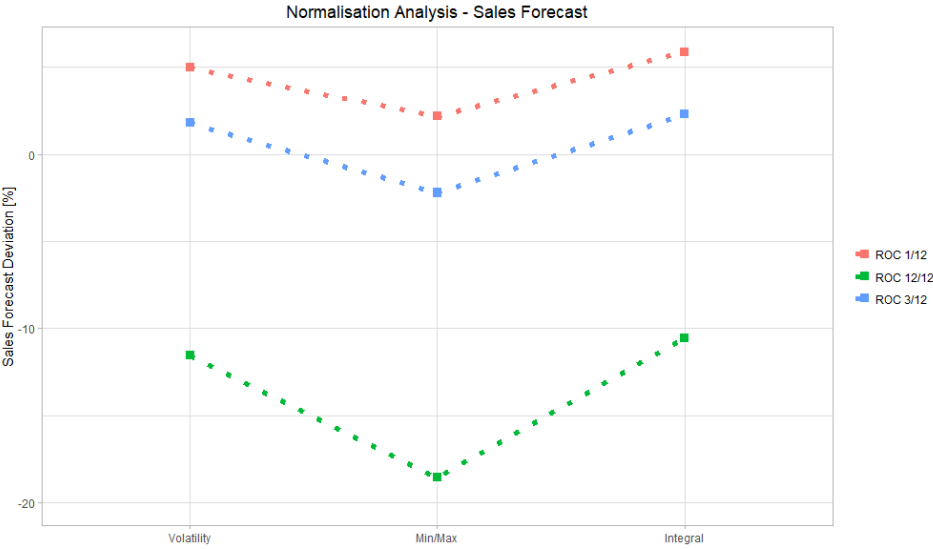


Figure D.3.: GNOSIS Normalisation Method

Correlation Method

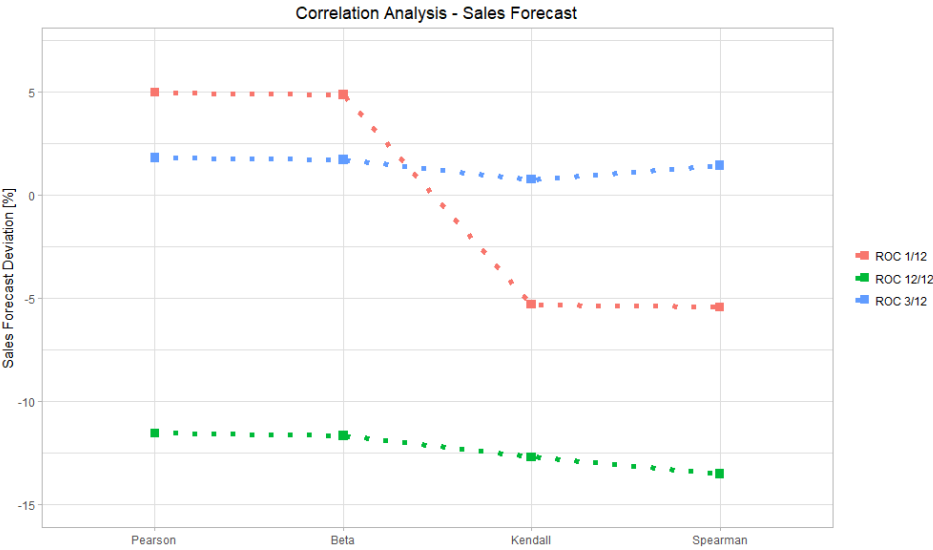


Figure D.4.: GNOSIS Correlation Method

Cross Correlation Value

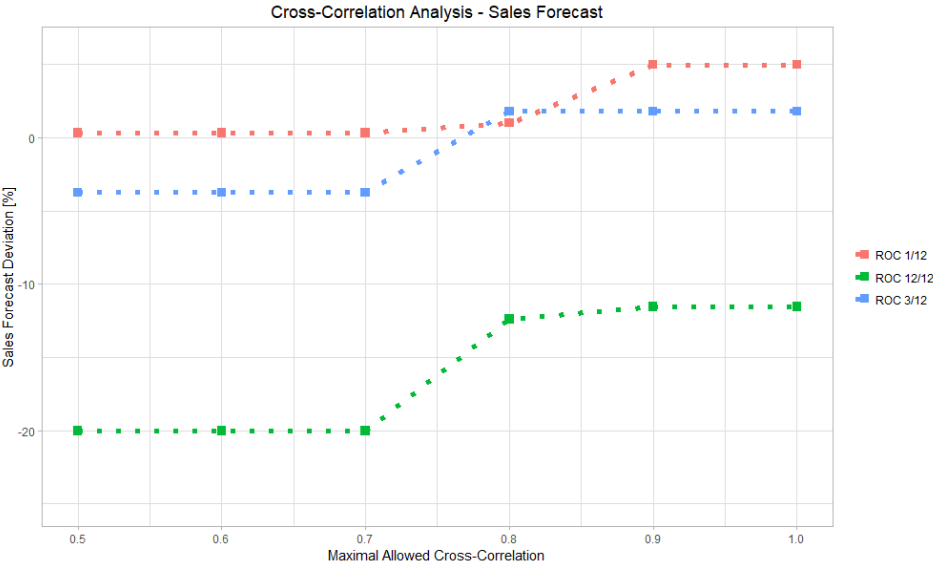


Figure D.5.: GNOSIS Cross Correlation

Aggregation Method

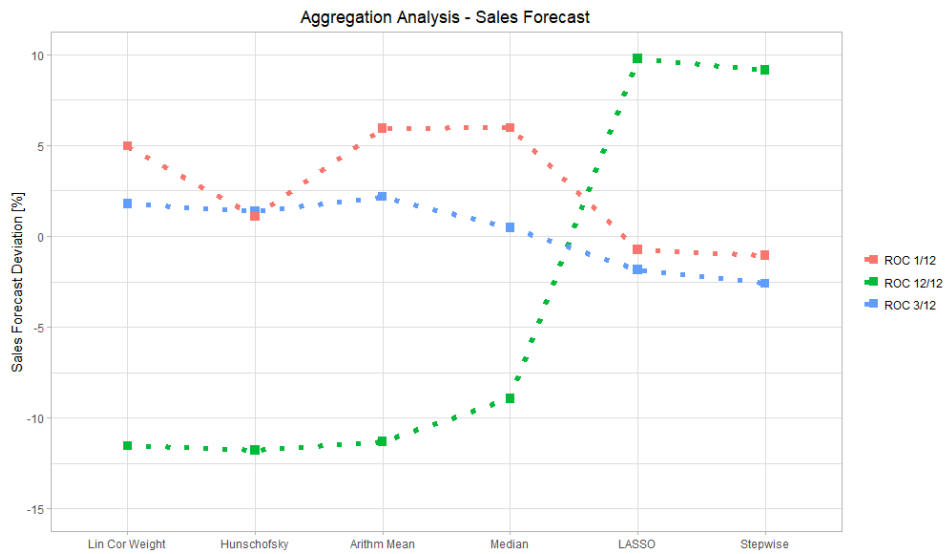


Figure D.6.: GNOSIS Aggregation Method

Summary Parameter Variation

Table D.1.: Best Parameters for Averaged Sales Forecast Deviation

Best Parameters for averaged Sales Forecast Deviation				
Method	Parameter ROC 12/12	Parameter ROC 3/12	Parameter ROC 1/12	
In-Sample Period	2003-01-01	2009-01-01	2008-01-01	
In-Sample Period End	2017-12-01	2017-12-01	2017-12-01	
Forecast Horizon	12	5	10	
Normalisation	Integral	Min/Max	Min/Max	
Mean Normalisation	Yes	Yes	Yes	
Correlation	Pearson	Beta	Spearman	
Cross-Correlation	0.8	1	0.8	
Aggregation	Hunschofsky	Median	LASSO	

E. Evaluation of GNOSIS - Usability: Additional Criteria

The evaluation of the minor usability problems can be seen in table E.1 and the evaluation of the cosmetic problems can be seen in table E.2.

Table E.1.: Evaluation of GNOSIS - Minor Usability Problems

Severity	Usability Criterion	GNOSIS
Minor Usability Problem	Has color been used specifically to draw attention, communicate organization, indicate status changes, and establish relationships?	1
	Has color been used with discretion?	1
	Does each icon stand out from its background?	2
	Have large objects, bold lines, and simple areas been used to distinguish icons?	1
	Is only (and all) information essential to decision making displayed on the screen?	1
	Are all icons in a set visually and conceptually distinct?	2
	Are field labels brief, familiar, and descriptive?	1

Table E.2.: Evaluation of GNOSIS - Cosmetic Problems Only

Severity	Usability Criterion	GNOSIS
Cosmetic Problem only	Have zones been separated by spaces, lines, color, letters, bold titles, rules lines, or shaded areas?	1
	Are borders used to identify meaningful groups?	1
	Are size, boldface, underlying, color, shading, or typography used to show relative quantity or importance of different screen items?	1
	Is color coding consistent throughout the system?	1