A THESIS
submitted in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE
in Business Management
at the
Julius-Maximilians-University of Wuerzburg

DATA AND TEXT MINING IN FINANCIAL MARKETS – CLASSICAL TIME SERIES AND MODERN MACHINE LEARNING TECHNIQUES APPLIED TO FORECAST FOREIGN EXCHANGE RATES

By
Michael Meier

Supervisor:
Professor Dr. Martin Kukuk
Chair for Econometrics
Faculty of Economics
University of Wuerzburg
Submission Date:
24.04.2017

Content

List of Figures

List of Tables

List of Symbols

In general:

t, j, s	time indices	x_{kt}	scalar for observation of feature k at time t
T	total number of time points	$\boldsymbol{x}$	TxI vector containing T observations of one time series
x_t	scalar containing observation at time t	$\boldsymbol{x_t}$	KxI vector containing K observations of K different time series at time t
$\boldsymbol{X}$	TxK matrix containing T observations of K different time series		
$\boldsymbol{x_k}$	TxI vector containing T observations of time series k		

Trend Adjusted ES models (2.1.2.1):

$\widetilde{\boldsymbol{x}}$	vector containing the level of a time series	α	smoothing parameter for the levels
b_t	trend of time series at time t	β	smoothing parameter for the trend

ARIMA models (2.1.2.2):

$\boldsymbol{e}$	TxI vector containing T error terms	μ	mean of time series
α_i	coefficients of AR parts $\forall\, i = 1, \dots, p$	β_i	coefficients of MA parts $\forall\, i = 1, \dots, p$
p	amount of autoregressive terms	q	amount of lagged error terms
d	order of integration	$I(d)$	integrated of order d
δ	slope of linear time trend	h	intercept of linear time trend
c	constant or a linear function	Δ	differencing operator
σ^2	variance	$E[x]$	expected value of x
$F(x)$	joint probability distribution of x	B	backshift operator
$\theta_p(B)$	lag polynomial for p AR components	$\emptyset_q(B)$	lag polynomial for q MA components
$\rho(s)$	autocorrelation for s lags	z	replacement of B in lag polynomials to obtain characteristic equation
λ_i	inverted roots $\forall\, i = 1, \dots, p$		
$\gamma(s)$	autocovariance for s lags	ψ_j	∞ coefficients for MA(∞) process
$\Psi(B)$	infinite lag polynomial with ∞ coefficients ψ_j	$\widetilde{\Psi}_j$	defined as $\widetilde{\Psi}_j = \sum_{i=j+1}^{\infty} \Psi_i$
g_t	stationary part in KPSS test	$\widehat{\omega}_j$	deviations from mean (in KPSS test)
S_t^2	defined as $\sum_{j=1}^{t} \widehat{\omega}_j$	η_j	defined as $\eta_j = 1 - \frac{j}{l_d+1}$ in KPSS test
l_d	defined as $l_d = q(\frac{T}{100})^{1/4}$ in KPSS test		

VAR models (2.1.2.3):

α_{12}	coefficient for feature 1 with respect to feature 2	K	number of time series in VAR
$\boldsymbol{\Gamma_s}$	covariance matrix for s lags		
$\boldsymbol{\Phi_i}$	KxK coefficient matrices for $i = 1, \dots, p$	$\boldsymbol{\Phi(B)}$	matrix lag polynomial for p AR parts
$\boldsymbol{\xi_t}$	Kpx1 vector of stacked vectors $\boldsymbol{x_t}$ to $\boldsymbol{x_{t-p+1}}$	$\boldsymbol{A}$	KpxKp matrix as defined in (40a)
$\boldsymbol{v_t}$	stacked Kpx1 vector containing K errors and (K-1)p zeros	$\boldsymbol{\Psi_i}$	KxK matrix of coefficients regarding the error terms in the Wold form
$\widetilde{\boldsymbol{\Psi}}(\boldsymbol{B})$	infinite matrix lag polynomial with ∞ coefficients $\boldsymbol{\Psi_i}$	$\boldsymbol{\delta}$	KxI vector containing K linear trends

VEC models (2.1.2.4):

$\boldsymbol{w}$	vector containing common trend	λ_i	coefficients of common trends
F	linear combination of time series	β_i	coefficients for linear combination or cointegration relationships
$\boldsymbol{z}$	vector containing several cointegration		
r	number of linear independent cointegration relationships	$\boldsymbol{B}$	matrix containing r cointegration vectors
$\boldsymbol{\Sigma}$	variance-covariance matrix of the error terms	$\boldsymbol{B}^*$	matrix of r normalized cointegration vectors
$\boldsymbol{\Phi}_i$	(k x k) coefficient matrices for $i = 1, \dots, p$		
$\boldsymbol{\Phi}_i$	(k x k) coefficient matrices for $i = 1, \dots, p$	$\boldsymbol{A}$	loading matrix containing the speed of adjustment parameters
$\boldsymbol{\Pi}$	defined as $\boldsymbol{\Pi} = \boldsymbol{A}\boldsymbol{B}'$		
$\boldsymbol{\Lambda}_1$	coefficients of regression of $\boldsymbol{x}_{t-1}$ on lagged differences	$\boldsymbol{\Upsilon}_i$	coefficients of regression of $\Delta\boldsymbol{x}_t$ on lagged differences
$\boldsymbol{u}_{\Delta,t}$	residuals of regression of $\Delta\boldsymbol{x}_t$ on lagged differences	$\boldsymbol{u}_{B,t}$	residuals of regression of $\boldsymbol{x}_{t-1}$ on lagged differences
$\boldsymbol{S}_{ij}$	matrices containing sum of residuals	λ_i	canonical correlations

Naïve Bayes Classifiers (2.2.2.1):

C	class, either zero or one	NA	news announcement
b_j	zero/one indicating absence/presence of word j in term document matrix	w_j	word j in term document matrix
		N_1	total number of NAs of class $C = 1$
$n_1(w_j)$	number of NAs of class $C = 1$ containing word j	m	number of words in term document matrix
N	total number of NAs (both classes)		

Support Vector Machines (2.2.2.2):

β_i	coefficients for determining hyperplane	$\boldsymbol{y}$	vector of responses
M	margin from the hyperplane	C	tuning parameter
ε_i	slack variables	D	distance of support vectors from hyperplane
α_i	Lagrange multipliers		
γ_t	Lagrange multipliers		

Neural Networks (2.2.2.3):

z_j^l	weighted input of neuron j in layer l	C_t	cost function at time t
w_{jk}^l	weight connecting neuron k in layer l-1 with neuron j in layer l	a_t^L	activation level of neuron j in layer l
		δ_j^l	error of neuron j in layer l
$\boldsymbol{B}$	matrix containing all biases	$\boldsymbol{W}$	matrix containing all weight
b_j^l	bias of neuron j in layer l	$\odot$	Hadamard product
∇	gradient operator	η	learning rate

List of Abbreviations

ADF	Augmented Dickey Fuller
AR	Autoregressive
ARMA	Autoregressive Moving Average
ARIMA	Autoregressive Integrated Moving Average
CI	Cointegrated
CSM	Classical Statistical Methods
DT	Deterministic Trend
ES	Exponential Smoothing
GBP	Great Britain Pound
KDD	Knowledge Discovery in Databases
KPSS	Kwiatkowski-Phillips-Schmidt-Shin
lt	linear trend
MA	Moving Average
ML	Machine Learning
MMH	Maximum Margin Hyperplane
NA	News Announcement
NN	Neural Network
nd	no drift
nzc	non-zero constant
PMI	Purchasing Manager Index
SP	Stationary Process
SSE	Sum of Squared Errors
ST	Stochastic Trend
SVM	Support Vector Machine
T	Trend
USD	US Dollar
VAR	Vector Autoregressive
VEC	Vector Error Correction
wd	with drift
zc	zero constant

I. Introduction

Every day governments and businesses are recording and reporting all kinds of information, like opinion polls, online questionnaires, electronic payments, electronic communication, air temperature, atmospheric pressure, blood pressure, GPS locations, videos of surveillance cameras and so forth. Some state, that humans have entered the era of Big Data, where very huge and very interesting data sets are increasingly accessible for a broad mass of people in a very simple way, sometimes through just some clicks on the mouse. We now live in an era in which vast quantities of data can be processed by more and more computational power. Much of this data has the potential to improve our decision making process in very different areas, if only there was a way of systematically making sense of all these data. The futurologist John Naisbitt described the situation as follows : „We are drowning in data but starving for knowledge" (Brown, 2014, p. 1). Because of the huge amount of available data, often some kind of "mining" in the data is necessary. Therefore, this thesis displays some possible techniques, to try to make sense out of and mine in data, whereby data will be common numeric data as well as textual data related to foreign exchange rates (the mining in numerical data is in the following referred to as *data mining*[1], whereby the mining in textual data is referred to as *text mining*[2]). While data and text mining can be seen as broader concepts, the mining itself is executed with *classical time series* as well as with *machine learning* (ML) techniques. To get an idea about these ***two basic types of techniques*** and in order to show their differences, a short introduction for classical time series in part ***A)*** and ML techniques in part ***B)*** is provided in the following.

[1] Data mining (as well as text mining) can be seen as the core part of a bigger process called knowledge discovery in databases, short KDD. KDD thereby consists of ***9 steps***: ***a)*** The goal of the KDD process, as well as the bigger picture of the whole domain and context where the KDD process should take place is identified. ***b)*** A target data set is created by selecting a subset of variables. ***c)*** The target data set is cleaned and preprocessed (for example by removing noise, handling missing values or changing time sequences). ***d)*** The dimension of the cleaned data set is reduced in order to find a more appropriate and compact representation of the data. ***e)*** An appropriate data mining technique is selected, which best fits the goal of the KDD process in step a (for example, a technique for a regression or a classification task is selected). ***f)*** From the chosen technique the appropriate model and the model specification is selected (for example, for a regression task, a multivariate linear regression model is chosen). ***g)*** Here the data mining process is taking place by trying to discover and model reasonable patterns and potential interactions in the data which are statistically valid, previously unknown and potentially useful. ***h)*** The newfound patterns are interpreted and evaluated (possibly, steps a to g are iterated). ***i)*** The discovered knowledge is either used and implemented into another system for further actions, or documented and reported to other interested parties. Seen in this larger context, Fayyad et al. define data mining as "… a step in the KDD process that consists of applying data analysis and discovery algorithms that produce a particular enumeration of patterns (or models) over the data" (Fayyad et al. 1996).

[2] Generally speaking, text mining refers to the task of extracting information from unstructured textual data. Thereby structured data, as opposed to unstructured data, has a high degree of organization, which means that the information is structured by separating it according to features like characteristic attributes or time into different columns and rows. The term "unstructured data" refers to data with a low degree of such an organizational structure. Examples for unstructured textual data are emails, full-text documents, HTML files etc. As textual data are the largest source of information available, automatically extracting knowledge from unstructured textual data has a huge potential for a variety of scientific as well as commercial issues. Some typical issues for text mining are - besides many more - tasks like information extraction (identifying the topic or key phrases of a text), text summarization (retaining the main points and overall meaning of a text), question answering (automatically finding answers to questions in one or more text files) or text classification (the thesis is dealing with this topic in section 2.2.2.1). Thereby, the task of classification in general is to choose the right class label for a given input. A typical example for a text classification problem is to detect whether a given e-mail is spam or not (based on the words and the characteristics of the words in the text) (Weiss et. al, 2010, p. 1-12).

A) Classical time series techniques

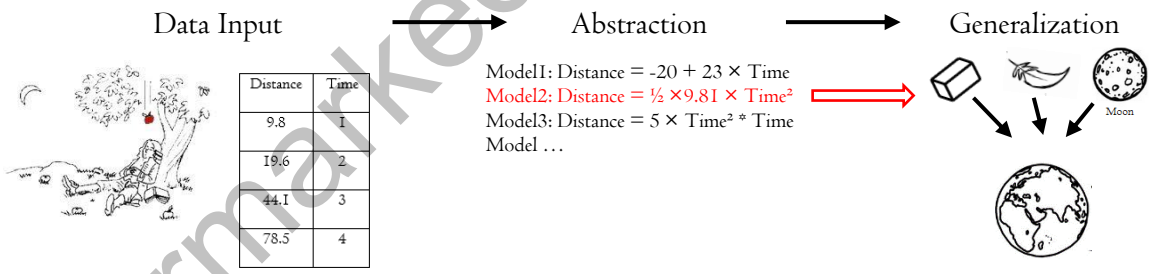
The term "classical" in classical time series techniques is meant to refer to classical statistical methods (in the following referred to as CSMs) as opposed to ML techniques. Both are concerned with the same question: how do we learn from data? Even when the difference is not well defined, it can be argued that ***CSM and ML techniques are different*** when it comes to the following ***three arguments: a)*** CSMs are much older than ML techniques. Modern statistics arose at the beginning of the 19th century when data was sparse, so creating models with strong assumptions could counteract the absence of huge amounts of data. ML in contrast came into existence only in the 1990´s as advances in digitalization and cheap computing power enabled data scientists to use and process bigger amounts of data (Kampakis, 2015, par. 9). ***b)*** CSMs and ML techniques also belong to different scientific disciplines. Where CSMs are a subfield of mathematics, dealing with finding relationships between variables to predict outcomes, ML is a subfield of computer science and artificial intelligence, dealing with the task to build systems and models that can automatically learn from data, instead of explicitly being programmed by instructors (Srivastava 2015). ***c)*** While CSMs work with a number of assumptions and focus more on inference and interpretability, ML methods are in general freed from most of these assumptions and they focus more on the performance of predictions. Thereby, CSMs are based on model building, coefficient estimation and a-priori hypotheses, where the modeler has to understand the assumptions about the relationships between the variables and the statistical properties of the estimators in order to interpret the results and conduct statistical inference on an underlying superpopulation (because they focus more on inference and interpretation, CSMs often prefer low dimensional data sets, simpler models and parsimony). In contrast, ML does not require data scientists to build finished models and instead train computers to do so by algorithms, which automatically process the data in order to find patterns and relationships in them. Thereby, evaluation and generalizability is obtained by measuring the prediction performance on novel, before unseen data sets, whereby usually no superpopulation model is specified (because they focus more on the performance of predictions, ML methods are often applied to high dimensional data sets with less concern about model complexity per se than about overfitting in general) (Shah, 2016, par. 4; Caffo 2016, min 1:06).

B) Machine learning techniques

The idea of making machines learn evolved from the following question: can machines learn from data alone - without explicitly programming them to perform specific tasks? If machines should be able to do this, they would need some form of intelligence – some form of skills, that make them learn by themselves. To understand this better, it can be helpful to shortly outline the necessities and concepts of learning (this will also disenchant the somehow magically sounding words by showing what machines are actually providing and what not when it is said, that they are "learning"). The basic ***learning process***, whether for humans or machines, can be divided into ***three components***: a) data input, b) abstraction and c)

generalization. ***a)*** The learning process needs stored accessible data as a factual basis for further reasoning (as described, the era of Big Data delivers plenty of them). ***b)*** In a second step, the learning process involves the translation of the data into an aggregated and abstracted form, so that the underlying structure of the data is represented in a meaningful way. The new abstracted information is normally in the form of a mathematical model, which tries to describe logical relationships and structured patterns among the data with equations and formulas. The choice of the model is normally not left to the machine. The model is chosen by statisticians who select the model depending on the learning task and the type of data being analyzed. But what the machine then does is automatically fitting a particular model to the dataset (called "training" in the ML context). When the training is completed, the fitted model now describes the original data in a summarized, abstract and hopefully very meaningful form. An inherent problem in the process of abstraction is the question how to model the underlying data in a useful way. There are myriad of underlying relationships that could be identified during the process of abstraction and countless ways to model these relationships. For this problem, the connection of the third step, i.e. the concept of generalization is used. ***c)*** Generalization means to verify if the model also fits good to unforeseen situations and objects. This means that the model not only have to fit the data in the training phase, but should also fit to newly emerging data. With the idea of generalization, the number of abstracted models can be reduced (examining the generalization quality of a model to a previously unseen new data set is referred to as "testing" in the ML context) (Lantz, 2013, p. 10-15). The three components of the learning process can be best illustrated with a small example:

Figure 1: The three components of a learning process



Source: Own illustration

In the first step, observational data are given as input (in the example the distance covered by a falling apple). Then, different models are fitted to the data in order to look for meaningful and abstract relationships and underlying patterns in the data. The last step then tries to find that model, which best generalizes to other situations and objects (for example to the distance covered by other falling objects like bricks or feathers or even to relationships like that between the moon and the earth). A commonly cited formal definition of ML claims that "… a machine is said to learn if it is able to take experience and utilize it such that its performance improves upon similar experiences" (Lantz, 2013, p. 10).

The following thesis deals with these two main types of techniques (classical time series and ML techniques) in theory as well as with empirical examples, in which the theories were applied to model and forecast foreign exchange rates. The structure of the thesis is as follows: the main part 2 deals with theoretical explanations. Part 2.1 displays some basic time series concepts, as well as four commonly used time series techniques. Part 2.2 provides some basic ML concepts, as well as three widely used ML techniques. In part 3, all techniques that have been outlined so far are applied to foreign exchange rates[3]. The thesis concludes with a summary of the main results in part 4.

2. Theoretical approaches

In the following theoretical part 2, classical time series (part 2.1) as well as ML techniques (part 2.2) are outlined. Each subchapter first depicts some general concepts and then describes some of the most commonly used techniques.

2.1. Classical time series techniques

After describing some basic time series concepts in part 2.1.1, four widely used time series techniques are presented in part 2.1.2.

2.1.1. Basic time series concepts

A time series is a sequence measured at (normally) equally spaced discrete points in time. Because of the non-random natural temporal ordering of time series data, observations are not independent from each other so that there can be autocorrelation, trends or seasonalities (the time series is said to have an internal structure). In this case, the law of large numbers as well as the central limit theorem do not apply anymore. Therefore, time series techniques try to account for possible internal structures in the data. Time series techniques can be divided into frequency-domain methods and time-domain methods, while the former include spectral and wavelet analysis, the latter deal with auto- and cross-correlation analysis. In this thesis, only time-domain methods are applied.

Time series analysis assumes, like most other analysis methods, that data consist of a systematic pattern and a random error, which makes the pattern difficult to identify. The patterns are normally divided in two classes: trends and seasonalities. While the former is a linear or non-linear pattern, which does not

[3] Nowadays exchange rates gain more and more importance due to the internationalization of modern businesses, the continuing growth in world trade, trends towards economic integration (free trade zones) and the rapid change of more and more interconnected international financial markets (with huge extensions in volume). Thereby, the exchange rate is very often something like the unpredictable stranger in financial calculations, which can bring salvation or disaster to many balance sheets. An exchange rate in general can be defined as the rate at which one currency is converted into another currency. There are a wide variety of factors which can influence exchange rates, such as inflation, interest rates or trade relationships. In the empirical part, the thesis is dealing with the USD/GBP exchange rate. Some basic concepts about the fundamental determinants of exchange rates are provided in Appendix D.

change over time, the pattern of the latter is repeating itself in systematic intervals over time. This thesis only deals with linear trends so that seasonalities are not regarded (because exchange rates should not contain seasonal patterns) (Nagpaul, 2005, p. 1ff).

2.1.2. Applied techniques

In the following, four widely used time series techniques are illustrated, namely Trend Adjusted ES (Exponential Smoothing), ARIMA (Autoregressive Integrated Moving Average), VAR (Vector Autoregressive) and VEC (Vector Error Correction) models.

2.1.2.1. Trend Adjusted ES models

A very plain forecasting method first applies a trend adjusted exponentially smoothing to the data to obtain a smoothed time series, and then uses this smoothed time series to make predictions. To illustrate the Trend Adjusted ES model, first a simple ES method (without trend adjustment) is explained. Such a simple ES method uses the weighted average of all past observations to smooth the time series, whereby the weights are constructed as follows: the oldest (the first) observation obtains the smallest weight and the most recent (the last) observation obtains the biggest weight, while for observations in between, the weights are decreasing exponentially from the last to the first observation. That the weights are exponentially decreasing can be derived by showing how the smoothed time series $\tilde{\boldsymbol{x}}$ (also called the level) is constructed out of the original time series $\boldsymbol{x}$. The level at time t is a weighted average of the current observation x_t and the level at time t-1 calculated as

$$\tilde{x}_t = \alpha x_t + (1-\alpha)\tilde{x}_{t-1}, \qquad (1)$$

where α is the smoothing parameter. If α is large, then more weight is given to the current observation of the original time series, and when α is small, more weight is given to the previous level. Iteratively replacing $\tilde{x}_{t-1}, \tilde{x}_{t-2}, \dots, \tilde{x}_{t-t+1}$ in (1) leads to

$$\tilde{x}_t = \sum_{j=0}^{t-1} \alpha(1-\alpha)^t x_{t-j} + (1-\alpha)^t x_0, \qquad (2)$$

which shows, that the current level is constructed as the weighted sum of all past observations (whereby the weights are exponentially decreasing). For a forecast based on a simple ES, the current level would be the prediction for the next observation, such that $\tilde{x}_{t+1} = \tilde{x}_t$. Now, for a forecast based on a Trend Adjusted ES, the forecast for the next period would be the previous level plus a trend component b_t, such that $\tilde{x}_{t+1} = \tilde{x}_t + b_t$. The trend component is defined as

$$b_t = \beta(\tilde{x}_t - \tilde{x}_{t-1}) + (1-\beta)b_{t-1} \qquad (3)$$

where β is now the smoothing parameter for the trend. This is again a weighted average, where, when β is large, more weight is given to the current trend (which is the change in levels from t-1 to t), and when β is small, more weight is given to the previous trend b_{t-1}. A large or small β does not mean, that the

trend adjustment is large or small, but that the trend is changing fast or slow. Iteratively replacing $b_{t-1}, b_{t-2}, \dots, b_{t-t+1}$, in (3) leads to

$$b_t = \sum_{j=0}^{t-1} \beta(1-\beta)^t \, (\tilde{x}_{t-j} - \tilde{x}_{t-j-1}) + (1-\alpha)^T b_0 \qquad (4)$$

From (2) it can be seen that what is necessary to obtain the levels $\tilde{x}_t$ are all original observations and a starting value x_0. From (4) it can be seen that what is necessary to obtain the trends b_t are all original levels and a starting trend b_0. Therefore, the parameters for the Trend Adjusted ES models are α, β, x_0 and b_0, which can be estimated by minimizing the sum of squared errors (SSE), defined as $SSE = \sum_{t=1}^{T} (\hat{x}_t - x_t)^2$, whereby $\hat{x}_t$ is a one-step-ahead forecast at time t-1 for time t, such that the errors are one-step-ahead within-sample forecast errors. Figure A.1.1 and Figure A.1.2 on page 62 show different time series where Trend Adjusted ES models have been applied (Hyndman, 2012, chap. 7.1 to 7.3).

2.1.2.2. ARIMA models

This section is divided into ***three subparts***. To illustrate the concepts of ARIMA models, some intuitions are provided in part ***A)***. Thereafter part ***B)*** provides some mathematical formulations. In a last part ***C)*** two common unit root tests are briefly presented.

A) Intuitions

This part A is divided into ***five smaller subparts*** dealing with ***1)*** deterministic as well as stochastic trends and stationarity, ***2)*** stationary ARIMAs, ***3)*** difference-stationary ARIMAs, ***4)*** trend-stationary ARIMAs, and ***5)*** a comparison between difference- and trend-stationary ARIMAs.

1) Deterministic trends (DT), stochastic trends (ST) and stationary processes (SP): In time series analysis, stationary and non-stationary time series can be distinguished. A stationary time series is one, whose properties are independent of time. Such a time series will always look similar, no matter at which point in time it is observed (because it has the same moments at every point in time). Time series with time-dependent deterministic parts, like for example trends, are non-stationary, because the mean of such a time series changes with time. In general, not only the mean, but also other properties of the series, like variance, autocorrelation or skewness could be time-dependent – in such cases, the time series would also be non-stationary. In most cases, only the first (the mean) and the second moments (variance and covariance) of a time series are considered. In the case of time-independent first and second moments, the time series is then said to be weak stationary[4] (Vogel, 2015, p.26f). In this paper, the focus lies on first order non-stationarity caused by time-dependent first moments, i.e. trends. To deal with stationarity, non-stationarity and trends, time series can be framed into the so-called Beveridge-Nelson-

[4] An example of a weak stationary process is a so called white noise process. A white noise process $\boldsymbol{e}$ is defined as $E[e_t] = 0, E[(e_t - \mu)(e_t - \mu)] = \sigma^2 \; \forall \; t$ and $E[(e_t - \mu)(e_{t-s} - \mu)] = 0 \; \forall \; t \neq s$. For such a white noise process, the mean is constantly zero and the second moments are independent of time.

Decomposition. Thereafter, a time series $\boldsymbol{x}$ can be decomposed into a trend (T) and a stationary process (SP), whereby the trend can be a deterministic trend (DT) or a stochastic trend (ST), so that

$$x_t = T + SP = DT + ST + SP \quad (5)$$

Non-stationarity, i.e. a ***time-dependent mean*** now can have ***two reasons: a)*** the stochastic process is exploding or ***b)*** a trend T is present. ***a)*** Univariate time series can be modeled as dependent from a constant, its own past values as well as from contemporaneous and past error terms. Thereby, whether a time series is exploding or not only depends on the coefficients of its own past values. The idea can be illustrated with a simple process $x_t = \alpha_1 x_{t-1} + e_t$, where $\boldsymbol{e}$ is white noise. For $x_{t-1} = 0$ and $e_t = 1$, the value of the process in t is $x_t = 1$. If in the next period $e_{t+1} = 0$, then the value of the time series will be $x_{t+1} = \alpha_1$. For $\alpha_1 > 1$, x_{t+1} will be greater than x_t. If $x_{t+1} > x_t$, then also $x_{t+2} > x_{t+1}$. This is true for all subsequent periods – therefore the process is exploding. If p lagged values are included in the process, then the process is exploding, if the sum of the p lagged values is greater than one in absolute value (Mazzoni, 2015, p. 5-20). ***b)*** Trends can be deterministic, stochastic or deterministic and stochastic (in this context *trend-stationary* and *difference-stationary* processes are distinguished). If a trend seems to have some plausible explanation, it is reasonable to try to model this trend in some manner – and because it can be modeled, it is called a deterministic trend. The deterministic trend can be modeled with regression, where the features are functions of time. If the deterministic trend is removed and the subsequent de-trended time series follows a SP, then such a time series is called a *trend-stationary* time series. Therefore, $x_t = DT + SP$ is called a *trend-stationary* time series (Gruber, 2011, p. 60ff). In contrast, a time series with a stochastic trend shows trends which cannot be explained or modeled. Changes, which cannot be explained and which occur in erratic ways, are said to be caused by random errors produced by a white noise process. If such changes have perpetuating[5] effects on the levels of the time series, then a trend can emerge (meaning that the series could stay above or below its mean for a long time). But due to the randomness of the perpetuating changes, such trends are not explainable – thus the name stochastic trends. If such a stochastic trend is present, then the *levels* have to be distinguished from the *changes* of the time series. As described, the *levels* of a time series with a stochastic trend are non-stationary. But, for most time series with stochastic trends, the *changes* of the time series are stationary[6]. Therefore, differencing can make a non-stationary time series stationary. If a time series is not stationary after differencing the time series one time, then the differencing can be repeated several times, until the

[5] With perpetuating changes is meant, that a change caused by an error will solidify in the level of the time series to some extend. If a series is defined as the sum of the past error terms, then the error terms will change the level 1:1, and thus, the change is 100% perpetually.

[6] For example, for the most simple time series with a stochastic trend, namely a random walk, defined as $x_t = x_{t-1} + e_t$, where $\boldsymbol{e}$ is white noise, the changes, defined as $\Delta x_t = x_t - x_{t-1} = e_t$ are by itself white noise, and therefore stationary (around a zero mean).

differenced time series finally gets stationary. In this sense, a time series $\boldsymbol{x}$ is integrated of order d (or short I(d)), if the d^{th} differencing of $\boldsymbol{x}$ is stationary, while the $(d\text{-}1)^{th}$ differencing of $\boldsymbol{x}$ is non-stationary. An order one integrated time series $\boldsymbol{x} \sim I(1)$ is therefore non-stationary in *levels* but stationary in first differences, i.e. in *changes*. If the stochastic trend is removed (the time series is differenced), and the subsequent differenced time series displays a SP, then such a time series is called a *difference-stationary* time series. Thereby it is crucial if the SP is stationary around a zero mean or around a non-zero mean. A non-zero mean in the changes of a time series would provide the levels of the time series with a drift, i.e. a deterministic trend. Therefore, $x_t = ST + SP$ can be called difference-stationary around a zero mean, and $x_t = DT + ST + SP$ can be called difference-stationary around a non-zero mean (Gruber, 2011, p. 63ff).

Hence, depending on the trends, a time series can be decomposed into at most three parts, so that $x_t = DT + ST + SP$, whereby not all parts are necessarily present in time series. While SP is always present in stochastic time series, this is not the case for DT and ST. Therefore, four time series can be reconstructed out of the Beveridge-Nelson-Decomposition, namely: $x_t = SP$, $x_t = ST + SP$, $x_t = DT + SP$ and $x_t = DT + ST + SP$. In the following, some intuitions are given for the four reconstructed time series. The intuitions are provided inside the framework of the widely used ARIMA models. With the ARIMA notation, all four reconstructed time series can be described. ARIMA models combine ARMA models with the concepts of integration, i.e. the differencing of time series. ARMA processes add autoregressive (AR) and moving average (MA) terms together into a single expression, such that an ARMA(p,q) is defined as

$$x_t = c + \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \ldots + \alpha_p x_{t-p} + e_t + \beta_1 e_{t-1} + \beta_2 e_{t-2} + \ldots + \beta_q e_{t-q} \qquad (6)$$

where p and q describe the amount of autoregressive as well as lagged error terms. Thus, for an ARMA(p,q) model, the features for the response $\boldsymbol{x}$ are a deterministic term c, p lagged values of $\boldsymbol{x}$ as well as a contemporary and q lagged errors. An intuitive way of thinking about ARMA models is, that they compute the actual value of $\boldsymbol{x}$ from the weighted sum of the last p values of $\boldsymbol{x}$ plus the weighted sum of the last q errors (plus a deterministic term and a contemporary error). The letter I in ARIMA displays the concept of integration. If a I(d) time series is differenced d times, and subsequently, an ARMA(p, q) model is applied, then this time series is said to follow an ARIMA(p, d, q) process (Vogel, 2015, p. 123f). Combining different values of p, d, q and c leads to a vast amount of possible models. However, in most cases, either p or q is equal to zero, and p plus q is often not greater than three. Also d is for most economic time series not greater than one[7]. The ***interpretation of c*** depends on whether the time series is

[7] In the following only integrated processes of order 1 are considered. I(2) or higher order integrated time series would display quadratic or higher order polynomial trends (which seem not very plausible for many financial time series). Nevertheless, for cases where the order of integration is not certain, the problem of overdifferencing can arise (meaning that an actually I(d)

a) stationary, b) trend-stationary or c) difference-stationary. ***a)*** If a time series is a stationary I(0) process, then the level of the series is stable over time around the time-independent constant c. Such time series are said to be mean stable. The series follows a horizontal line with a stable zero mean if $c = 0$ or a stable non-zero mean if $c \neq 0$[8] (Hyndman, 2012, chap. 8.5). ***b)*** If a time series $\boldsymbol{x}$ is trend-stationary, then c would be time-dependent and displays a deterministic trend, for example $c(t) = h + \delta t$. In this case, the trend can be removed by subtracting the trend from the series to obtain a de-trended time series $\tilde{\boldsymbol{x}}$ such that $\tilde{x}_t = x_t - (h + \delta t)$. When an ARIMA(p,q) is applied to the de-trended time series $\tilde{\boldsymbol{x}}$, then $c = 0$ (because the intercept h was also removed by de-trending the time series)[9]. ***c)*** If the series is a difference-stationary I(1) process, then the level of the series is not stable over time (but the changes are). If an ARIMA(p,q) is applied to the differenced time series, then a constant $c \neq 0$ will give the time series a drift, so that the series is growing for $c > 0$ and declining for $c < 0$[10] (Hyndman, 2012, chap. 8.1). The following examples combine the ARIMA notation with the described suffixes zc/nzc/lt/nd/wd[11] whereby corresponding figures are shown in Appendix A.2.

2) Stationary ARIMAs ($x_t = SP$): For an ARIMA(0,0,0)(zc), equation (6) collapses to a simple stationary white noise process $x_t = e_t$, with erratic changes around zero. For an ARIMA(0,0,0)(nzc) defined as $x_t = c + e_t$, the mean of the white noise process μ is just shifted up or down horizontally. An ARIMA(1,0,0)(zc) corresponds to an AR(1) process $x_t = \alpha x_{t-1} + e_t$. For an AR(1) process to be mean stable, α must be smaller than 1 in absolute value, so that the absolute value of the time series is declining with a rate of $1 - \alpha$ (otherwise the process would be exploding). Because of the autocorrelation in $\boldsymbol{x}$, changes are now persistent[12]. Theoretically their impact is endless but it converges to zero with a speed depending on α. For an ARIMA(1,0,0)(nzc) with a non-zero constant, the AR(1) process would just be shifted up or down horizontally depending on c. The mean can be calculated as $\mu = \frac{c}{1-\alpha}$ and therefore a higher α would provoke a higher mean. Figure A.2.1 on page 63 shows a white noise process

time series is differenced t+1 times). Overdifferencing can complicate the model building process and should therefore be avoided (overdifferencing can be identified relatively easy, because differencing a non-stationary time series reduces the variance while differencing an already stationary time series increases the variance).

[8] This will be denoted thereafter with zc for "zero constant" and nzc for "non-zero constant", such that if an ARIMA model is applied to a I(0) stationary time series, this will be denoted as ARIMA(p,0,q)(nzc) for a non-zero constant and ARIMA(p,0,q)(zc) for a zero constant.

[9] If an ARIMA is applied to a de-trended trend-stationary time series, this will be denoted as ARIMA(p,0,q)(lt) with lt for "linear trend".

[10] A drift will be denoted with nd for "no drift" if $c = 0$ or wd for "with drift" if $c \neq 0$, such that when an ARMA model is applied to an I(1) difference-stationary time series this will be denoted as ARIMA(p,1,q)(nd) in case the time series has no drift and ARIMA(p,1,q)(wd) in case the time series is with a drift.

[11] For an explanation for zc/nzc/lt/nd/wd see footnote 7-10. Stationary time series will have the suffix zc/nzc, trend-stationary time series the suffix lt and difference stationary time series will have the suffix nd/wd in the following.

[12] As opposed to perpetually, meaning that a change has a persistent impact on the levels (but because the levels are reverting back to the mean, changes do not solidify in the levels and are therefore not perpetually).

with $c = 0$, a white noise process with $c = 2$, as well as an AR(1) process with $\alpha = 0.8$ and $c = 0$ and an AR(1) process with $\alpha = -0.3$ and $c = 2.5$ (Stier, 2001, p.44ff).

For an ARIMA(0,0,1)(zc), equation (6) becomes a MA(1) process $x_t = \beta e_{t-1} + e_t$. For a MA(1) process, the impact of a random error will last only one more subsequent period. For such a process, the sign of the impact in the following period will be reversed if $\beta < 0$ and the strength of the impact can be enhanced or reduced depending on whether $|\beta| > 1$ or $|\beta| < 1$. This is illustrated on page 63 in Figure A.2.2 on the left side, where an ARIMA(0,0,1)(zc), with an enhancing effect, as well as an ARIMA(0,0,1)(nzc) with a reducing effect of the error term is shown.

For higher orders of p or q, $\boldsymbol{x}$ depends on more lagged $\boldsymbol{x}$ variables or more lagged error terms $\boldsymbol{e}$. Depending on the values of α_1 to α_p and β_1 to β_q all kinds of different AR and MA models can be constructed, while a stationary process always shows a reversion to the mean after an impact of an error term occurred. In general, an impact of an error for a MA(q) processes will last only q periods, while the impact of an error term for an AR(p) process is theoretically infinite but decaying (independent of the order p but dependent on the coefficients α_1 to α_p) (Mills; Markellos, 2008, p. 22ff). This is illustrated in Figure A.2.2 on the right side, where an ARIMA(2,0,0)(zc) and an ARIMA(0,0,2)(nzc) is shown.

If AR and MA processes are combined together to display ARMA processes, then the levels are constructed out of AR and MA parts. The subsequent changes in the levels after an error term occurred are therefore defined by these AR and MA parts together, whereby the MA part contributes to only q subsequent levels after an error occurred, while the AR contribution is theoretically infinite (like for simple AR and MA processes) (Stier, 2001, p. 57). An example for an ARIMA (1,0,1)(zc) is shown in Figure A.2.3 on page 64.

3) Difference-stationary ARIMAs ($x_t = ST + SP$ and $x_t = DT + ST + SP$): For the most simple difference-stationary process, called a random walk, equation (6) becomes $x_t = x_{t-1} + e_t$. In ARIMA notation this would correspond to ARIMA(0,1,0)(nd). In the case of such a random walk, the time series is differenced once, and to the differenced series, an ARMA(0,0)(zc) is applied. A random walk with no drift is equivalent to an AR(1) process where $\alpha = 1$. Because $\boldsymbol{x}$ is the result of the sum of the past error terms, a random walk follows a stochastic trend and is therefore non-stationary[13]. The first differences of a random walk with no drift are equal to $\Delta x_t = e_t$ (they are white noise and therefore stationary). An ARIMA(0,1,0)(wd) is called a random walk with drift, while now the ARMA(0,0)(nzc), which is applied to the differenced time series, is white noise with a non-zero constant c. A non-zero constant in the differences means that the time series is growing in a deterministic way. A random walk without drift

[13] The non-stationarity results from a time-dependent variance $E[(x_t - \mu)(x_t - \mu)] = t\sigma^2, \forall\, t$, and not from a time-dependent mean (because the mean of a random walk is a constant and therefore time-independent).

would just follow a stochastic trend, but a random walk with drift follows a stochastic as well as a deterministic trend (called a drift) (Gruber, 2011, p. 61f). To illustrate this, an ARIMA(0,1,0)(nd) and an ARIMA(0,1,0)(wd) as well as the corresponding first differences are illustrated in Figure A.2.4 on page 64.

When a difference-stationary I(1) time series is combined with an AR(1)(zc) process, the model would display the process $x_t = x_{t-1} + \alpha(x_{t-1} - x_{t-2}) + e_t$. Written differently, this gives $\Delta x_t = \alpha \Delta x_{t-1} + e_t$, which shows that the differenced time series Δx_t follows an AR(1) process. In this case, the series is non-stationary and follows a stochastic trend, again with a time-dependent variance. For the series to be difference-stationary, the AR properties must now hold for the differences, i.e. α must be smaller than one in absolute value (otherwise the process is exploding). Now, not the levels follow a reversion back to the mean, but the differences are converging to zero, meaning that one change will produce further changes but with decreasing strength. The series itself is not converging, so that all changes are to some extend perpetuated in the series. Similar to a random walk, $\boldsymbol{x}$ equals the sum of all last changes plus a weighted sum of the lagged changes. For $0 < \alpha < 1$, the impact of an error is enhanced, because a change caused by an error will provoke further decaying changes in the same direction (thereby, the series is not oscillating). For $-1 < \alpha < 0$ the impact of an error is reduced in the subsequent periods because a change caused by an error will provoke further decaying changes in the opposite direction (thereby, the series is oscillating). This is illustrated in Figure A.2.5 on page 65, where two different ARIMA(1,1,0)(nd) are shown with their corresponding first differences. Similar to a random walk with drift, a non-zero constant in an ARIMA(1,1,0)(wd) provides the series with a drift, leading to a twist of the original time series. The time series then again follows a deterministic as well as stochastic trend (whereby the changes follow a stationary AR process around a non-zero mean). In general, when differenced time series are combined with AR processes, all AR properties now must hold for the differenced time series. The same is true when a MA(q) process is applied to a differenced time series. Then the MA properties must hold for the differenced time series. This is illustrated for an AR and a MA process in first differences in Figure A.2.6 on page 65, where an ARIMA(1,1,0)(wd) and an ARIMA(0,1,1)(wd) is shown.

If AR and MA processes are jointly applied to a differenced time series, then the differences are compounds of AR and MA parts and therefore changes in differences after an error occurred are compounds of AR and MA parts, while again, the MA part only contributes to q subsequent changes in differences (in general, all ARIMA(p,0,q) properties now must hold for the differenced time series). An example of an ARIMA (1,1,1)(nd) with corresponding first differences, as well as with AR and MA parts, is provided in Figure A.2.7 on page 66.

4) Trend-stationary ARIMAs ($x_t = DT + SP$): Trend-stationary I(0) time series are stationary around a deterministic trend. After removing the deterministic trend, an ARMA(p,q) can be applied to the de-trended time series. The most simple trend-stationary time series would be a white noise process around a linear trend, which corresponds to ARIMA(0,0,0)(lt). The de-trended time series also could display a time series with AR and/or MA components (Gruber, 2011, p. 60f). Two examples of trend-stationary ARIMAs are provided in Figure A.2.8 on page 66, where an ARIMA(0,0,0)(lt), as well as an ARIMA(1,0,0)(lt) is shown.

5) Trend- vs difference stationarity ARIMAs: As described above, differencing can make I(1) time series stationary. But differencing also can make an I(0) trend-stationary time series stationary (in this case, not the removal of the trend, but the differencing removes the non-stationarity)[14]. But the behavior and the forecast of a difference- and a trend-stationary time series vary. If a trend-stationary time series is erroneously differenced, the forecast would contain an error. The error occurs because for a difference-stationary time series changes caused by an error term are to some extend perpetually on the long run (for a random walk to 100%, and for ARIMA(p,1,q) processes the materialization depends on the orders of p and q), whereas for a time series with a linear trend, changes caused by the error term are reversed to 100% and therefore they are not perpetually on the long run. For a difference-stationary time series with drift, the first differences are reverting back to the mean of the first differences, meaning that there is a reversion to the growth-rate (but with a changed shifted trend-line). For a trend-stationary time series, the level of the time series is reverting back to the linear trend, meaning that there is a reversion to the (not shifted) trend-line (Cowpertwait & Metcalfe, 2009, p.138). This is illustrated in Figure A.2.9 on page 67, where an ARIMA(1,1,0)(wd) and an ARIMA(1,0,0)(lt), as well as their forecasts are shown.

B) Mathematical formulations

This part B is again divided into ***five smaller subparts*** dealing with ***1)*** deterministic as well as stochastic trends and stationarity, ***2)*** stationary ARIMAs, ***3)*** difference-stationary ARIMAs, ***4)*** trend-stationary ARIMAs, and ***5)*** a comparison between difference- and trend-stationary ARIMAs.

1) Deterministic trends (DT), stochastic trends (ST) and stationary processes (SP): As described above, stationarity means, that the probabilities of the time series are time-independent. Thereby, with regards to the theoretical moments of a time series, *strict* and *weak* stationary processes can be distinguished. When the joint probability distribution for a stochastic variable $\boldsymbol{x}$ is defined as $F(x_1, x_1, \dots, x_T) = P(X_1 \leq x_1, X_2 \leq x_2, \dots, X_T \leq x_T)$, then a process is 1st order stationary, if $F(x_t) = F(x_{t+s})\ , \forall\ t, s$, a process is 2nd order stationary, if $F(x_t, x_{t+j}) = F(x_{t+s}, x_{t+j+s})\ , \forall\ t, j, s$ and a process is nth order stationary, if $F(x_t \dots x_{t+n}) = F(x_{t+s} \dots x_{t+s+n})\ , \forall\ t, n, s$. For a time series to be *strict* stationary, it

[14] The definition of difference-stationarity prohibits to call differenced trend-stationary time series difference-stationary (regardless from the fact, that they are stationary after differencing).

must be n[th] order stationary, which means, that the joint distribution of a time series with length n is not dependent on time shifts s. Such strict requirements are often not necessary, so that it is sufficient to define stationarity in a weaker form. For a process to be weak stationary, it must be 2[nd] order stationary. The weak stationarity requires

$$E[x_t] = \mu, \forall\, t \tag{7}$$

$$E[(x_t - \mu)(x_{t-s} - \mu)] = \gamma_s, \forall\, t, s \tag{8}$$

with μ as the mean of $\boldsymbol{x}$. This means, that the mean and the variance (for $s = 0$) are not dependent on time and that the covariances (for $s \neq 0$) only depend on the distance s (Pfaff et al., 2008, p. 3ff).

The ARMA models depicted before, defined as

$$x_t = c + \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \ldots + \alpha_p x_{t-p} + e_t + \beta_1 e_{t-1} + \beta_2 e_{t-2} + \ldots + \beta_q e_{t-q} \tag{9}$$

can be written with a backshift operator in a more compact form. The backshift operator B is defined as $B^s x_t = x_{t-s}$[15], and therefore, $x_t - \alpha_1 x_{t-1} - \alpha_2 x_{t-2} + \cdots - \alpha_p x_{t-p}$ can be written as

$$x_t - \alpha_1 B x_t - \alpha_2 B^2 x_t - \ldots - \alpha_p B^p x_t = (1 - \alpha_1 B - \alpha_2 B^2 - \ldots - \alpha_p B^p) x_t = \theta_p(B) x_t \tag{10}$$

and $e_t + \beta_1 e_{t-1} + \beta_2 e_{t-2} + \ldots + \beta_q e_{t-q}$ can be analogously written as

$$(1 + \beta_1 B + \beta_2 B^2 + \ldots + \beta_q B^q) e_t = \emptyset_q(B) e_t \tag{11}$$

Therefore, an ARMA(p,q) can then be expressed as $\theta_p(B) x_t = c + \emptyset_q(B) e_t$[16]. When applying an ARIMA model, the time series is first differenced d times and then an ARMA model is applied. When the d times differenced time series is denoted as $\Delta^d x_t$, whereby Δ displays the differencing operator, then replacing x_t with the differenced time series $\Delta^d x_t$ leads to $\theta_p(B) \Delta^d x_t = c + \emptyset_q(B) e_t$. The differenced time series can also be expressed with the backshift operator as $\Delta^d x_t = (1 - B)^d x_t$, which leads to

$$\theta_p(B)(1 - B)^d x_t = c + \emptyset_q(B) e_t \tag{12}$$

Setting a backshift polynomial to zero and treating the backshift operator B as a number by replacing it with z leads to the so called characteristic equation. The characteristic equation for the backshift

[15] Applying the operator one time results in the value of the time series, which is lagged by one period ($B x_t = x_{t-1}$). Applying the operator on an already lagged time series $B x_t$, results in the value of the time series, which is lagged by two periods ($B^2 x_t = BB x_t = B x_{t-1} = x_{t-2}$). Additionally, $B^0 = 1$, such that $B^0 x_t = x_t$. Applying the operator to a constant c does not change the value of the constant, such that $B^s c = c\ \forall\, s$.

[16] A backshift polynomial of order p can in general be displayed with the backshift operator as $P(B) = p_0 + p_1 B + p_2 B^2 + \ldots + p_p B^p$. All calculation rules which hold for polynomials also hold for backshift polynomials. A backshift polynomial applied to an observation of a time series gives $P(B) x_t = (p_0 + p_1 B + p_2 B^2 + \cdots + p_p B^p) x_t = p_0 x_t + p_1 B x_t + x_t p_2 B^2 + \cdots + x_t p_p B^p = p_0 x_t + p_1 x_{t-1} + p_2 x_{t-2} + \cdots + p_p x_{t-p}$. Often, the backshift polynomial is normalized with $p_0 = 1$. Moreover, for $B = 1$, the backshift polynomial results in the sum of the p coefficients $P(1) = \sum_{i=1}^{p} p_i$. The backshift polynomial applied to a constant results in the sum of the coefficients multiplied by the constant $P(B)c = (p_0 + p_1 B + p_2 B^2 + \cdots + p_p B^p)c = \sum_{i=1}^{p} p_i\, c = P(1)c$. The z-transformed $P(z)$ to a backshift polynomial $P(B)$ is a polynomial in a (complex) variable z with the same coefficients p_1 to p_k like the backshift polynomial $P(B)$, such that $P(z) = p_0 + p_1 z + p_2 z^2 + \cdots + p_p z^p$.

polynomial of the error terms, for example, becomes $\emptyset_q(z) = 0$ (Mills & Markellos, 2008, p. 14ff). In the following, it will be shown how a time series $\boldsymbol{x}$ can be decomposed into a trend (T) and a stationary process (SP), whereby T can be a deterministic trend (DT) or a stochastic trend (ST), so that $x_t = T + SP = DT + ST + SP$. While a SP is always a part of a stochastic time series, DT and ST can be present or not, leading to the possible combinations: $x_t = SP$, $x_t = ST + SP$, $x_t = DT + SP$ and $x_t = DT + ST + SP$. The four possible time series will be depicted in the following.

2) Stationary ARIMAs ($x_t = SP$): For an ARIMA(0,0,q) or MA(q) processes, equation (12) becomes

$$x_t = c + e_t + \beta_1 e_{t-1} + \beta_2 e_{t-2} + \ldots + \beta_q e_{t-q} = c + \emptyset_q(B) e_t \tag{13}$$

As a finite sum of stationary white noise terms, a MA(q) process has a time-invariant mean, variance as well as auto-covariance and is therefore stationary. Because all errors have a mean of zero, the mean of $\boldsymbol{x}$ is just the constant c. The variance is $Var[x_t] = \sigma_e^2(1 + \beta_1 + \ldots + \beta_q)$ because each of the error terms has the same variance and the error terms are mutually independent[17]. The autocorrelation function[18], for $s \geq 0$, is given by

$$\rho(s) = \begin{cases} 1 & s = 0 \\ \sum_{i=0}^{q-s} \beta_{i+s}\beta_i / \sum_{i=0}^{q} \beta_i^2 & s = 1, \ldots, q \\ 0 & s > q \end{cases} \tag{14}$$

with $\beta_0 = 1$. The autocorrelation function is zero when $s > q$, because x_t and x_{t+s} then consist of sums of independent white noise terms with covariance zero. For a MA(q) process to be unique, the process must be invertible[19]. When the roots of the characteristic equation of the error terms all exceed unity in absolute value, so that $\emptyset_q(z) \neq 0$ for $|z| \leq 1$, then the process is called invertible (Cowpertwait & Metcalfe, 2009, p. 121ff).

For an ARIMA(p,0,0) or AR(p) processes, (12) becomes

$$\theta_p(B) x_t = (1 - \alpha_1 B - \alpha_2 B^2 - \ldots - \alpha_p B^p) x_t = c + e_t \tag{15}$$

Therefore, an AR(p) process only depends on its own lagged values (sometimes referred to as the inner dependence of the process). In order to see, whether an AR(p) process is stationary or not, it is necessary to look at the characteristic equation of the backshift polynomial of the AR parts

---

[17] $Var(x_t) = E[x_t x_t] - E[x_t]^2 = E\left[(c + e_t + \beta_1 e_{t-1} + \beta_2 e_{t-2} + \ldots + \beta_q e_{t-q})(c + e_t + \beta_1 e_{t-1} + \beta_2 e_{t-2} + \ldots + \beta_q e_{t-q}) - (c + e_t + \beta_1 e_{t-1} + \beta_2 e_{t-2} + \ldots + \beta_q e_{t-q})^2\right] = E[c^2 + c e_t + c\beta_1 e_{t-1} + \ldots + c\beta_q e_{t-q} + e_t c + e_t e_t + e_t \beta_1 e_{t-1} + \ldots + e_t \beta_q e_{t-q} + \ldots + \beta_q e_{t-q} c + \beta_q e_{t-q} \beta_1 e_{t-1} + \ldots + \beta_q e_{t-q} \beta_q e_{t-q}] - c^2 = c^2 + \sigma_e^2 + \beta_1^2 \sigma_e^2 + \ldots + \beta_q^2 \sigma_e^2 - c^2 = \sigma_e^2 (1 + \beta_1 + \ldots + \beta_q)$.

[18] $Cov(x_t, x_{t-1}) = E[x_t x_{t-1}] - E[x_t] E[x_{t-1}] = E[(c + e_t + \beta_1 e_{t-1} + \beta_2 e_{t-2} + \ldots + \beta_q e_{t-q})(c + e_{t-1} + \beta_1 e_{t-2} + \beta_2 e_{t-3} + \ldots + \beta_q e_{t-q-1})] - c^2 = E[c^2 + c e_{t-1} + c\beta_1 e_{t-2} + \ldots + c\beta_q e_{t-q-1} + e_t c + e_t e_{t-1} + e_t \beta_1 e_{t-2} + \ldots + e_t \beta_q e_{t-q-1} + \ldots + \beta_q e_{t-q} c + \beta_q e_{t-q} e_{t-1} + \ldots + \beta_q e_{t-q} \beta_q e_{t-q-1}] - c^2 = c^2 + \beta_1 \sigma_e^2 + \beta_2 \beta_1 \sigma_e^2 + \ldots + \beta_q \beta_{q-1} \sigma_e^2 - c^2 = \sigma_e^2 \sum_{i=0}^{q-1} \beta_{i+1} \beta_i$, with $\beta_0 = 1$.

[19] Invertible means, that a MA process can be expressed as a stationary AR process of infinite order. For example $x_t = e_t + \beta e_{t-1} = (1 - \beta B) e_t$ can be expressed as $e_t = (1 - \beta B)^{-1} x_t = x_t + \beta x_t + \beta^2 x_t + \ldots$.

$$\theta_p(z) = 0 \qquad (16)$$

For a polynomial of degree p there are p roots. The roots of the characteristic equation are dependent on the coefficients α_1 to α_p. If the coefficients are in that way, that a (complex) root in absolute values in the characteristic equation lies inside the unit circle, then the process is a non-stationary explosive process[20]. If all absolute values of the (complex) roots in (16) are outside the unit circle, then the process is stationary[21]. If the coefficients α_1 to α_p are in this way, that one (complex) root in absolute values in (16) lies exactly on the unit circle, then the process is non-stationary and contains a so called unit root[22]. Changing sides of the backshift polynomial in (15) results in a MA(∞) process $x_t = \theta_p(\boldsymbol{B})^{-1}c + \theta_p(\boldsymbol{B})^{-1}e_t$. For an AR(1) process without constant this would be equal to $x_t = (1+\alpha B)^{-1}e_t = (1+\alpha B+\alpha^2B^2+\alpha^3B^3+\ldots)e_t = e_t+\alpha e_t+\alpha^2e_{t-2}+\alpha^3e_{t-3}+\ldots = \sum_{j=0}^{\infty}\alpha^j e_{t-j}$. This means that an AR(p) process can be transformed into an MA(∞) process, if the AR process is stationary. As MA(∞) processes are the sum of infinite past error terms, the mean of an AR(1) process is zero, i.e. $E[x_t] = 0$ (Cowpertwait & Metcalfe, 2009, p. 79ff). The variance of an AR(1) process is $Var[x_t] = Var[e_t+\alpha e_t+\alpha^2e_{t-2}+\alpha^3e_{t-3}+\ldots] = (1+\alpha+\alpha^2+\alpha^3+\ldots)^2\sigma_e^2 = \sigma_e^2/(1-\alpha^2)$ and the auto-covariance function (called Yule-Walker equation)[23] is

$$\gamma(s) = \alpha\gamma(s-1) \qquad \text{for } s > 0 \qquad (17)$$

For an ARIMA(p,0,q) process, equation (12) becomes $\theta_p(B)x_t = c + \emptyset_q(B)e_t$. In order to be a stationary ARIMA(p,0,q) process only the AR part matters, and therefore the condition $\theta_p(z) \neq 0$ for $|z| \leq 1$ needs to hold, which means that there must not be a solution for $\theta_p(z)$, which is smaller than one (for a solution $|z| = 1$, the process would be integrated). The ARIMA(p,0,q) process is invertible

[20] A process where a (complex) null in absolute values in (16) lies inside the unit circle, can be stationary, but then it would not be causal. Not causal means, that x_t could be explained by his own values in the future x_{t+k} and/or by future shocks e_{t+k}, where $k > 1$. This solutions are not reasonable for time series predictions and are thus excluded.

[21] If an AR(p) process is stationary, then it has the same constant expected value $E[x_t] = \mu$ for all observations. Therefore, taking expectations of $x_t = c + \alpha_1x_{t-1} + \alpha_2x_{t-2}+\ldots+\alpha_qx_{t-p} + e_t$ leads to $\mu = c + \alpha_1\mu + \alpha_2\mu+\ldots+\alpha_p\mu$ and to the expected mean $\mu = \frac{c}{1-\alpha_1-\alpha_2-\ldots-\alpha_p}$.

[22] An intuitive way to think about the stationary-condition can be reached by factorizing (15). (15) can be factorized into p AR(1) processes with autoregressive coefficients λ_1 to λ_p. The coefficients λ_1 to λ_p are the so called inverted roots. The original autoregressive coefficients α_1 to α_p can be calculated out of the inverted roots. In this sense, the inverted roots are the building blocks for the original autoregressive coefficients α_1 to α_p. For an AR(p) process to be stationary, all this (complex) inverted unit roots λ_1 to λ_p must lie inside the unit circle in absolute values, such that $|\lambda_i| < 1 \; \forall \; i = 1, \ldots, p$.

[23] For example the covariance for an AR(1) process would be $Cov(x_t, x_{t-1}) = E[x_tx_{t-1}] - E[x_t]E[x_{t-1}] = E\left[(c+\sum_{j=0}^{\infty}\alpha^j e_{t-j})(c+\sum_{j=0}^{\infty}\alpha^j e_{t-j-1})\right] - c^2 = E[(c+e_t+\alpha e_{t-1}+\alpha^2e_{t-2}+\alpha^3e_{t-3}\ldots)(c+e_{t-1}+\alpha e_{t-2}+\alpha^2e_{t-3}+..)] - c^2 = E[c^2+ce_{t-1}+c\alpha e_{t-2}+\ldots+e_tc+e_te_{t-1}+e_t\alpha e_{t-2}+\ldots+\alpha e_{t-1}c+\alpha e_{t-1}e_{t-1}+\alpha e_{t-1}\alpha e_{t-2}+\ldots+\alpha^2e_{t-2}c+\alpha^2e_{t-2}e_{t-1}+\alpha^2e_{t-2}\alpha e_{t-2}+\ldots+\alpha^3e_{t-3}c+\alpha^3e_{t-3}e_{t-1}+\alpha^3e_{t-3}\alpha e_{t-2}+\alpha^3e_{t-3}\alpha^2e_{t-3}] - c^2 = \alpha\sigma_e^2+\alpha^3\sigma_e^2+\alpha^5\sigma_e^2+\ldots = \alpha\sigma_e^2\sum_{j=0}^{\infty}\alpha^{2j} = \frac{\alpha\sigma_e^2}{1-\alpha^2} = \alpha\gamma(0)$, whereby $\gamma(s) = \alpha^s\sigma_e^2\sum_{j=0}^{\infty}\alpha^{2j} = \frac{\alpha^s\sigma_e^2}{1-\alpha^2}$ and therefore $\gamma(s)$ can be calculated out of $\gamma(s-1)$, for example $\gamma(2) = \alpha^2\sigma_e^2\sum_{j=0}^{\infty}\alpha^{2j} = \frac{\alpha^2\sigma_e^2}{1-\alpha^2} = \alpha\gamma(1)$.

when the roots of the polynomial of the error terms $\emptyset_q(B)$ exceed one in absolute value. To derive the properties for an ARIMA(p,0,q) process, they are illustrated for an ARIMA(1,0,1)(zc) process in the following. Therefore, the process $x_t = \alpha x_{t-1} + e_t + \beta e_{t-1}$ is re-written in terms of white noise components and with the backshift operator as $x_t = \alpha x_{t-1} + e_t + \beta e_{t-1} = (1-\alpha)^{-1}(1+\beta B)e_t$. This again can be expanded and reformulated to

$$x_t = (1 + \alpha B + \alpha^2 B^2 + \alpha^3 B^3 + \ldots)(1 + \beta B)e_t \quad (18a)$$

$$x_t = \sum_{i=0}^{\infty} \alpha^i B^i (1 + \beta B)e_t = (1 + \sum_{i=0}^{\infty} \alpha^{i+1} B^{i+1} + \sum_{i=0}^{\infty} \alpha^i \beta B^{i+1})e_t \quad (18b)$$

$$x_t = e_t + (\alpha + \beta) \sum_{i=1}^{\infty} \alpha^{i-1} e_{t-1} \quad (18c)$$

From (18c) can be seen that the mean is zero. The variance is given by

$$Var(x_t) = Var(e_t + (\alpha + \beta) \sum_{i=1}^{\infty} \alpha^{i-1} e_{t-1}) = \sigma_e^2 + \sigma_e^2 (\alpha + \beta)^2 (1 - \alpha^2)^2 \quad (19)$$

the covariance for $s > 0$ by

$$Cov(\mathrm{x_t}, \mathrm{x_t}) = (\alpha + \beta)\alpha^{s-1}\sigma_e^2 + (\alpha + \beta)^2 \sigma_e^2 \alpha^s \sum_{i=1}^{\infty} \alpha^{2i-2} \quad (20a)$$

$$Cov(\mathrm{x_t}, \mathrm{x_t}) = (\alpha + \beta)\alpha^{s-1}\sigma_e^2 + (\alpha + \beta)^2 \sigma_e^2 \alpha^s (1 - \alpha^2)^{-1} \quad (20b)$$

and therefore the autocorrelation by

$$\rho(k) = \frac{\alpha^{k-1}(\alpha+\beta)(1+\alpha\beta)}{1+\alpha\beta+\beta^2} \quad (21)$$

Equation (21) again implies that $\rho(s) = \alpha\rho(s-1)$ (Cowpertwait & Metcalfe, 2009, p. 127). Not only stationary AR, but all stochastic processes with a finite variance and without a stochastic trend can be transformed into MA(∞) processes. This leads to the Wold decomposition, which states that every stationary time series can be decomposed into a constant and a stationary part, whereby the stationary part is represented by a MA(∞) process, so that

$$x_t = c + \sum_{j=0}^{\infty} \psi_j e_{t-j} \quad (22a)$$

$$x_t = c + \Psi(B)e_t \quad (22b)$$

Such a time series displays a stationary process (SP) with constant, so that $x_t = SP$ (Gruber, 2011, p. 29).

3) Difference-stationary ARIMAs ($x_t = ST + SP$ and $x_t = DT + ST + SP$): For a difference-stationary time series, there exists a Wold form

$$\Delta x_t = \delta + \Psi(B)e_t \quad (23)$$

with $\Psi(1) \neq 0$ and $\sum_{j=0}^{\infty} |\psi_j| < \infty$, where the first condition ensures, that trend-stationary processes are excluded from the definition of difference-stationarity[24]. From (23), the Beveridge-Nelson-

[24] This is explained best with a small example for a trend-stationary process $x_t = (h + \delta t) + e_t - \alpha e_{t-1}$, where $\boldsymbol{e}$ is white noise. In such a process, the change in the differences would be $\Delta x_t = \delta + \Delta e_t + \alpha \Delta e_{t-1}$. For $\alpha = 1$, changes in differences of the error term Δe_t are reversed to 100% (bringing the time series back to the trend). For example, if $e_{t-1} = 0$, $e_t = 1$ and $e_{t+1} = 0$, then $\Delta e_t = 1$ and $\Delta e_{t+1} = -1$. Then the first change will shift the time series away from the trend, i.e.

Decomposition can be applied by transforming the infinite lag-polynomial $\Psi(B)$. Therefore, the lag-polynomial is subtracted by $\Psi(1)$, leading to

$$\Psi(B) - \Psi(1) = 1 + \Psi_1 B + \Psi_2 B + \Psi_3 B + \ldots - 1 - \Psi_1 - \Psi_2 - \Psi_3 - \ldots \quad (24a)$$

$$\Psi(B) - \Psi(1) = \Psi_1(B-1) + \Psi_2(B^2-1) + \Psi_3(B^3-1) + \cdots \quad (24b)$$

$$\Psi(B) - \Psi(1) = (B-1)[\Psi_1 + \Psi_2(B-1) + \Psi_3(B^2+B+1) + \ldots \quad (24c)$$

$$\Psi(B) - \Psi(1) = (B-1)[(\Psi_1 + \Psi_2 + \Psi_3 + \cdots) + (\Psi_2 + \Psi_3 + \Psi_4 + \cdots)B + (\Psi_3 + \Psi_4 + \Psi_5 + \ldots)B^2 +] = (B-1)\widetilde{\Psi}(B) \quad (24d)$$

with $\widetilde{\Psi}_j = \sum_{i=j+1}^{\infty} \Psi_i$. Therefore, $\Psi(B)$ can be rewritten as $\Psi(B) = \Psi(1) + (B-1)\widetilde{\Psi}(B)$, so that (23) becomes

$$\Delta x_t = \delta + [\Psi(1) + (B-1)\widetilde{\Psi}(B)]e_t \quad (25)$$

If $\boldsymbol{x}$ is integrated of order one, then it can be expressed as the sum of all changes as

$$x_t = x_0 + \sum_{i=1}^{t} \Delta x_i \quad (26)$$

Putting (25) into (26) leads to

$$x_t = x_0 + \sum_{i=1}^{t} \delta + \left[\Psi(1) + (B-1)\widetilde{\Psi}(B)\right]e_i \quad (27a)$$

$$x_t = x_0 + \delta t + \Psi(1)\sum_{i=1}^{t} e_i + \widetilde{\Psi}(B)e_0 - \widetilde{\Psi}(B)e_t = DT + ST + SP \quad (27b)$$

whereby $x_0 + \delta t = DT$, $\Psi(1)\sum_{i=1}^{t} e_i = ST$ and $\widetilde{\Psi}(B)e_0 - \widetilde{\Psi}(B)e_t = SP$. Therefore, every integrated process can be de-composed into a DT, a ST and a SP. While a ST and a SP is always a part of an integrated stochastic time series, DT can be present or not. If $\delta = 0$, then the time series would display the process $x_t = ST + SP$ and for $\delta \neq 0$, the time series would display the process $x_t = DT + ST + SP$ (Neusser, 2011, p. 113ff).

4) Trend-stationary ARIMAs ($x_t = DT + SP$): For a trend-stationary process $\Psi(1) = 0$ holds and therefore (27) leads to the Wold form

$$\mathrm{x_t} = x_0 + \delta t + \widetilde{\Psi}(B)e_0 - \widetilde{\Psi}(B)e_t \quad (28)$$

Changes by the process Δx_t are determined by δ (hence the trend is deterministic). Changes from this trend are stochastic and stationary, represented by the MA(∞) process $\widetilde{\Psi}(B)e_0 - \widetilde{\Psi}(B)e_t$. A trend-stationary process therefore is stationary around a deterministic trend. A trend-stationary process can be made stationary by subtracting the deterministic trend δt from the time series to receive a de-trended stationary time series $\tilde{x}_t$ with the Wold form $\tilde{x}_t = x_0 + \widetilde{\Psi}(B)e_0 - \widetilde{\Psi}(B)e_t$.

$\Delta x_t = \delta + \Delta e_t = \delta + 1$. But for $\alpha = 1$, the level change would be reversed to 100%, bringing the time series back to the trend, i.e. $\Delta x_{t+1} = \delta - \Delta e_{t-1} = \delta - 1$. If α is smaller than one, then the time series would not revers back to the trend and an error would have a persistent effect on the long run level of the time series. Therefore, for trend-stationary time series $\Psi(1) = 0$ must hold. In the example, $\Psi(B) = 1 - \alpha B$ and therefore $\Psi(1) = 1 - \alpha = 0$ holds for $\alpha = 1$. The same is true for higher order polynomials of the error term with a unit root, such that $\Psi(1) = 0$ (what means that these cases are trend-stationary and are excluded from the definition of difference-stationarity).

5) Trend- vs difference stationarity ARIMAs: As described above, differencing also can make a trend-stationary process stationary. The difference in forecasting a differenced trend-stationary process versus a difference-stationary process can be illustrated with a small example. The first differences of the trend-stationary process $x_t = h + \delta t + e_t$ are $\Delta x_t = \delta + \Delta e_t$, which is a stationary process around the mean δ[25]. Supposed that $e_t = 1$, the forecast of such a (differenced) trend-stationary time series would be $E[\Delta x_{t+1}] = E[\delta] + E[\Delta e_{t+1}]$. With $E[e_t] = 1$ and $E[e_{t+1}] = 0$, the expected change of the error term becomes $E[\Delta e_{t+1}] = -1$ and therefore the expected change of the time series becomes $E[\Delta x_{t+1}] = \delta - 1$. In contrast, the forecast of a difference-stationary random walk $x_t = x_{t-1} + e_t$ would be $E[\Delta x_{t+1}] = \delta$[26]. Therefore, if a trend-stationary process is falsely differenced and regarded as a I(1) process, the forecast for the first differences of this trend-stationary process would be erroneously $E[\Delta x_{t+1}] = \delta$, while the true forecast should be $E[\Delta x_{t+1}] = \delta - 1$ (Pfaff et al., 2008, p. 53f).

C) Unit root tests

As described, ARMA models can be applied to the levels or to the differences of the time series (depending on whether the time series is integrated or not). In order to detect whether a time series is integrated, i.e. in order to test whether a time series contains a unit root, there are so-called unit root tests. ***Two commonly used tests*** for unit root detection are ***1)*** the Augmented Dickey Fuller (ADF) test and ***2)*** the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test. These tests will briefly be described in the following.

1) ADF test: For the ADF test, the $H_0: \alpha_1 = 1$ is tested for the following equation

$$x_t = c + \alpha_1 x_{t-1} + \alpha_2(x_{t-2} - x_{t-1}) + \ldots + \alpha_{p+1}(x_{t-p} - x_{t-p-1}) \quad (29)$$

This means that it is tested whether the time series contains a random walk part (in which case $\alpha_1 = 1$). Additionally, the ADF test assumes, that the differenced series is a stationary process, which can be approximated by an AR(p) process. Because $\hat{\alpha}_1$ does not follow a standard distribution, the critical values have been approximated by simulations. Additionally, the limiting distributions also depend on the deterministic terms, so when deterministic terms are added to (29), then the simulations will yield other critical values[27] (Schneider; Mentemeier, 2010, p. 41f).

2) KPSS test: For the KPSS test, the $H_0: y_t \sim I(0)$ is tested against the alternative hypothesis that the series is integrated of order one. It is assumed that $y_t = x_t + g_t$, whereby x_t is the random walk

[25] A linear combination of two stationary processes is itself stationary. Therefore, Δe_t is stationary.

[26] The differences are $\Delta x_t = \delta + e_t$. Taking expectations results in $E[\Delta x_t] = \delta$.

[27] If a linear trend is needed in the test for y_t, then only a constant must be added, because if $y_t = h + \delta t + x_t$, then $\Delta y_t = \delta + \Delta x_t$. Similarly, if it should be included in testing for y_t, if the deterministic part in y_t is only a constant ($y_t = h + x_t$), then no deterministic part needs to be included in (19).

component with $x_t = x_{t-1} + e_t$, where e_t is white noise and g_t is a stationary process. If a time series has a stochastic trend (or a random walk component), then the variance is dependent on time and will get larger with time (as described above). Let $\widehat{\sigma_\infty^2}$ be an estimator for the long time variance of the stationary process z_t, then, when y_t contains a random walk component, the variance of y_t will be much higher than the (long term) variance of a stationary process. In this case, the KPSS test statistic, which is defined as

$$KPSS = \frac{\sum_{t=1}^{T} S_t^2}{T^2 \widehat{\sigma_\infty^2}} \tag{30}$$

(where $S_t^2 = \sum_{j=1}^{t} \widehat{\omega}_j$ and $\widehat{\omega}_j = x_t - \mu$) will converge to zero. The long term variance of g_t is estimated as

$$\widehat{\sigma_\infty^2} = \frac{1}{T}\sum_{t=1}^{T} \widehat{\omega}_j{}^2 + 2\sum_{j=1}^{l_d} \eta_j \left(\frac{1}{T}\sum_{t=j+1}^{T} \widehat{\omega}_t \widehat{\omega}_{t-j}\right) \tag{31}$$

with $\eta_j = 1 - \frac{j}{l_d+1}$ and $l_d = q(\frac{T}{100})^{1/4}$, where $q = 4$ or $q = 12$ is suggested. Thus, the null hypothesis of stationarity is rejected for large values of the KPSS test statistic. The critical values are received from simulations and are also dependent on whether a deterministic trend is suspected or not. (Schneider; Mentemeier, 2010, p. 41f & Lütkepohl; Krätzig, 2004, p. 63f).

2.1.2.3. VAR models

The following VAR models will consider multiple time series in order to study their dynamic relationships. This section is divided into ***three subparts***. To illustrate the concepts of VAR models, some intuitions are provided in part ***A)***. Thereafter part ***B)*** provides some mathematical formulations. A small part ***C)*** will deal with the question of which VAR model to choose.

A) Intuitions

This part A) contains ***four subparts*** dealing with ***1)*** deterministic as well as stochastic trends and stationarity, ***2)*** stationary VARs, ***3)*** difference-stationary VARs and ***4)*** trend-stationary VARs.

1) Deterministic trends (DT), stochastic trends (ST) and stationary processes (SP): A k-dimensional VAR(p) model combines k time series in that way, that every time series is not only dependent on its own lagged values (its auto-correlations), but also on lagged correlations of the other k-1 variables (in the following called cross-correlations). Additionally, it is assumed that every time series has "its own" error term, so that there are k different error terms (whereby it is not required that the error terms are *contemporaneously* uncorrelated, but it is required that the *lagged* cross-correlations of the k error terms are uncorrelated). The order p displays (like for the univariate ARIMA models) how many lagged auto- and cross-correlations are considered (Sheppard, 2010, p.322).

Analogous to univariate processes, multivariate processes can be decomposed into a multivariate DT, a multivariate ST and a multivariate SP, such that $\boldsymbol{x_t} = T + SP = DT + ST + SP$. In order to illustrate the behavior of VAR models, the simplest VAR, a two dimensional VAR(1), is considered in the following, defined as

$$\begin{aligned} x_{1t} &= c_1 + \alpha_{11}x_{1t-1} + \alpha_{12}x_{2t-1} + e_{1t} \\ x_{2t} &= c_2 + \alpha_{21}x_{1t-1} + \alpha_{22}x_{2t-1} + e_{2t} \end{aligned} \tag{32}$$

whereby c_1 and c_2 could also be time-dependent in the case of a deterministic trend. Non-stationarity for VAR models can have ***two causes: a)*** an exploding process or ***b)*** the presence of a trend. ***a)*** For the case that $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are not just simple AR processes (α_{11} and α_{21} are not zero) the levels of the time series are dependent from each other. Therefore, whether the time series $\boldsymbol{x_1}$ is exploding or not, not only depends on α_{11} and α_{12}, but also on the fact, if $\boldsymbol{x_2}$ is exploding or not (and hence on α_{21} and α_{22}). Therefore, the coefficients must be in that way that both time series have a reversion to the mean. ***b)*** As described, trends can be deterministic or stochastic. Following the definitions of trend- and difference-stationary time series for the univariate case, a multivariate trend-stationary VAR process would be a process, where the time series are first de-trended, and subsequently a VAR is applied to the de-trended time series (corresponding to $\boldsymbol{x_t} = DT + SP$). Additionally, a difference-stationary VAR would be a process where the time series are first differenced (one time in most cases), and subsequently a VAR is applied to the differenced time series (corresponding to $\boldsymbol{x_t} = ST + SP$ for a stationary VAR in differences around a zero mean and $\boldsymbol{x_t} = DT + ST + SP$ for a stationary VAR in differences around a non-zero mean).

2) Stationary VARs ($\boldsymbol{x_t} = \boldsymbol{SP}$): For a VAR to be stationary there must be some kind of reversion to the mean for both variables. For $\alpha_{12} = 0$ and $\alpha_{21} = 0$ $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ would be two independent AR(1) processes (so that the reversion will follow the properties for univariate time series described in 2.1.2.2). But for the general case, where $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are inter-dependent (α_{12} and α_{21} are not zero), the reversion is dependent on all coefficients α_{11}, α_{12}, α_{21} and α_{22}. Thereby, the reversion to the mean can be alternating or non-alternating. If $\alpha_{12} \neq 0$ and α_{11} is big, compared to α_{12}, then the reversion of $\boldsymbol{x_1}$ is more dependent on its own lagged values (making $\boldsymbol{x_1}$ almost a simple AR(1) process). In this case, the reversion of $\boldsymbol{x_1}$ is non-alternating for an α_{11} greater than zero and alternating for an α_{11} smaller than zero[28]. This is illustrated in Figure A.3.1 on page 67, where two VAR(1) are shown with their auto- and cross-regressive parts.

[28] The same reasoning holds for α_{22} and α_{21} with regard to $\boldsymbol{x_2}$, only that the autocorrelation part of $\boldsymbol{x_1}$ is α_{11}, whereas the autocorrelation part of $\boldsymbol{x_2}$ is α_{22}.

If $\boldsymbol{x_2}$ is more dependent on the lagged values of $\boldsymbol{x_1}$ than on its own lagged values, $\boldsymbol{x_2}$ is somehow following $\boldsymbol{x_1}$. If $\boldsymbol{x_1}$ is an almost simple AR(1) process, then, α_{11} and α_{21} are the important coefficients for the system. In this case, if α_{11} and α_{21} have the same sign, then $\boldsymbol{x_2}$ and $\boldsymbol{x_1}$ are positively correlated, because, for example, if both are positive, then a positive shock of $\boldsymbol{e_1}$ will result in a positive effect on the levels of $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$. While if α_{11} and α_{12} have opposite signs, then $\boldsymbol{x_2}$ and $\boldsymbol{x_1}$ are negatively correlated, because, for example if α_{11} is positive and α_{21} is negative, then a positive shock of $\boldsymbol{e_1}$ will result in a positive effect on the level of $\boldsymbol{x_1}$, but in a negative effect on the level of $\boldsymbol{x_2}$. In cases where $\boldsymbol{x_2}$ is following $\boldsymbol{x_1}$, the reversion to the mean is non-alternating for both processes if α_{11} is greater than zero. This is illustrated in Figure A.3.2 on page 68, where two VAR(1) are shown – one with positive and one with negative correlations between $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$.

In contrast to non-alternating reversions to the mean (still for the case, where $\boldsymbol{x_2}$ is following $\boldsymbol{x_1}$), the reversion is alternating for both processes if α_{11} is smaller than zero. For a timely contemporaneous alternation, $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ would be positively correlated, while for a timely contrarian alternation, $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ would be negatively correlated. This is shown in Figure A.3.3 on page 68, where two VAR(1)s with a contemporaneous alternating and a contrarian alternating reversion to the mean are shown.

So far, the constant has been set to zero. When the constants are not both zero and the two time series are dependent on each other (α_{12} and α_{21} are not zero), then both time series will be shifted up- or downwards, depending on the constants. Thereby, only one constant needs to be different from zero to shift both time series up or down. This is the case, because, for example for $c_1 \neq 0$ and $c_2 = 0$, $\boldsymbol{x_1}$ and also $\boldsymbol{x_2}$ will be shifted, because $\boldsymbol{x_2}$ depends on the (shifted) level of $\boldsymbol{x_1}$. Thereby, the shift of $\boldsymbol{x_1}$ does not only depend on c_1 and its own autoregressive coefficient α_{11}, but also on the (now shifted) level of $\boldsymbol{x_2}$ (which is also dependent on its own autocorrelation coefficient α_{22}). Therefore, for a two-dimensional stationary VAR with non-zero constants, the mean of the time series will depend on the two constants c_1 and c_2, and also on all four coefficients α_{11}, α_{12}, α_{21} and α_{22}. An example is provided in Figure A.3.4 on page 69, where a VAR with one constant and a VAR with two constants different from zero is shown.

3) Difference-stationary VARs ($\boldsymbol{x_t = ST + SP}$ and $\boldsymbol{x_t = DT + ST + SP}$): Because VAR models should only be applied to variables with the same order of integration, ***two cases*** can be distinguished. ***a)*** VAR models can be either applied to the *levels of all* time series (called a VAR in levels), ***b)*** VAR models can be applied to the *differences of all* time series. Thereby, the cases ***b1)***, where a relationship between the levels is considered (called a VEC model) and ***b2)***, where a relationship between the levels is not considered (called a VAR in differences), can be distinguished. The two applications of VAR models are described in the following: ***a)*** If a VAR applied to the levels is stationary, then there would be no need

for differencing any of the time series (VAR in levels). ***b1)*** If the levels of the time series follow stochastic trends, then a relationship between the stochastic trends could be considered. If this is the case, then, for example for a two-dimensional VAR(1), a change in the level of one variable (let us say $\boldsymbol{x_1}$) will provoke a change in the level of the other variable ($\boldsymbol{x_2}$). If $\boldsymbol{x_1}$ contains a stochastic trend, then changes caused by an error term are to some extend perpetually on the long run for the level of $\boldsymbol{x_1}$. If $\boldsymbol{x_2}$ is dependent on $\boldsymbol{x_1}$ (so that α_{21} is not zero), then a (perpetual) change in the levels of $\boldsymbol{x_1}$ (caused by the stochastic trend) also would provoke a perpetual change in the level of $\boldsymbol{x_2}$ (and therefore $\boldsymbol{x_2}$ also would follow a stochastic trend). In this case, there would be a relationship between the stochastic trends. The same arguments would be true, if the series contained a stochastic trend and a deterministic trend (a stochastic trend with drift). Integrated time series with relationships between the stochastic trends should be modeled with VEC models as described in 2.1.2.4[29]. ***b2)*** If one does not want to consider possible relationships in levels, then (all) time series can be first differenced and subsequently a VAR can be applied to the differenced processes (VAR in differences). An example of a two-dimensional VAR in differences is shown in Figure A.3.5 on page 69.

4) Trend-stationary VARs ($\boldsymbol{x_t = DT + SP}$): Similar to the univariate case, a (linear) trend could be present in one or more time series. If one time series contains a deterministic trend and the time series is inter-dependent, then a VAR in levels will not be stationary (because analogous to the arguments above, a change in levels of one variable due to the trend would provoke changes in the levels of the other variables. In order to make the time series stationary, they could be de-trended and subsequently a VAR could be applied to the de-trended time series, assuming that the time series deviations from the trend line follow a VAR model. This would mean, for example for a two-dimensional VAR(1), that the deviations from the trend lines of $\boldsymbol{x_1}$ would be dependent on its own past deviations from the trend line plus the lagged deviations from the trend line of $\boldsymbol{x_2}$. An example of such a process, where $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ have been de-trended and subsequently a VAR(1) was applied, is shown in Figure A.3.6 on page 70.

B) Mathematical formulations

This part B) also contains ***four subparts*** dealing with ***1)*** deterministic as well as stochastic trends and stationarity, ***2)*** stationary VARs, ***3)*** difference-stationary VARs and ***4)*** trend-stationary VARs.

1) Deterministic trends (DT), stochastic trends (ST) and stationary processes (SP): Weak stationarity in the multivariate case is defined similar to the univariate case as

$$E[\boldsymbol{x}_t] = \boldsymbol{\mu}, \forall\, t \tag{33a}$$

$$E[(\boldsymbol{x}_t - \boldsymbol{\mu})(\boldsymbol{x}_{t-s} - \boldsymbol{\mu})] = \boldsymbol{\Gamma}_s, \forall\, \mathrm{t, s} \tag{33b}$$

[29] Through simply differencing the time series, the information about a possible relationship between the levels of the time series would be lost. But this information could be used to improve the model and the forecasts. Therefore, if present, such information should be used by applying a VEC model as described in 2.1.2.4.

where the covariance function is defined as

$$\boldsymbol{\Gamma}_s = \begin{bmatrix} \gamma_{11}(s) & \dots & \gamma_{1k}(s) \\ \vdots & \ddots & \vdots \\ \gamma_{k1}(s) & \dots & \gamma_{kk}(s) \end{bmatrix} \tag{34}$$

In the following, it will be shown how a multivariate time series $\boldsymbol{x_t}$ (analogous to the univariate case) can be decomposed in a trend (T) and a stationary process (SP), whereby T can be a deterministic trend (DT) or a stochastic trend (ST), so that

$$\boldsymbol{x_t} = T + SP = DT + ST + SP \tag{35}$$

Similar to the univariate case, while a SP is always part of stochastic multivariate time series, DT and ST can be present or not, which leads to the possible combination: $\boldsymbol{x_t} = SP$, $\boldsymbol{x_t} = ST + SP$, $\boldsymbol{x_t} = DT + SP$ and $\boldsymbol{x_t} = DT + ST + SP$.

2) Stationary VARs ($\boldsymbol{x_t} = \boldsymbol{SP}$): If at every point in time t, k different variables are observed, so that $\boldsymbol{x_t} = (x_{1t} + \dots + x_{kt})$, then a VAR(p) process is defined as

$$\boldsymbol{x_t} = \boldsymbol{c} + \boldsymbol{\Phi}_1 \boldsymbol{x}_{t-1} + \dots + \boldsymbol{\Phi}_p \boldsymbol{x}_{t-p} + \boldsymbol{e_t} \tag{36}$$

where $\boldsymbol{\Phi}_i$ are (k x k) dimensional coefficient matrices for $\boldsymbol{i} = 1, \dots, p$, $\mathbf{e}_t$ is a k-dimensional white noise process[30] and $\boldsymbol{c}$ is a (k x 1) vector of time-independent intercept or time-dependent deterministic trend parameters. A VAR(p) can be displayed analogous to the univariate AR(p) with a matrix backshift polynomial, so that

$$\boldsymbol{\Phi(B)} \boldsymbol{x_t} = \boldsymbol{c} + \boldsymbol{e_t} \tag{37}$$

where the matrix backshift polynomial is $\boldsymbol{\Phi(B)} = \boldsymbol{I}_n - \boldsymbol{\Phi}_1 \boldsymbol{B} - \boldsymbol{\Phi}_2 \boldsymbol{B}^2 - \dots - \boldsymbol{\Phi}_p \boldsymbol{B}^p$. For a VAR(p) to be stationary, all roots of

$$|\boldsymbol{I}_n - \boldsymbol{\Phi}_1 \boldsymbol{z} - \boldsymbol{\Phi}_2 \boldsymbol{z}^2 - \dots - \boldsymbol{\Phi}_p \boldsymbol{z}^p| \tag{38}$$

must lie outside the unit circle. Similarly, stationarity can also be tested by calculating the eigenvalues of the so called companion form, whereby a VAR(p) is rewritten in a stacked form as a VAR(1) process as

$$\boldsymbol{\xi_t} = \mathbf{A} \boldsymbol{\xi}_{t-1} + \boldsymbol{v_t} \tag{39}$$

where

$$\boldsymbol{\xi}_t = \begin{bmatrix} \boldsymbol{x_t} \\ \vdots \\ \boldsymbol{x}_{t-p+1} \end{bmatrix} \qquad \boldsymbol{A} = \begin{bmatrix} \boldsymbol{\Phi_1} & \boldsymbol{\Phi_2} & \dots & \boldsymbol{\Phi}_{p-1} & \boldsymbol{\Phi}_p \\ \boldsymbol{I} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \boldsymbol{I} & \mathbf{0} \end{bmatrix} \tag{40a}$$

and

[30] With $E[\boldsymbol{e_t}] = \mathbf{0}_k, E[\boldsymbol{e_t}, \boldsymbol{e'}_{t-s}] = \mathbf{0}_{kxk}$ and $\boldsymbol{E}[\boldsymbol{e_t}, \boldsymbol{e'}_t] = \Sigma$, where Σ is a positive definite matrix.

$$\boldsymbol{v}_t = \begin{bmatrix} \boldsymbol{e}_t \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} \qquad E[\boldsymbol{\xi}_t \boldsymbol{\xi}_t{}'] = \begin{bmatrix} \boldsymbol{\Sigma} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & & \mathbf{0} \\ \vdots & & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \tag{40b}$$

where $\boldsymbol{\xi}_t$ and $\mathbf{v}_t$ is of dimension kp x 1 and $\boldsymbol{A}$ is of dimension kp x kp. If the moduli of the eigenvalues of $\boldsymbol{A}$ are less than one, then the VAR(p) process is stationary (Pfaff et al., 2008, p. 23ff). In the following, the stationarity condition, as well as the properties of the VAR models are illustrated on a VAR(1) process, defined as

$$\boldsymbol{x}_t = \boldsymbol{c} + \boldsymbol{\Phi}_1 \boldsymbol{x}_{t-1} + \boldsymbol{e}_t \tag{41}$$

Such a VAR(1) then can be transformed into a MA(∞) form

$$\boldsymbol{x}_t = (\boldsymbol{I}_n - \boldsymbol{\Phi}_1)^{-1}\boldsymbol{c} + (\boldsymbol{I}_n - \boldsymbol{\Phi}_1)^{-1}\boldsymbol{e}_t = \sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \, \boldsymbol{c} + \sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \, \boldsymbol{e}_t \tag{42}$$

In this case, the matrix $\boldsymbol{A}$ of the companion form is just equal to $\boldsymbol{\Phi}_1$, so that for stationarity the eigenvalues of $\boldsymbol{\Phi}_1$ must all be less than one in absolute value. The eigenvalue condition ensures that $\boldsymbol{\Phi}_1^i$ converges to zero as i grows large[31]. Because $\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \, \boldsymbol{e}_t = \sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \, \mathbf{0} = \mathbf{0}$, taking expectations from (42) leads to the mean of the VAR(1) process, defined as

$$E[\boldsymbol{x}_t] = (\boldsymbol{I}_n - \boldsymbol{\Phi}_1)^{-1}\boldsymbol{c} \tag{43}$$

The eigenvalues of $\boldsymbol{\Phi}_1$ are also important for determining the mean, because if an eigenvalue of $\boldsymbol{\Phi}_1$ is close to one, then $(\boldsymbol{I}_n - \boldsymbol{\Phi}_1)^{-1}$ will contain large values and the unconditional mean will be large (similar to the mean of a univariate AR(1) process $(1-\alpha)^{-1}c$). To derive the variance of a VAR(1) process, it is helpful to express the VAR in deviations form by subtracting the mean (or the time-dependent deterministic trend). Because that de-meaned time series $\tilde{\mathbf{x}}_t$ will have zero means, the expectation of (42) simplifies to $E[\tilde{\boldsymbol{x}}_t] = E[\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \, \boldsymbol{e}_t]$, so that the variance can be calculated as

$$E[(\boldsymbol{x}_t - \boldsymbol{\mu})(\boldsymbol{x}_t - \boldsymbol{\mu})'] = E[\tilde{\boldsymbol{x}}_t \tilde{\boldsymbol{x}}_t'] = E[(\textstyle\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \, \boldsymbol{e}_{t-i})\big(\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \, \boldsymbol{e}_{t-i}\big)'] \tag{44a}$$

$$E[(\boldsymbol{x}_t - \boldsymbol{\mu})(\boldsymbol{x}_t - \boldsymbol{\mu})'] = E[\textstyle\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \, \boldsymbol{e}_{t-i}\boldsymbol{e}_{t-i}'\big(\boldsymbol{\Phi}_1^i\big)'] \tag{44b}$$

$$E[(\boldsymbol{x}_t - \boldsymbol{\mu})(\boldsymbol{x}_t - \boldsymbol{\mu})'] = \textstyle\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \, E[\boldsymbol{e}_{t-i}\boldsymbol{e}_{t-i}']\big(\boldsymbol{\Phi}_1^i\big)' = \sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \, \Sigma\big(\boldsymbol{\Phi}_1^i\big)' \tag{44c}$$

Once again, the eigenvalues of $\boldsymbol{\Phi}_1$ play an important role. Because the eigenvalues determine the persistence of the shocks, an eigenvalue close to one means higher persistence and therefore a higher variance. The autocovariance matrices, defined as

$$\boldsymbol{\Gamma}_s = E[(\boldsymbol{x}_t - \boldsymbol{\mu})(\boldsymbol{x}_{t-s} - \boldsymbol{\mu})'] \text{ and } \boldsymbol{\Gamma}_{-s} = E[(\boldsymbol{x}_t - \boldsymbol{\mu})(\boldsymbol{x}_{t+s} - \boldsymbol{\mu})'] \tag{45}$$

are not symmetric anymore in the multivariate case. Instead, they are symmetric in their transpose, so that $\boldsymbol{\Gamma}_s \neq \boldsymbol{\Gamma}_{-s}$ but $\boldsymbol{\Gamma}_s = \boldsymbol{\Gamma}_s'$. They can be calculated from (39) as

[31] It can be shown, that if all eigenvalues of $\boldsymbol{A}$, λ_i, for $i = 1,2,\ldots,k$, are less than 1 in modulus ($|\lambda_i| < 1$), then the series $\sum_{j=0}^{m} \boldsymbol{A}^m = \boldsymbol{I}_k + \boldsymbol{A} + \boldsymbol{A}^2 + \boldsymbol{A}^3 + \ldots + \boldsymbol{A}^m \rightarrow (\boldsymbol{I}_n - \boldsymbol{A})^{-1}$ as $m \rightarrow \infty$, because $\boldsymbol{A}^m \rightarrow 0$ as $m \rightarrow \infty$.

$$\boldsymbol{\Gamma}_s = E[(\boldsymbol{x}_t - \boldsymbol{\mu})(\boldsymbol{x}_{t-s} - \boldsymbol{\mu})'] = E[(\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \boldsymbol{e}_t)(\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \boldsymbol{e}_{t-s})'] \quad (46a)$$

$$\boldsymbol{\Gamma}_s = E[(\sum_{i=0}^{s-1} \boldsymbol{\Phi}_1^i \mathbf{e}_{t-i})(\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \mathbf{e}_{t-i-s})' + (\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^s \boldsymbol{\Phi}_1^{\boldsymbol{i}} \mathbf{e}_{t-i-s})(\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^{\boldsymbol{i}} \mathbf{e}_{t-i-s})'] \quad (46b)$$

$$\boldsymbol{\Gamma}_s = \mathbf{0} + \boldsymbol{\Phi}_1^s E\left[(\sum_{i=0}^{s-1} \boldsymbol{\Phi}_1^{\boldsymbol{i}} \boldsymbol{e}_{t-i-s})(\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^{\boldsymbol{i}} \boldsymbol{e}_{t-i-s})'\right] = \boldsymbol{\Phi}_1^s \boldsymbol{V}[\boldsymbol{x}_t] \quad (46c)$$

where $\boldsymbol{\Phi}_1^s \boldsymbol{V}[\mathbf{x}_t]$ is the symmetric covariance matrix of $\mathbf{x}_t$[32]. This result is again similar to the auto-covariance function of an AR(1) process $\gamma_s = \alpha^s \sigma_e^2 (1-\alpha^2) Var[x_t]$ (Sheppard, 2010, p. 323ff).

Like an univariate stationary process, also a multivariate stationary VAR(p) process can be written in the Wold form as

$$\boldsymbol{x}_t = \boldsymbol{\Psi}_0 + \boldsymbol{\Psi}_1 \boldsymbol{e}_{t-1} + \boldsymbol{\Psi}_2 \boldsymbol{e}_{t-2} + \ldots \quad (47)$$

displaying a stationary process with a constant, so that $\boldsymbol{x}_t = SP$.

3) Difference-stationary VARs ($\boldsymbol{x}_t = \boldsymbol{ST} + \boldsymbol{SP}$) and ($\boldsymbol{x}_t = \boldsymbol{DT} + \boldsymbol{ST} + \boldsymbol{SP}$): Analogous to the univariate case, a multivariate integrated process can be de-composed into

$$\boldsymbol{x}_t = \boldsymbol{x}_0 + \boldsymbol{\delta} t + \boldsymbol{\Psi}(\mathbf{1}) \sum_{i=1}^{t} \boldsymbol{e}_i + \widetilde{\boldsymbol{\Psi}}(\boldsymbol{B}) \boldsymbol{e}_0 - \widetilde{\boldsymbol{\Psi}}(\boldsymbol{B}) \boldsymbol{e}_t = DT + ST + SP \quad (48)$$

so that every multivariate integrated process can be de-composed into a DT, a ST and a SP (again, while a ST and a SP are always part of a multivariate integrated stochastic time series, DT can be present or not. If $\boldsymbol{\delta} = \mathbf{0}$, then the time series will be $\boldsymbol{x}_t = ST + SP$ and for $\boldsymbol{\delta} \neq \mathbf{0}$, the time series will be $\boldsymbol{x}_t = DT + ST + SP$.

4) Trend-stationary VARs ($\boldsymbol{x}_t = \boldsymbol{DT} + \boldsymbol{SP}$): For a trend-stationary process $\boldsymbol{\Psi}(\mathbf{1}) = \mathbf{0}$, and therefore (48) leads to

$$\boldsymbol{x}_t = \boldsymbol{x}_0 + \boldsymbol{\delta} t + \widetilde{\boldsymbol{\Psi}}(\boldsymbol{B}) \boldsymbol{e}_0 - \widetilde{\boldsymbol{\Psi}}(\boldsymbol{B}) \boldsymbol{e}_t \quad (49)$$

The trend of the time series is determined by the function $\boldsymbol{\delta t}$. Changes from this trend are stochastic and stationary, represented by the MA(∞) part $\widetilde{\boldsymbol{\Psi}}(\boldsymbol{B}) \boldsymbol{e}_0 - \widetilde{\boldsymbol{\Psi}}(\boldsymbol{B}) \boldsymbol{e}_t$. Multivariate trend-stationary processes are therefore stationary around its deterministic trends, whereby subtracting the deterministic trends $\boldsymbol{\delta t}$ from the time series leads to stationary processes $\widetilde{\boldsymbol{x}}_t$ with the Wold form $\widetilde{\boldsymbol{x}}_t = \boldsymbol{x}_0 + \widetilde{\boldsymbol{\Psi}}(\boldsymbol{B}) \boldsymbol{e}_0 - \widetilde{\boldsymbol{\Psi}}(\boldsymbol{B}) \boldsymbol{e}_t$ (Neusser, 2011, p.230ff).

C) VAR in differences, VAR in levels or VEC model

The choice, which kind of VAR to choose, can be justified as follows. In order to test the order of integration, ADF or KPSS tests as described in 2.1.2.2 C) can be univariately applied to the time series. Subsequently, depending on the order of the time series (which has to be the same for all series), a VAR

[32] $\boldsymbol{\Gamma}_{-s}$ can be calculated similarly as $\boldsymbol{\Gamma}_{-s} = E[(\boldsymbol{x}_t - \boldsymbol{\mu})(\boldsymbol{x}_{t+s} - \boldsymbol{\mu})'] = E[(\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \boldsymbol{e}_{t-i})(\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \boldsymbol{e}_{t-i+s})'] = E[(\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \boldsymbol{e}_{t-i}) + E\left[(\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \boldsymbol{e}_{t-i+k})(\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^s \boldsymbol{\Phi}_1^i \boldsymbol{e}_{t-i+s})'\right] = \mathbf{0} + E[(\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \boldsymbol{e}_{t-i})(\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^s \boldsymbol{\Phi}_1^i \boldsymbol{e}_{t-i})'] = E[(\sum_{i=0}^{s-1} \boldsymbol{\Phi}_1^i \boldsymbol{e}_{t-i-s})(\sum_{i=0}^{\infty} \boldsymbol{\Phi}_1^i \boldsymbol{e}_{t-i-s})'](\boldsymbol{\Phi}_1')^s = \boldsymbol{V}[\boldsymbol{x}_t](\boldsymbol{\Phi}_1')^s$.

in levels (in case that all series are integrated of order zero), a VAR in differences (in case that all variables are integrated of order one and no co-integration should be considered or was detected) or a VEC model (as described in the following) can be applied.

2.1.2.4. VEC models

This section is divided into ***three subparts*** A), B) and C). To illustrate the concepts of VEC models some intuitions are provided in part ***A)***, where four cases with different co-integration relationships will be described. Thereafter, part ***B)*** provides further mathematical formulations. Part ***C)*** then presents a popular test (the Johansen test) to identify co-integration relationships and to model VEC models.

A) Intuitions

The VEC model not only considers the relationships modeled by the VAR coefficients, but also corrects for co-integration relationships. In *most cases* a linear combination of integrated processes (with the same order of integration) is again integrated of the same order like the processes itself[33]. But, in special cases, a linear combination of integrated processes can exist, which is of a smaller order than the integrated processes. In such a case, the time series are called co-integrated. Co-integration can be explained by so-called common trends[34]. A prerequisite for co-integration is that the processes are integrated. If the processes are integrated, then they follow a stochastic trend (next to eventually a deterministic trend). Co-integration then can emerge when some processes follow the same stochastic trend (it can be imagined that one or more underlying stochastic trends are present, where the k processes are linked to one or more of these underlying stochastic trends, such that they are sharing these common (stochastic) trends). In order to get some intuitions about co-integration and common trends, the following ***four cases*** will be described: ***1)*** two variables with one common trend, ***2)*** two variables with two common trends, ***3)*** three variables with one common trend and ***4)*** three variables with two common trends.

1) Two variables with one common trend: The following co-integrated bivariate system with one common trend $\boldsymbol{w}$ is assumed

$$x_{1t} = \lambda_1 w_t + e_{1t} \sim I(1);\ x_{2t} = \lambda_2 w_t + e_{2t} \sim I(1);\ w_t = w_{t-1} + e_{3t} \sim I(1) \qquad (50)$$

where $\boldsymbol{e_i}$, for $i = 1, 2, 3$, are white noise processes. In this example the two processes are just multiples of the underlying common trend $\boldsymbol{w}$. In such a case, a linear combination F of both variables with a smaller order of integration (I(0)) than the processes itself (I(1)) can be found, so that $F = \beta_1 \boldsymbol{x_{1t}} +$

[33] While it is always true, that a linear combination of two I(0) processes is itself a I(0) process.

[34] In most practical cases – as in this thesis in the following – the integrated processes are of order one (I(1)) and, if co-integrated, the linear combination (then called co-integration relationship) is of order zero (I(0)), and therefore stationary.

$\beta_2 \boldsymbol{x}_{2t} \sim I(0)$[35]. If *one* such a linear combination F is found, then an *infinite* amount of stationary linear combinations of the two processes exist, because if $\beta_1 \boldsymbol{x}_{1t} + \beta_2 \boldsymbol{x}_{2t} \sim I(0)$, then also $\lambda(\beta_1 \boldsymbol{x}_{1t} + \beta_2 \boldsymbol{x}_{2t}) \sim I(0)$. In order to reach unambiguity, the coefficients are normalized by setting the first coefficient β_1 to one, so that $\beta_1 = 1$ and $\beta_2 = \frac{\beta_2}{\beta_1}$. Therefore, for the case of two processes and one common trend as depicted in (50), there is one normalized co-integration relationship $F = \boldsymbol{x}_{1t} - \frac{\lambda_2}{\lambda_1} \boldsymbol{x}_{2t} \sim I(0)$[36]. The number of unique linear independent co-integration relationships is denoted as r and also called the rank of co-integration, so that in the case of two variables and one common trend $r = 1$. An example of a bivariate co-integrated process is given in Figure A.4.1 on page 70, where an underlying random walk, as well as two co-integrated processes (sharing a common random walk) are shown.

2) Two variables with two common trends: With two variables there exists no possibility that there are two common underlying trends, so that the two processes are co-integrated, given that the common trends are present with different weights in the processes. To illustrate this, two variables and two (not necessarily common) underlying trends can be imagined, whereby ***three* cases** can be distinguished. ***a)*** In a first case, the two variables can have *non-common* trends, so that $\boldsymbol{x}_1(\boldsymbol{w}_1)$ and $\boldsymbol{x}_2(\boldsymbol{w}_2)$. In this case, the two variables are independently integrated and cannot be co-integrated[37]. ***b)*** In a second case, one process $\boldsymbol{x}_1$ has two I(1) parts ($\boldsymbol{w}_1$ and $\boldsymbol{w}_2$) and the second process $\boldsymbol{x}_2$ only shares one I(1) part, so that $\boldsymbol{x}_1(\boldsymbol{w}_1, \boldsymbol{w}_2)$ and $\boldsymbol{x}_2(\boldsymbol{w}_2)$. In this case, there would be one common trend and one non-common trend, whereby the non-common trend could not be cancelled out, and therefore, the two processes also cannot be co-integrated. ***c)*** As a last possibility, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ could have two I(1) parts ($\boldsymbol{w}_1$ and $\boldsymbol{w}_2$). In this case, there would be two common trends, but the two processes are also not co-integrated if the trends are not present with the same weights in both processes, so that the two common trends can cancel out in a linear combination *F*. If the trends are present with the same weights, then there is just *one true* underlying trend,[38] so that for two variables, there cannot be two true underlying common trends where the processes are co-integrated.

---

[35] Because inserting $\boldsymbol{x}_{1t} = \lambda_1 w_t + e_{1t}$ and $\boldsymbol{x}_{2t} = \lambda_2 w_t + e_{2t}$ into $F = \beta_1 \boldsymbol{x}_{1t} + \beta_2 \boldsymbol{x}_{2t}$ results in $F = \beta_1 \boldsymbol{x}_{1t} + \beta_2 \boldsymbol{x}_{2t} = \beta_1 \lambda_1 w_t + e_{1t} + \beta_2 \lambda_2 w_t + e_{2t}$. To make F stationary, the I(1) process w_t needs to cancel out. This is the case if $\beta_1 \lambda_1 = -\beta_2 \lambda_2$ or equally if $\frac{\beta_2}{\beta_1} = -\frac{\lambda_1}{\lambda_2}$.

[36] As described in footnote 30, F is stationary for $\frac{\beta_2}{\beta_1} = -\frac{\lambda_1}{\lambda_2}$. With $\beta_1 = 1$, F is stationary for $\beta_2 = -\frac{\lambda_1}{\lambda_2}$, and therefore, the unique normalized co-integration relationship is $F = \boldsymbol{x}_{1t} - \frac{\lambda_2}{\lambda_1} \boldsymbol{x}_{2t}$.

[37] Because if

$x_{1t} = \lambda_1 w_{1t} + e_{1t} \sim I(1), \quad x_{2t} = \lambda_2 w_{2t} + e_{2t} \sim I(1),$

$w_{1t} = w_{1t-1} + e_{3t} \sim I(1) \quad w_{2t} = w_{2t-1} + e_{4t} \sim I(1),$

then, for the linear combination $F = \beta_1 x_{1t} + \beta_2 x_{2t} = \beta_1 \lambda_1 w_1 + e_{1t} + \beta_2 \lambda_2 w_{2t} + e_{2t}$, no combination of β_1 and β_2 exists, such that w_{1t} and w_{2t} cancels each other out, what would be necessary to make $F \sim I(0)$.

[38] If the processes and common trends are defined as

$x_{1t} = \lambda_{11} w_{1t} + \lambda_{12} w_{2t} + e_{1t} \sim I(1), \quad x_{2t} = \lambda_{21} w_{1t} + \lambda_{22} w_{2t} + e_{2t} \sim I(1),$

3)Three variables with one common trend: In the case that there are three processes and one common trend, the three processes are just multiples of the common shared trend

$$x_{1t} = \lambda_1 w_t + e_{1t} \sim I(1); \qquad x_{2t} = \lambda_2 w_t + e_{2t} \sim I(1) \tag{51a}$$

$$x_{3t} = \lambda_3 w_t + e_{3t} \sim I(1); \qquad w_t = w_{t-1} + e_{4t} \sim I(1) \tag{51b}$$

In such a case, each process can be displayed as a multiple of the other processes, for example

$$x_{1t} = \frac{\lambda_1}{\lambda_2} x_{2t} + \frac{1}{\lambda_2} e_{2t} + e_{1t},\ x_{2t} = \frac{\lambda_2}{\lambda_3} x_{3t} + \frac{1}{\lambda_3} e_{3t} + e_{2t} \text{ and } x_{3t} = \frac{\lambda_3}{\lambda_1} x_{1t} + \frac{1}{\lambda_1} e_{1t} + e_{3t} \tag{52}$$

Therefore, three normalized co-integration relationships exist, which are stationary I(0) processes[39]. But because one co-integration relationship can be calculated from the other two processes, there are only two *linearly independent* normalized co-integration relationships[40] (therefore $r = 2$). An example with three variables and one common trend is shown in Figure A.4.2 on page 71.

4) Three variables with two common trends: In general, a system with three variables and two common trends can be defined as

$$x_{1t} = \lambda_{11} w_{1t} + \lambda_{12} w_{2t} + e_{1t} \sim I(1) \tag{53a}$$

$$x_{2t} = \lambda_{21} w_{1t} + \lambda_{22} w_{2t} + e_{2t} \sim I(1) \tag{53b}$$

$$x_{3t} = \lambda_{31} w_{1t} + \lambda_{32} w_{2t} + e_{3t} \sim I(1) \tag{53c}$$

Where $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are random walks. As seen in (2), two variables and two common trends cannot be co-integrated, therefore, all three variables are needed to get a stationary linear combination. In such a case, the weights β_2 and β_3 for the processes $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ must be in this way, that w_1 and w_2 cancel out in the normalized linear combination $F = x_{1t} + \beta_2 x_{2t} + \beta_3 x_{3t}$. More precisely, the conditions $\lambda_{11} = \beta_2 \lambda_{21} + \beta_3 \lambda_{31}$ and $\lambda_{12} = \beta_2 22 + \beta_3 \lambda_{32}$ must hold. Because a common trend is only common if it is present in at least two processes, a system with three variables and two common trends must at least be defined as

$$x_{1t} = \lambda_{11} w_{1t} + \lambda_{12} w_{2t} + e_{1t} \sim I(1) \tag{54a}$$

$w_{1t} = w_{1t-1} + e_{3t} \sim I(1), \quad w_{2t} = w_{2t-1} + e_{4t} \sim I(1),$

then, there exists the case, where the trends are present with the same weights, if $\frac{\lambda_{11}}{\lambda_{12}} = \frac{\lambda_{21}}{\lambda_{22}}$. In this case, a linear combination *F* could be found, such that $F \sim I(0)$, because with $\frac{\lambda_{11}}{\lambda_{12}} = \frac{\lambda_{21}}{\lambda_{22}}$, the second process $\boldsymbol{x}_2$ can be displayed as $x_{2t} = \frac{\lambda_{22}}{\lambda_{12}} \lambda_{11} w_{1t} + \frac{\lambda_{21}}{\lambda_{11}} \lambda_{12} w_{2t} + e_{2t} = \frac{\lambda_{22}}{\lambda_{12}} (\lambda_{11} w_{1t} + \lambda_{21} w_{2t}) + e_{2t}$, and in $F = \beta_1 (\lambda_{11} w_{1t} + \lambda_{12} w_{2t}) + \beta_2 \frac{\lambda_{22}}{\lambda_{12}} (\lambda_{11} w_{1t} + \lambda_{12} w_{2t}) + e_{1t} + e_{2t}$ the two I(1) processes $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ would cancel out for $\beta_2 = -\frac{\lambda_{12}}{\lambda_{22}}$. But in this case, there would be only one true underlying common trend $\lambda_{11} w_{1t} + \lambda_{12} w_{2t}$.

[39] This would be $F_{12} = x_{1t} - \frac{\lambda_1}{\lambda_2} x_{2t} - \frac{\lambda_1}{\lambda_2} e_{2t} - e_{1t} \sim I(0)$, $F_{23} = x_{2t} - \frac{\lambda_2}{\lambda_3} x_{3t} - \frac{\lambda_2}{\lambda_3} e_{3t} - e_{2t} \sim I(0)$ *and* $F_{13} = x_{1t} - \frac{\lambda_1}{\lambda_3} x_{3t} - \frac{\lambda_1}{\lambda_3} e_{3t} - e_{1t} \sim I(0)$.

[40] For example putting $x_{2t} = \frac{\lambda_2}{\lambda_3} x_{3t} + \frac{\lambda_2}{\lambda_3} e_{3t} + e_{2t}$ into $F_{12} = x_{1t} - \frac{\lambda_1}{\lambda_2} x_{2t} - \frac{\lambda_1}{\lambda_2} e_{2t} - e_{1t}$ yields $F_{12} = x_{1t} - \frac{\lambda_1}{\lambda_2} \left(\frac{\lambda_2}{\lambda_3} x_{3t} + \frac{\lambda_2}{\lambda_3} e_{3t} + e_{2t} \right) - \frac{\lambda_1}{\lambda_2} e_{2t} - e_{1t} = x_{1t} - \frac{\lambda_1}{\lambda_3} x_{3t} - \frac{\lambda_1}{\lambda_3} e_{3t} - \frac{\lambda_1}{\lambda_2} e_{2t} - \frac{\lambda_1}{\lambda_2} e_{2t} - e_{1t} = x_{1t} - \frac{\lambda_1}{\lambda_3} x_{3t} - \frac{\lambda_1}{\lambda_3} e_{3t} - e_{1t} = F_{13}$.

$$x_{2t} = \lambda_2 w_{1t} + e_{2t} \sim I(1) \quad (54b)$$

$$x_{3t} = \lambda_3 w_{2t} + e_{3t} \sim I(1) \quad (54c)$$

In this case, β_2 and β_3 needs to be $\frac{\lambda_{11}}{\lambda_2}$ and $\frac{\lambda_{12}}{\lambda_3}$, so that the normalized co-integrations relationship is

$$F = x_{1t} - \frac{\lambda_{11}}{\lambda_2} x_{2t} + \frac{\lambda_{12}}{\lambda_3} x_{3t} + \frac{\lambda_{11}}{\lambda_2} e_{2t} + \frac{\lambda_{12}}{\lambda_3} e_{3t} + e_{1t} \sim I(0) \quad (55)$$

Figure A.4.3 on page 71 shows an example with three variables and two common trends.

Similar arguments hold for cases with more variables. The case with three variables and three common trends is also just possible in special cases (in which, similar to (2)), there would be less than three *true* common trends). In general, $k - r = m$ holds, where k is the number of variables, r is the number of co-integration relationships and m is the number of true common trends (Gruber, 2011, p. 66ff).

B) Mathematical formulations

A formal definition of co-integration is as follows: k univariate processes $\mathbf{x}_1, \ldots, \mathbf{x}_k$ are co-integrated of order (d,b), if all processes are integrated of the same order d and a linear combination $\boldsymbol{z}$ of the processes exists, where $\boldsymbol{z}$ is of order (d-b), so that

$$x_{1t} \sim I(d), x_{2t} \sim I(d), \ldots, x_{kt} \sim I(d); \qquad z_t = \beta_1 x_{1t} + \beta_2 x_{2t} + \ldots + \beta_k x_{kt} \sim I(d-b) \quad (56)$$

The co-integration relationship $\boldsymbol{z} = \boldsymbol{\beta}' \boldsymbol{x}_t$ is then a univariate process with a smaller order of integration than the k processes $\mathbf{x}_1, \ldots, \mathbf{x}_k$. $\boldsymbol{\beta}$ is the so-called co-integrations vector. For the most common forms, where $\boldsymbol{x}_i \sim I(1)\ \forall\ i = 1, \ldots, k$, $\boldsymbol{\beta}$ transforms k integrated processes of order one into a stationary process $\boldsymbol{z}$. Thereby, the co-integration relationship $\beta_1 x_{1t} + \beta_2 x_{2t} + \ldots + \beta_k x_{kt} \sim I(d-b)$ is not unambiguous, because every multiple of the relationship will also be co-integrated, so that if $\beta_1 x_{1t} + \beta_2 x_{2t} + \cdots + \beta_k x_{kt} \sim I(d-b)$, then also $\lambda(\beta_1 x_{1t} + \beta_2 x_{2t} + \cdots + \beta_k x_{kt}) \sim I(d-b)$, so that in fact there are infinite linear dependent co-integration relationships. Therefore, it is necessary to normalize $\boldsymbol{\beta}$ to $\boldsymbol{\beta}^* = (1, \beta_2, \ldots, \beta_k)$ to make the infinite co-integration relationships unambiguous. But $\boldsymbol{z}$ does not need to be a univariate process in the case that there exist more than one linear independent co-integration relationship. In this case, $\boldsymbol{\beta}_1$ to $\boldsymbol{\beta}_k$ will be vectors and the r linear independent co-integration relationships can be written as

$$\boldsymbol{B x_t} = \begin{pmatrix} \beta_{11} & \cdots & \beta_{1k} \\ \vdots & \ddots & \vdots \\ \beta_{r1} & \cdots & \beta_{rk} \end{pmatrix} \begin{pmatrix} x_{1t} \\ \vdots \\ x_{kt} \end{pmatrix} = \begin{pmatrix} \beta_{11} x_{1t} + \beta_{12} x_{2t} + \ldots + \beta_{1K} x_{kt} \\ \cdots \\ \beta_{r1} x_{1t} + \beta_{r2} x_{2t} + \ldots + \beta_{rK} x_{kt} \end{pmatrix} \quad (57)$$

Like in the case of a univariate co-integration relationship, the r linear independent co-integration relationships are also not unambiguous, because every multiple of a relationship is again co-integrated, so that if $\boldsymbol{B x_t} \sim I(d-b)$, then also $\boldsymbol{\lambda B x_t} \sim I(d-b)$, so that

$$\boldsymbol{\lambda B x_t} = \begin{pmatrix} \lambda_1(\beta_{11} x_{1t} + \beta_{12} x_{2t} + \ldots + \beta_{1k} x_{kt}) \\ \cdots \\ \lambda_r(\beta_{r1} x_{1t} + \beta_{r2} x_{2t} + \ldots + \beta_{rk} x_{kt}) \end{pmatrix} \qquad \text{with } \boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \lambda_r \end{bmatrix} \quad (58)$$

Again, normalization can make the r co-integration relationships unambiguous. Because the co-integration rank r can be between 1 and k-1, the r normalized co-integration relationships are then calculated as a linear combination of the processes $\boldsymbol{x}_{k-r}, \boldsymbol{x}_{k-r+1}, \dots, \boldsymbol{x}_k$ and one of the processes $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_{k-r-1}$ such that $\boldsymbol{B}^* = (\boldsymbol{I}, \boldsymbol{B}_{k-r})$ with

$$\boldsymbol{B}^* = \begin{pmatrix} 1 & \cdots & 0 & \beta^*_{1(k-r)} & \cdots & \beta^*_{1k} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \beta^*_{r(k-r)} & \cdots & \beta^*_{rk} \end{pmatrix} \tag{59}$$

(Gruber, 2011, p.69ff). A VAR(p) process can be transformed into its VEC model form by a recursive process illustrated as follows on a VAR(3), defined as $\boldsymbol{x}_t = \boldsymbol{\Phi}_1 \boldsymbol{x}_{t-1} + \boldsymbol{\Phi}_2 \boldsymbol{x}_{t-2} + \boldsymbol{\Phi}_3 \boldsymbol{x}_{t-3} + \boldsymbol{e}_t$. Adding and subtracting $\boldsymbol{\Phi}_3 \boldsymbol{x}_{t-2}$ to the right side gives

$$\boldsymbol{x}_t = \boldsymbol{\Phi}_1 \boldsymbol{x}_{t-1} + \boldsymbol{\Phi}_2 \boldsymbol{x}_{t-2} + \boldsymbol{\Phi}_3 \boldsymbol{x}_{t-2} - \boldsymbol{\Phi}_3 \boldsymbol{x}_{t-2} + \boldsymbol{\Phi}_3 \boldsymbol{x}_{t-3} + \boldsymbol{e}_t \tag{60a}$$

$$\boldsymbol{x}_t = \boldsymbol{\Phi}_1 \boldsymbol{x}_{t-1} + (\boldsymbol{\Phi}_2 + \boldsymbol{\Phi}_3) \boldsymbol{x}_{t-2} + \boldsymbol{\Phi}_3 \Delta \boldsymbol{x}_{t-2} + \boldsymbol{e}_t \tag{60b}$$

Further adding and subtracting $(\boldsymbol{\Phi}_2 + \boldsymbol{\Phi}_3) \boldsymbol{x}_{t-1}$ to the right side gives

$$\boldsymbol{x}_t = \boldsymbol{\Phi}_1 \boldsymbol{x}_{t-1} + (\boldsymbol{\Phi}_2 + \boldsymbol{\Phi}_3) \boldsymbol{x}_{t-1} - (\boldsymbol{\Phi}_2 + \boldsymbol{\Phi}_3) \boldsymbol{x}_{t-1} + \dots$$
$$\dots (\boldsymbol{\Phi}_2 + \boldsymbol{\Phi}_3) \boldsymbol{x}_{t-2} + \boldsymbol{\Phi}_3 \Delta \boldsymbol{x}_{t-2} + \boldsymbol{e}_t \tag{61a}$$

$$\boldsymbol{x}_t = (\boldsymbol{\Phi}_1 + \boldsymbol{\Phi}_2 + \boldsymbol{\Phi}_3) \boldsymbol{x}_{t-1} - (\boldsymbol{\Phi}_2 + \boldsymbol{\Phi}_3) \Delta \boldsymbol{x}_{t-1} + \boldsymbol{\Phi}_3 \Delta \mathbf{x}_{t-2} + \mathbf{e}_t \tag{61b}$$

Subtracting $\mathbf{x}_{t-1}$ finally gives

$$\boldsymbol{x}_t - \boldsymbol{x}_{t-1} = (\boldsymbol{\Phi}_1 + \boldsymbol{\Phi}_2 + \boldsymbol{\Phi}_3) \boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1} - (\boldsymbol{\Phi}_2 + \boldsymbol{\Phi}_3) \Delta \boldsymbol{x}_{t-1} + \boldsymbol{\Phi}_3 \Delta \boldsymbol{x}_{t-2} + \boldsymbol{e}_t \tag{62a}$$

$$\Delta \boldsymbol{x}_t = (\boldsymbol{\Phi}_1 + \boldsymbol{\Phi}_2 + \boldsymbol{\Phi}_3 - \boldsymbol{I}) \boldsymbol{x}_{t-1} - (\boldsymbol{\Phi}_2 + \boldsymbol{\Phi}_3) \Delta \boldsymbol{x}_{t-1} + \boldsymbol{\Phi}_3 \Delta \boldsymbol{x}_{t-2} + \boldsymbol{e}_t \tag{62b}$$

This equation can be relabeled as $\Delta \boldsymbol{x}_t = \boldsymbol{\Pi} \boldsymbol{x}_{t-1} - \boldsymbol{\Pi}_1 \Delta \boldsymbol{x}_{t-1} + \boldsymbol{\Pi}_2 \Delta \boldsymbol{x}_{t-2} + \boldsymbol{e}_t$, so that the general VEC model for a VAR(p) is defined as

$$\Delta \boldsymbol{x}_t = \boldsymbol{c} + \boldsymbol{\Pi} \boldsymbol{x}_{t-1} + \boldsymbol{\Pi}_1 \Delta \boldsymbol{x}_{t-1} + \dots + \boldsymbol{\Pi}_{p-1} \Delta \boldsymbol{x}_{t-p+1} + \boldsymbol{e}_t \tag{63}$$

with $\boldsymbol{\Pi} = \boldsymbol{A}\boldsymbol{B}'$, where $\boldsymbol{A}$ contains the speed of adjustment from deviations of the co-integration relationship and $\boldsymbol{B}$ contains the co-integrating vectors, and therefore $\boldsymbol{\Pi} = -\boldsymbol{I}_k + \sum_{i=1}^{p} \boldsymbol{\Phi}_i$ and $\boldsymbol{\Pi}_j = \sum_{i=j+1}^{p} \boldsymbol{\Phi}_i$ [41] (Sheppard, 2010, p. 343ff). For most cases, where the processes are at most I(1) processes, the left hand side of (59) is stationary. On the right side of (63), the stationary lagged differences as well as the error-correction term $\boldsymbol{\Pi} \boldsymbol{x}_{t-1}$ appears and therefore $\boldsymbol{\Pi} \boldsymbol{x}_{t-1}$ must also be stationary in order to balance the VEC model. Thereby *three* cases considering the rank of $\boldsymbol{\Pi}$ (short $(rk(\boldsymbol{\Pi}))$ can be distinguished – because $rk(\boldsymbol{\Pi})$ lies between zero and k such that $0 \leq rk(\boldsymbol{\Pi}) \leq k$. 1). If $rk(\boldsymbol{\Pi}) = 0$, then no cointegration relationship exists and $\boldsymbol{\Pi} \boldsymbol{x}_{t-1}$ vanishes from (63) making it stationary ((63) then

[41] Equation (63) is called the transitory form as opposed to the long-run form $\Delta \mathbf{x}_t = \mathrm{c} + \boldsymbol{\Pi} \mathbf{x}_{t-p} - \Pi_1 \Delta \mathbf{x}_{t-1} + \dots + \Pi_{p-1} \Delta \mathbf{x}_{t-p+1} + \mathbf{e}_t$, where the levels of the process now enter lagged by p periods instead of one and the coefficients Π_j, defined as $\Pi_j = -(I - \boldsymbol{\Phi}_1 - \dots - \boldsymbol{\Phi}_j)$, for $j = 1 \dots p$, now represent cumulative long-run impacts instead of transitory effects.

would be a VAR in differences). 2) If $rk(\boldsymbol{\Pi}) = k$, then there are k independent relationships, which are all stationary, but these relationships are no co-integration relationships, because $rk(\boldsymbol{\Pi}) = k$ can only be the case when all processes are already stationary and not integrated (no co-integration relationship is possible without integrated processes). 3) For the case that the matrix $\boldsymbol{\Pi}$ does not have full rank, i.e. $0 < rk(\boldsymbol{\Pi}) < k$, two matrices $\boldsymbol{A}_{kxr}$ and $\boldsymbol{B}_{kxr}$ exist so that $\boldsymbol{\Pi} = \boldsymbol{AB}$`. Because $\boldsymbol{AB}`\boldsymbol{x}_{t-1}$ is stationary, also the r linear independent co-integration relationships $\boldsymbol{B'x}_{t-1}$ are stationary. Therefore, $rk(\boldsymbol{\Pi})$ is equal to the cointegration rank of the system. Then, a VEC model with co-integration rank r, $0 < r < k$, can be interpreted as follows: The matrices $\boldsymbol{\Pi}_j$ are the weights of the autoregressive structure of the differenced process $\Delta\boldsymbol{x}_t$ (just like in an ordinary VAR(p) model in differences), the matrix $\boldsymbol{B}$ describes the long-run co-integration relationships whereby $\boldsymbol{\alpha}$ describes the speed of adjustment of the processes to the long-run equilibrium ($\boldsymbol{A}$ is called the loading matrix) (Pfaff et al., 2008, p. 77ff).

C) Johansen test for co-integration

The Johansen test for co-integration can be applied to any number of suggested co-integration relationships. The test is based on a maximum likelihood estimation of the residuals of the VEC model

$$\boldsymbol{e}_t = \Delta\boldsymbol{x}_t - \boldsymbol{AB}`\boldsymbol{x}_{t-1} - \boldsymbol{\Pi}_1\Delta\boldsymbol{x}_{t-1} - \ldots - \boldsymbol{\Pi}_{p-1}\Delta\boldsymbol{x}_{t-p+1} \quad (64)$$

where (64) is obtained from (63) by setting $\boldsymbol{c}$ to 0 and solving for $\boldsymbol{e}_t$. The likelihood function of $\boldsymbol{e}_t$ is defined as

$$L(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Pi}_1, \ldots, \boldsymbol{\Pi}_{p-1}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{TK}{2}}(det\,\boldsymbol{\Sigma})^{-\frac{T}{2}}\, exp(-\tfrac{1}{2}\textstyle\sum_{t=1}^{T} \boldsymbol{e}_t'\boldsymbol{\Sigma}^{-1}\,\boldsymbol{e}_t) \quad (65)$$

with $\boldsymbol{\Sigma}$ as the variance-covariance matrix of $\boldsymbol{e}_t$. The likelihood function depends on the parameters $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Pi}_1, \ldots, \boldsymbol{\Pi}_{p-1}$ and $\boldsymbol{\Sigma}$, whereby $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Pi}_1, \ldots, \boldsymbol{\Pi}_{p-1}$ are included in $\boldsymbol{e}_t$ via (64).

The test can be described in ***three steps:*** *1)* correction for lagged influences, *2)* estimation of the co-integration matrix $\boldsymbol{B}$ and *3)* estimation of the rank of co-integration (thereby determining $\boldsymbol{B}$).

1) Correction for lagged influences: The aim of the first step is to obtain that part of the information in $\boldsymbol{x}_{t-1}$, which is relevant for determining $\Delta\boldsymbol{x}_t$, i.e. to check, whether there is information in $\boldsymbol{x}_{t-1}$ (co-integration relationships) which helps explaining $\Delta\boldsymbol{x}_t$. Thereby, both $\boldsymbol{x}_{t-1}$ and $\Delta\boldsymbol{x}_t$ are overlaid by influences of the lagged differences $\Delta\boldsymbol{x}_{t-1}, \ldots, \Delta\boldsymbol{x}_{t-p+1}$. Therefore, the information in $\boldsymbol{x}_{t-1}$ and $\Delta\boldsymbol{x}_t$, which is not due to $\Delta\boldsymbol{x}_{t-1}, \ldots, \Delta\boldsymbol{x}_{t-p+1}$, is extracted (contained in the residuals $\boldsymbol{u}_{\Delta,t}$ and $\boldsymbol{u}_{B,t}$) via the regressions

$$\Delta\boldsymbol{x}_t = \boldsymbol{\Upsilon}_1\Delta\boldsymbol{x}_{t-1} + \ldots + \boldsymbol{\Upsilon}_{p-1}\Delta\boldsymbol{x}_{t-p+1} + \boldsymbol{u}_{\Delta,t} \quad (66a)$$

$$\boldsymbol{Bx}_t = \boldsymbol{x}_{t-1} = \boldsymbol{\Lambda}_1\Delta\boldsymbol{x}_{t-1} + \ldots + \boldsymbol{\Lambda}_{p-1}\Delta\boldsymbol{x}_{t-p+1} + \boldsymbol{u}_{B,t} \quad (66b)$$

Then the remaining influences of $\boldsymbol{x}_t$ on $\Delta\boldsymbol{x}_t$ (which is stored in the residuals of the regressions (66a) and (66b)) can be estimated with the regression $\boldsymbol{u}_{\Delta,t} = \boldsymbol{AB}'\boldsymbol{u}_{B,t} + \boldsymbol{e}_t$.

2) Estimation of the co-integration matrix $\boldsymbol{B}$*:* The correction for lagged influences already maximizes the likelihood function (65) for $\boldsymbol{\Pi_1, \dots, \Pi_{p-1}}$. Therefore, it only depends on $\boldsymbol{A, B}$ and $\boldsymbol{\Sigma}$. Combining (64) with (65) then leads to

$$L(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{TK}{2}} (det\, \boldsymbol{\Sigma})^{-\frac{T}{2}} \dots$$
$$\dots exp(-\tfrac{1}{2}\textstyle\sum_{t=1}^{T} \left(\boldsymbol{u}_{\Delta,t} - \boldsymbol{A}\boldsymbol{B}'\boldsymbol{u}_{B,t}\right)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{u}_{\Delta,t} - \boldsymbol{A}\boldsymbol{B}'\boldsymbol{u}_{B,t})) \quad (67)$$

If $\boldsymbol{B}$ was known, then $\boldsymbol{A}$ and $\boldsymbol{\Sigma}$ could be calculated out of $\boldsymbol{B}$ with

$$\widehat{\boldsymbol{A}}(\boldsymbol{B}) = -\boldsymbol{S}_{01}\boldsymbol{B}(\boldsymbol{B}'\boldsymbol{S}_{11}\boldsymbol{B})^{-1} \quad (68a)$$

$$\widehat{\boldsymbol{\Sigma}}(\boldsymbol{B}) = \boldsymbol{S}_{00} - \boldsymbol{S}_{01}\boldsymbol{B}(\boldsymbol{B}'\boldsymbol{S}_{11}\boldsymbol{B})^{-1}\boldsymbol{B}'\boldsymbol{S}_{10} \quad (68b)$$

whereby

$$\boldsymbol{S}_{00} = \tfrac{1}{T}\textstyle\sum_{t=1}^{T} \boldsymbol{R}_{\Delta,t}\boldsymbol{R}'_{\Delta,t} \qquad \boldsymbol{S}_{11} = \tfrac{1}{T}\textstyle\sum_{t=1}^{T} \boldsymbol{R}_{B,t}\boldsymbol{R}'_{B,t} \quad (69a)$$

$$\boldsymbol{S}_{01} = \tfrac{1}{T}\textstyle\sum_{t=1}^{T} \boldsymbol{R}_{\Delta,t}\boldsymbol{R}'_{B,t} \qquad \boldsymbol{S}_{11} = \tfrac{1}{T}\textstyle\sum_{t=1}^{T} \boldsymbol{R}_{B,t}\boldsymbol{R}'_{B,t} \quad (69b)$$

When (68a) and (68b) are applied to (67), then (67) simplifies to

$$L(\boldsymbol{B}) = (2\pi)^{-\frac{TK}{2}} \left(det\, \widehat{\boldsymbol{\Sigma}}(\boldsymbol{B})\right)^{-\frac{T}{2}} exp(\tfrac{TK}{2}) \quad (70)$$

because $-\frac{1}{2}\sum_{t=1}^{T} \left(\boldsymbol{u}_{\Delta,t} - \widehat{\boldsymbol{A}}(\boldsymbol{B})\boldsymbol{B}'\boldsymbol{u}_{B,t}\right)'\widehat{\boldsymbol{\Sigma}}(\boldsymbol{B})^{-1}(\boldsymbol{u}_{\Delta,t} - \widehat{\boldsymbol{A}}(\boldsymbol{B})\boldsymbol{B}'\boldsymbol{u}_{B,t}) = \frac{TK}{2}$. Maximizing (70) then is equal to minimizing the determinant $\det \widehat{\boldsymbol{\Sigma}}(\boldsymbol{B})$. The minimum of this determinant is given by

$$det_{min}\, \widehat{\boldsymbol{\Sigma}}(\boldsymbol{B}) = det(\boldsymbol{S}_{00}) \textstyle\prod_{i=1}^{K}(1 - \lambda_i) \quad (71)$$

where λ_i are the canonical correlations[42] between $\boldsymbol{u}_{\Delta,t}$ and $\boldsymbol{u}_{B,t}$. These canonical correlations then are sorted into descending order, so that $\lambda_1 > \lambda_2 > \dots > \lambda_n$.

3) Estimation of the rank of co-integration: Testing the rank of co-integration is similar to testing how many canonical correlations are different from zero. This is tested in an iterative process, where the first null hypothesis is that $rank(\boldsymbol{AB}') = 0$, which is tested against the alternative hypothesis that $rank(\boldsymbol{AB}') = 1$. If the rank is zero, i.e. if the greatest canonical correlation λ_1 is zero, then there are no co-integration relationships (the testing procedure would be finished then). If λ_1 is not zero, then there is at least one co-integration relationship. To test, whether there might be more co-integration relationships, it is necessary to continue the procedure by testing whether the second largest canonical correlation λ_2 differs significantly from zero (if true, the test is done, if not, the test continues analogously). Testing whether the canonical correlations are zero is possible by applying (71) to (70), which leads to

$$L(\boldsymbol{B}) = (2\pi)^{-\frac{TK}{2}} (det(\boldsymbol{S}_{00}) \textstyle\prod_{i=1}^{K}(1 - \lambda_i))^{-\frac{T}{2}} exp(\tfrac{TK}{2}) \quad (72)$$

[42] See Gruber (2011) p. 132f for a description for canonical correlations.

Taking logarithms and reordering then yields

$$L(\boldsymbol{B}) = \frac{TK}{2} - \frac{TK}{2} ln(2\pi) - \frac{T}{2} det\, \boldsymbol{S}_{00} - \frac{T}{2} \sum_{i=1}^{K} ln(1 - \lambda_i) \quad (73)$$

The likelihood function then only depends on the rank of co-integration. (73) can now be used to implement a likelihood ratio test, for example to test if $rank(\boldsymbol{AB}') = r + 1$ versus the alternative hypothesis that $rank(\boldsymbol{AB}') = r$ by taking differences of (73) so that (for the case of the Johansen-max-test)

$$L_{r+1}(\boldsymbol{B}) - L_r(\boldsymbol{B}) = -\frac{T}{2} ln(1 - \lambda_{r+1}) \quad (74)$$

This is because the constant terms $\frac{TK}{2} - \frac{TK}{2} ln(2\pi) - \frac{T}{2} det\, \boldsymbol{S}_{00}$, as well as $\lambda_1, \dots, \lambda_r$ cancel out. Because $ln(1) = 0$, (74) becomes zero, when $\lambda_{r+1} = 0$ (Gruber, 2011, p. 127ff).

2.2. Machine Learning techniques

After first describing some basic ML concepts in part 2.2.1, three common ML techniques are presented in part 2.2.2.

2.2.1. Basic Machine Learning concepts

The following section will give a short overview of ML algorithms by briefly describing ***three dimensions*** according to which they can be categorized, namely ***A)*** methods, ***B)*** tasks and ***C)*** models.

A) ML methods – supervised vs unsupervised

In cases where the independent variable is not part of the data set (called an unlabeled data set), only unsupervised learning algorithms can be applied, whereas when the independent variable is part of the data set, then supervised learning is possible (the data set can be called labeled). A labeled data set is often split into a training and a testing data set. Supervised learning uses the labels (the response variable) to train the model on the training data. In this training process the abstraction of the learning process mentioned in the introduction takes place, so that a model is automatically fitted to the training data. In order to test the quality of generalization of the fitted model, the testing data set is used to test how the model works on before unseen data (data which have not been part of the model fitting process). In cases where the response variable is not known, a manual labelling of the data set is necessary to apply supervised learning algorithms. This, however, can be labor-intensive and time-consuming and is often not applicable for big data sets. In such a case learning is nevertheless possible using unsupervised learning algorithms. Examples of unsupervised learning methods would be clustering (clustering data into different groups without prior information about the groups) or association rules (finding specific rules that can describe relationships in the data).

B) ML tasks – classification vs regression

A very common task in ML is the case in which the data needs to be classified into two different classes (called binary classification). A typical example of a binary classification task is to distinguish if a given email should be assigned to the category "spam" or "non-spam". Sometimes, however, there might be the case where there are more than two categories. In such a case the task is a multi-class classification. Classifying news articles into domains like "sport", "business" or "politics" can be regarded as an example for a multi-class classification task. In many cases it might however even be more desired to predict a real number instead of discrete classes. Such tasks are called regression tasks. An example could be the prediction of a housing price given some features like "the square meters of the living space", "the number of bed rooms" or "the crime rate of the district".

C) ML model – probabilistic vs geometric vs logical

Another helpful dimension to think about ML algorithms is the type of model they use (either probabilistic, geometric or logical). Geometric models use concepts such as lines, planes and distances in high dimensional Cartesian instance spaces. Even when high dimensional, these models can provide intuitive explanations. Examples for geometric models are k-Nearest-Neighbor, k-Means or Support Vector Machines (SVMs). Probabilistic models use concepts like probability distributions, conditional probabilities or joint probabilities, often based on the Bayes´ rule. Examples for probabilistic models are Naïve Bayes Models or Gaussian Mixture Models. Logical models use rules, such as if $feature = A, then\ class = 1$ and if $feature = B, then\ class = -1$. Such rules can be stacked together to form complex models. Logical models, to some extent, can provide intuitions or even good explanations of the observed phenomena, because most rules are often understandable and reasonable for individuals. Examples of logical models are Decision Trees or Association Ruled Models (Flach, 2012, p. 21-48).

2.2.2. Applied techniques

In the following section 2.2.2, three widely used ML techniques, namely Naïve Bayes Classifiers, Support Vector Machines (SVMs) and Neural Networks (NNs) are presented.

2.2.2.1. Naive Bayes Classifiers

The following algorithm can be completely called a Multinomial Naïve Bayes Classifier. The term "Multinomial" thereby describes the term document model, while the terms "Naïve Bayes" describe the process of the classifier. The application of the Multinomial Naïve Bayes Classifier can be described in ***two steps: 1)*** first the term document model is depicted followed by ***2)*** the description of the classification process.

1) Document model – Multinomial: A term document model translates words into a matrix of numbers, so that subsequent calculations become possible. The multinomial term document model thereby is based

on a bag of words representation, meaning, that the words are simply taken out of the document without regarding the context or the syntax of the words. Before the words of a document are translated into a matrix, usually, some preprocessing is applied to the document (so that ordinarily not all words are chosen from the document[43]). The term "document" here refers to the whole document, such that a document contains several texts (texts like for example e-mails, messages or news announcements). All words chosen from a document are then translated into a matrix, whereby the rows refer to the words of the document (so that each distinct word represents one row, which means that equal words are only represented once, i.e. in one row) and the columns refer to the different texts (so that the number of columns is equal to the number of texts in the document). The multinomial term document matrix then displays the amounts of words in different texts, so that the entry of the matrix in column t and row i indicates the amount of word i occurring in text t. The resulting matrix, which is often called a sparse matrix (because most of its entries are zeros), then can be used for further calculations, so that classifiers can now be applied to the matrix.

2) Process of classification – Naïve Bayes: Naïve Bayes classifiers are linear classifiers which are simple to implement, but have been found to perform surprisingly well in different ML tasks, for example in e-mail spam classification. The word "naïve" in the name Naïve Bayes Classifier comes from the somehow naïve assumption that all attributes of one example are independent from each other (given the context of a class). In the case of news announcements, it would be assumed, that all words in the announcement are independent from each other - given a specific classification. For example, in the news announcement: "United States manufacturing PMI came in at 48.2, above forecasts (47)", which is classified as positive for the USD/GBP exchange rate (because the Purchasing Manager Index PMI was higher than expected), it would be assumed, that the word "manufacturing" is independent from all other words in the announcement (so, for example, from the word "above"). For such short term news announcements, the naïve assumption of independent words may not be a big problem. This assumption is more problematic for other text documents, like for example spam e-mails. In a spam e-mail, the assumption, that the word "offer" occurs independent form – let us say - the word "buy" may not be too realistic, because in a spam e-mail, one would often find sentences like: "This is a special offer, buy it now". So it would be more realistic to assume, that if the spam email contains the word "offer", then, the probability that the email also contains words like "buy", "discount" or "cheap" will be much higher. As mentioned above, in the case of short text news announcements, the assumption of independent words may not be a problem in most of the announcements.

[43] Very common words like "or", "a" or "the" etc. (called stopwords) and also additionally numbers and punctuations are often removed from the document. Also the words could be stemmed to the roots of the words in hope to treat similar words with the same root as equal words.

The Naïve Bayes classifier is in the following explained by the more intuitive task of email classification as spam (S) or non-spam (NS). Considering an email (E), whose class is represented by C (with two possible classes $C = S$ for spam or $C = NS$ for non-spam). Then, the Naïve Bayes Classifier classifies an email as the class with the highest so called posterior probability $P(C|E)$. So, if the probability that the given email belongs to class S, which is $P(C = S|E)$, is greater than the probability that the given email belongs to class NS, which is $P(C = NS|E)$, then the email will be classified as spam. The probability of belonging to either of the classes can be expressed using Bayes´ Theorem as

$$P(C|E) = \frac{P(E|C)\,P(C)}{P(E)} \tag{75}$$

Therefore, to classify a new email, it is necessary know if $P(C = S|E) > P(C = NS|E)$, which is then equivalent to

$$P(E|C = S)P(C = S) > P(E|C = NS)\,P(C = NS) \tag{76}$$

(because $P(E)$ is cancelling out). In order to classify an email as spam or non-spam, ***two terms*** are necessary: ***a)*** P(E|C) and ***b)*** $P(C)$ (for both classes). ***a)*** In order to calculate $P(E|C = S)$, it is necessary to first get the likelihoods for the individual words w_j belonging to class $C = S$, which is $P(w_j|C = S)$, with $j = 1, \dots, m$ and m as the number of all distinct words in the term document matrix. By making the Naïve Bayes assumption, it is possible to write the probability that an email is belonging to class S in terms of the individual likelihoods of all words as

$$P(E|C = S) = \prod_{j=1}^{m} P(w_j|C = S) \tag{77}$$

The terms $P(w_j|C = S)$ can be obtained by using the training data set of labeled emails. Let thereby $n_S(w_j)$ be the frequency of word w_j occurring in emails of class $C = S$ and let N_S be the total number of words of emails of class $C = S$ in the training data set. Then, it is possible to estimate the likelihood of an individual word w_j belonging to class $C = S$ with

$$\hat{P}(w_j|C = S) = \frac{1 + n_S(w_j)}{m + N_S} \tag{78}$$

as the normed relative frequency of word w_j occurring in emails of class $C = S$. The intuitive relative frequency $\frac{n_S(w_j)}{N_S}$, which is the frequency of word w_j occurring in emails of class S is thereby normed with 1 in the nominator and m in the denominator. The one in the nominator ensures that the nominator will not be zero in cases where a word did not occur in any email of class S ($n_S(w_j) = 0$ in this case). Without the one in the nominator, $\hat{P}(w_j|C = S)$ would be zero and because of the multiplication in (77), $P(E|C = S)$ would also be (erroneously) zero. The m term in the denominator in (78) then compensates for the ones in the nominator to normalize $\hat{P}(w_j|C = S)$. After computing (78) for each word it is possible to multiply them, as in (77), over all m distinct words m to receive $P(E|C = S)$.

b) After computing $P(E|C = S)$ it is also necessary to compute $P(C = S)$ to verify (76). The probability $P(C = S)$, which is the probability that an email will be of class $C = S$, is again estimated by the training data set as the proportion of the spam emails to all emails in the training data set

$$\hat{P}(C = S) = \frac{N_S}{N} \tag{79}$$

where N_S is the total number of emails of class $C = S$ and N is the total number of all emails in the training data set.

With the estimations of $\hat{P}(E|C = S)$ and $\hat{P}(C = S)$, the left side of (76) is solved. To receive the right hand side of (76), ***a)*** and ***b)*** are analogously repeated for the negative class $C = NS$. With verifying (76) a new email can finally be classified as either spam or non-spam (Lantz, 2013, p. 97-123 & Shimodaira, 2015, p. 1-9). Appendix B.I provides a full example of the multinomial term document model and the Naïve Bayes Classifier applied to five fictive emails.

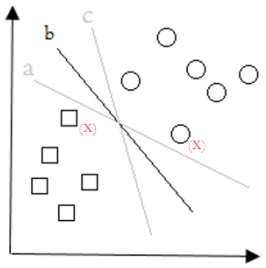
2.2.2.2. Support Vector Machines

To illustrate the concepts of SVMs ***three cases*** are distinguished in the following. In part ***A)*** the case of a linear separable data set with a linear class boundary, in part ***B)*** a not linearly separable data set with a linear class boundary and in part ***C)*** a not linearly separable data set with a non-linear class boundary is described.

A) The linear separable case (with linear class boundary) – the Maximal Margin Classifier

The main idea of a SVM is, to use n-dimensional surfaces to define the relationship between the features and the response. A precondition thereof is, that the data are linearly separable. In an n-dimensional space, spanned by the n features, this surface (also called a hyperplane) defines a boundary with the aim of best separating the data into similar groups according to their response. SVMs can be used for both classification and regression tasks, but they are for the beginning best understood when the idea is represented in a two-dimensional feature space with the task of classifying data into two different groups. The two different groups are represented in the right figure by squares and circles. The task of the SVM is to identify that hyperplane which separates the two classes (a hyperplane in a two-dimensional space is just a line). As illustrated in Figure 2 with a,b and c, there are more lines which separate the two classes. So which one is the best separator? One way to answer this question is to search that line which creates the greatest separation between the classes, because such a separator is more likely to generalize best to future data. In order to illustrate this, a square and a circle are marked with a red (x) in the upper figure. If future observations are similar to this points, but vary just a little bit in the features, then the lines a and c are more likely to classify the data wrongly (if the square would move up, or the circle would move down a

Figure 2: Three different decision boundaries

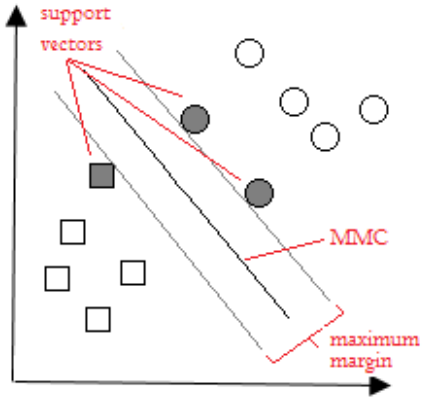


Source: Own illustration

bit). The separator, which will generalize best to future data is such, that it maximizes a margin between the classes – for this reason, this separator is called Maximum Margin Hyperplane (MMH). In order to maximize the margin between the classes, the MMH classifier uses so-called support vectors. Support vectors are those observations which are closest to the MMH. In a two dimensional space, these observations are just points, representing two features (as illustrated in Figure 3 on the left). But in a multidimensional space, these observations are vectors, representing multiple features (that is why they are called support vectors). A key property of a SVM is, that only the support vectors are necessary to define the MMH (in this sense, only the support vectors are supporting the MMH). With this property, SVMs can store classification models in a very compact way (Lantz, 2013, p. 225ff).

The illustrations mentioned so far can now be extended to p dimensions (while still applying the concepts to a binary classification task using linearly separable data). As illustrated, a hyperplane divides a p-dimensional space into two subspaces. A hyperplane in a p-dimensional space is a flat affine subspace of dimension p-1, which is defined as

Figure 3: Support vectors and MMH



Source: Own illustration

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p = 0 \tag{80}$$

where every p-dimensional observation $\boldsymbol{x}_t = (x_{t1} + x_{t2} + \ldots + x_{tp})$ which satisfies the equation lies on the hyperplane. At the same time, this means that every observation which does not satisfy (80) will lie on one side of the hyperplane (also called a subspace). Calculating the sign of the left hand side of the equation defines on which side on the hyperplane the observation $\boldsymbol{x}_t$ lies. In the case that the two classes are labeled with 1 and -1, so that the response $\boldsymbol{y} = y_1, y_2, \ldots, y_T \in \{-1, 1\}$, a separating hyperplane must fulfill

$$\beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \ldots + \beta_p x_{tp} > 0 \quad \text{if} \quad y_t = 1 \tag{81a}$$

$$\beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \ldots + \beta_p x_{tp} < 0 \quad \text{if} \quad y_t = -1 \tag{81b}$$

for all $t = 1, \ldots, T$. Equations (81a) and (81b) can simultaneously be written as

$$y_t(\beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \ldots + \beta_p x_{tp}) > 0 \tag{82}$$

for all $t = 1, \ldots, T$. With these definitions, an observation $\boldsymbol{x}_t$ is assigned to class 1 if $\beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \ldots + \beta_p x_{tp}$ is positive and $\boldsymbol{x}_t$ is assigned to class -1 if $\beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \ldots + \beta_p x_{tp}$ is negative. In the first case, the multiplication with the class label 1 does not matter, and in the latter case the multiplication with the class label -1 leads to a positive number (so for both classes, (82) results in a positive number). At the same time this means that if $y_t(\beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \ldots + \beta_p x_{tp})$ is far bigger than zero, then the observation $\boldsymbol{x}_t$ is also far away from the separating hyperplane (which means

that such an observation could be classified with more confidence). For observations of both classes, the distance or margin (M) from the hyperplane can thus be defined as $M = y_t(\beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \ldots + \beta_p x_{tp})$. To find the hyperplane with the greatest separation (out of the infinite number of possible separating hyperplanes) and to maximize the margin, the MMH is defined as

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{maximize } M} \tag{83a}$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1 \text{ and} \tag{83b}$$

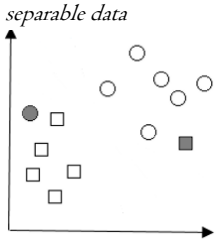
$$y_t(\beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \ldots + \beta_p x_{tp}) \geq M \; \forall \; t = 1, \ldots, T \tag{83c}$$

It can be shown that with constraint (83b) the perpendicular distance for the t[th] observation to the hyperplane is given by $y_t(\beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \ldots + \beta_p x_{tp})$. So (83b) in combination with (83c) ensures that the perpendicular distance of all observations from the hyperplane is at least of margin *M*. (83a) chooses $\beta_0, \beta_1, \beta_2, \ldots, \beta_p$ to maximize the perpendicular distances (only the distances of the observations which are closest to the separating hyperplane must be maximized because (83c) does not affect observations which are already far away from the hyperplane. Therefore, (83a) maximizes the distances of the support vectors as defined above (James et al., 2013 p. 337-343).

B) The non-linear-separable case (with linear class boundary) – the Support Vector Classifier

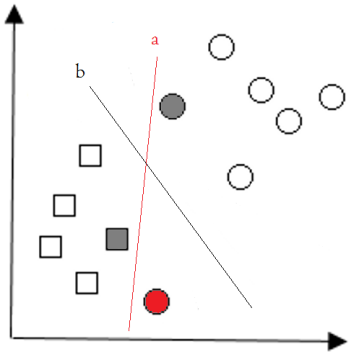
If there is no such a hyperplane that the data can be divided in separate classes, then the MMH does not exist and the optimization problem (83a) - (83c) has no solution with $M > 0$. An example of a non-separable two-dimensional data set is shown in Figure 4 on the left. But even in cases where a separable hyperplane is defined, this might not be desirable if the separator gets very sensitive to individual observations. Such a case is shown in Figure 5 on the right. In that case, one single observation (the circle in red) leads to a dramatic change of the MMH (from line b to line a). This hyperplane is not satisfactory because it has a very small margin. With a small margin the hyperplane is, like mentioned above, very sensitive (which means that new observations, which would vary just a little bit from the grey circle or grey square, would be easily classified wrongly by the separator a). The classifier b would be more preferable because it will likely generalize better to future observations, even if it does not correctly classify all the training data. In cases of non-separable data, or in cases in which it is not desired to perfectly separate the data, it is possible to extend the concept of the MMH in such a way that a hyperplane almost separates the classes correctly, using a so-called soft-margin. A soft-margin means in this context that not all training observations need to be strictly on the right side of the margin. In fact, some observations are not only

Figure 4: Non-linear separable data



Source: Own illustration

Figure 5: Very sensitive seperator



Source: Own illustration

allowed to be on the incorrect side of the margin, but they also are allowed to be on the incorrect side of the hyperplane (in the case of non-linear separable data, the latter is necessarily the case). Such a soft-margin classifier is the solution to the following optimization problem

$$\underset{\beta_0, \beta_1, \ldots, \beta_p, \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T}{\textit{maximize}\ M} \tag{84a}$$

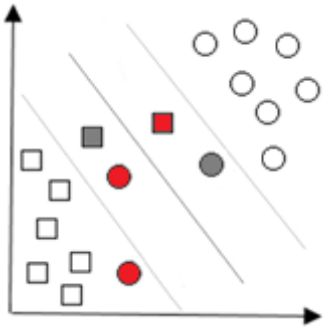
$$\textit{subject to}\ \sum_{j=1}^{p} \beta_j^2 = 1\ \textit{and} \tag{84b}$$

$$y_t(\beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \ldots + \beta_p x_{tp}) \geq M\ (1 - \varepsilon_i)\ \forall\ t = 1, \ldots, T \tag{84c}$$

$$\varepsilon_t \geq 0, \quad \sum_{t=1}^{T} \varepsilon_t \leq C \tag{84d}$$

where C is a non-negative tuning parameter and $\varepsilon_1, \ldots, \varepsilon_T$ are so-called slack variables, which allow individual observations to be on the wrong side of the margin or the hyperplane. The value of the slack variable can tell us if the observation is located on the right side of the margin or the hyperplane. $\varepsilon_t = 0$ for example means that the t[th] observation is on the right side of the margin, $\varepsilon_t \in]0, 1[$ means, that the t[th] observation is on the wrong side of the margin, but on the right side of the hyperplane and $\varepsilon_t > 1$ means, that the observation is on the wrong side of the hyperplane. The tuning parameter C is the sum of the slack variables and defines the amount of permitted violations of the margin. If $C = 0$, then no violations of the margin are allowed. (84a) – (84d) then simply becomes the MMH from equations (83a) – (83c), if a MMH exists at all. For $C > 0$, some observations are allowed to be on the wrong side of the margin or the hyperplane. The tuning parameter in this way regulates the toleration towards violations of the margin. With a rising C the margin will widen and correspondingly with a decreasing C the margin will narrow. The soft-margin classifier has the aforementioned property, that only observations, that lie on the wrong side of the margin or the hyperplane affect the classifier (the so-called support vectors). Figure 6 on the left shows a case with five support vectors: two on the right side of the hyperplane but on the wrong side of the margin, for whom $\varepsilon_i \in]0, 1[$ would hold (in grey) and three on the wrong side of the hyperplane, for whom $\varepsilon_i > 1$ would hold (in red). Figure B.2.1 and B.2.2 on pages 74f show the effect of a rising C where linear class boundaries have been applied to linear separable and non-linear separable data sets.

Figure 6: Five support vectors on the wrong side of the margin/hyperplane



Source: Own illustration

C) The non-linear-separable case (with non-linear class boundary) - the Support Vector Machine

In the aforementioned case B), where the data was not linear-separable, the classifier still tried to separate the classes by means of a linear class boundary. However, in practice there are often cases where a linear class boundary might not be suitable. In such cases the features can be transformed into a high-dimensional feature space, similar to linear regressions which use higher-order polynomials as features. To illustrate this idea, a possible quadratic relationship in the data should be regarded. In such a case, a

support vector classifier would not use the p feature vectors $\boldsymbol{x_1, x_2, \dots, x_p}$, but the 2p feature vectors $\boldsymbol{x_1, x_1^2, x_2, x_2^2 \dots, x_p, x_p^2}$. The optimization problem would then become

$$\underset{\beta_0, \beta_{11}, \beta_{12} \dots, \beta_{p1}, \beta_{p2}, \varepsilon_1, \varepsilon_2, \dots, \varepsilon_T}{\textit{maximize } M} \tag{85a}$$

$$\textit{subject to } \sum_{j=1}^{p} \sum_{k=1}^{2} \beta_{jk}^2 = 1 \tag{85b}$$

$$y_t(\beta_0 + \beta_{11} x_{t1} + \beta_{12} x_{t1}^2 + \dots + \beta_{p1} x_{tp} + \beta_{p2} x_{tp}^2) \geq M\,(1 - \varepsilon_t)\ \forall\ t = 1, \dots, T \tag{85c}$$

$$\varepsilon_i \geq 0, \quad \sum_{i=1}^{n} \varepsilon_i \leq C \tag{85d}$$

In this case, the decision boundary $y_t(\beta_0 + \beta_{11} x_{t1} + \beta_{12} x_{t1}^2 + \dots + \beta_{p1} x_{tp} + \beta_{p2} x_{tp}^2)$ would be a quadratic polynomial. Following this idea, the feature space could be transformed with higher-order polynomials, with interaction terms or other functions. The problem then is that there are many possible ways to transform the feature space, thus one would end up with a huge number of features. Computationally this would soon become unmanageable. For this reason, the SVM uses kernels, which are computationally more efficient transformations of the features. To reach this, kernels do not transform the features themselves, but the inner products of the features. To understand why it is sufficient to transform just the inner product of the features, the following paragraph describes the optimization problem in greater detail (James et al., 2013, p. 343-350).

Because the classification of an observation t to either class 1 or class -1 depends on whether the sign of $\beta_0 + \boldsymbol{\beta' x_t}$ is positive or negative, a positive rescaling will not affect the classification. For simplification, $\beta_0, \beta_1, \dots, \beta_p$ of the optimal separating hyperplane will be rescaled so that $\beta_0 + \boldsymbol{\beta' x_t} = 1$ for the observations of class 1, which are closest to the optimal separating hyperplane, and $\beta_0 + \boldsymbol{\beta' x_t} = -1$ for the observations of class -1, which are closest to the optimal separating hyperplane (the hyperplanes $\beta_0 + \boldsymbol{\beta' x} = \pm 1$ are called canonical hyperplanes). In absolute terms, for the closest observations of both classes (the support vectors) then $|\beta_0 + \boldsymbol{\beta' x_t}| = 1$ holds. The distance D of an observation t to the optimal separating hyperplane in general is given by $D = \frac{|\beta_0 + \boldsymbol{\beta' x_t}|}{||\boldsymbol{\beta}||}$, where $\left|\left|\boldsymbol{\beta}\right|\right|$ represents the norm of the vector $\boldsymbol{\beta}$. In combination with the scaling, the distance for the support vectors is equal to $D_{SV} = \frac{1}{||\boldsymbol{\beta}||}$, and the margin M, as twice the distance of the closest observations, is then $M = \frac{2}{||\boldsymbol{\beta}||}$. Instead of maximizing M, it is also possible to minimize a function L dependent on $\left|\left|\boldsymbol{\beta}\right|\right|$ like

$$L(\boldsymbol{\beta}) = \frac{1}{2}\left|\left|\boldsymbol{\beta}\right|\right|^2, \text{ subject to } \beta_0 + \boldsymbol{\beta' x_t} = 1\ \forall\ t \tag{86}$$

After canceling the root of $\boldsymbol{\beta'\beta}$ with the square in (86), the constraint optimization problem can be expressed as the following Lagrange function, also known as the primal formulation

$$L(\beta_0, \boldsymbol{\beta}, \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\beta'\beta} - \sum_{t=1}^{T} \alpha_t (y_t(\beta_0 + \boldsymbol{\beta' x_t}) - 1) \tag{87}$$

where $\boldsymbol{\alpha}$ are the T Lagrange multipliers $\alpha_1, \ldots, \alpha_T$ and thus $\alpha_t \geq 0$ must hold for all T observations. (87) can be reformulated to

$$L(\beta_0, \boldsymbol{\beta}, \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\beta}'\boldsymbol{\beta} - \boldsymbol{\beta}'(\sum_{t=1}^{T} \alpha_t \boldsymbol{x}_t) + \beta_0(\sum_{t=1}^{T} \alpha_t y_t) + \sum_{t=1}^{T} \alpha_t \quad (88)$$

The first-order conditions with respect to β_0 and $\boldsymbol{\beta}$ are then

$$\frac{\partial L}{\partial \beta_0} = -\sum_{t=1}^{T} \alpha_t y_t = 0 \quad (89a)$$

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = \boldsymbol{\beta} - \sum_{t=1}^{T} \alpha_t y_t \boldsymbol{x}_t = 0 \quad (89b)$$

Solving (89b) for $\boldsymbol{\beta}$ and replacing it in (88) leads to the dual formulation, also known as the Wolfe dual

$$W(\boldsymbol{\alpha}) = \sum_{t=1}^{T} \alpha_t - \frac{1}{2}\sum_{j=1}^{T}\sum_{t=1}^{T} \alpha_t \alpha_j \, y_t y_j \boldsymbol{x}_t' \boldsymbol{x}_j \quad (90)$$

(90) now needs to be maximized with respect to $\boldsymbol{\alpha}$ and subject to $\alpha_t \geq 0$ and $\sum_{t=1}^{T} \alpha_t y_t = 0$. In the optimizing problem (90), the optimal α_t will be found to be $\alpha_t = 0$ for all observations, which are not closest to the separating hyperplane. In this sense searching for the maximum-margin is equivalent to searching for the support vectors, since the separating hyperplane is defined by $\boldsymbol{\beta} = \sum_{t=1}^{T} \alpha_t y_t \boldsymbol{x}_t$. In the case of a soft-margin, the optimization problem becomes

$$L(\boldsymbol{\beta}) = \frac{1}{2}\left|\left|\boldsymbol{\beta}\right|\right|^2 + C\sum_{t=1}^{T} \varepsilon_t, \textit{ subject to } \beta_0 + \boldsymbol{\beta}'\boldsymbol{x}_t = 1 - \varepsilon_t \textit{ and } \varepsilon_t \geq 0 \;\; \forall \, t \quad (91)$$

and the primal formulation becomes

$$L(\beta_0, \boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\varepsilon}) = \frac{1}{2}\boldsymbol{\beta}'\boldsymbol{\beta} - \boldsymbol{\beta}'(\sum_{t=1}^{T} \alpha_t y_t \boldsymbol{x}_t) + \beta_0(\sum_{t=1}^{T} \alpha_t y_t) + \sum_{t=1}^{T} \alpha_t + \ldots$$
$$\ldots \sum_{t=1}^{T}(C - \alpha_t - \gamma_t)\,\varepsilon_t \quad (92)$$

where γ_t are the T Lagrange multipliers for the second constraint $\varepsilon_t \geq 0$. The first order conditions with respect to ε_t are $C - \alpha_t - \gamma_t = 0$ for all t. Since γ_t cannot be negative, $\alpha_t \leq C$ must hold as an additional condition. $C - \alpha_t - \gamma_t = 0$ also means that the second constraint drops in (92) so that the Wolfe dual stays unchanged

$$W(\boldsymbol{\alpha}) = \sum_{t=1}^{T} \alpha_t - \frac{1}{2}\sum_{j=1}^{T}\sum_{t=1}^{T} \alpha_t \alpha_j \, y_t y_j \boldsymbol{x}_t' \boldsymbol{x}_j \quad (93)$$

besides it is now subject to $0 \leq \alpha_t \leq C$ and $\sum_{t=1}^{T} \alpha_t y_t = 0$. Thereby the training data can be grouped into ***three subsamples***, where observations can be either a) no support vectors, b) support vectors on the margin (on the canonical hyperplane) or c) support vectors inside the margin. ***a)*** For $\alpha_t = 0$, the observation is not a support vector and thus lies outside the margin. ***b)*** For $0 \leq \alpha_t \leq C$, the observation is a support vector, which lies on a canonical hyperplane. ***c)*** For $\alpha_t = C$, γ_t must be zero due to the first order condition $c - \alpha_t - \gamma_t = 0$ derived with respect to ε_t. A multiplier of zero means, that the constraint is not binding on the lower bound, which means that $\varepsilon_t \neq 0$. Slack variables greater than zero signify that the margin is violated and that the observation is lying on the wrong side of the margin or hyperplane.

This foregoing more detailed description aimed to show that for the kernel transformation it is sufficient to transform just the inner product of the features. This can now be seen in (90) or (93). Because the features just appear as inner products in the optimization problem, it is sufficient to also only modify the inner product of the features and not the features themselves. Therefore a SVM uses kernels to enlarge the feature space in order to accommodate to non-linear boundaries between the classes. In this sense, SVMs can be described as extensions of support vector classifiers using non-linear transformed inner products of the feature space. The inner product of two observations $\boldsymbol{x_t}$ and $\boldsymbol{x_j}$ is thereby transformed using different kernel functions $K(\boldsymbol{x_t}, \boldsymbol{x_j})$, such as for example the polynomial kernels of degree d, defined as

$$K(\boldsymbol{x_k}, \boldsymbol{x_j}) = (1 + \sum_{k=1}^{p} x_{tk}\, x_{tk})^d \tag{94}$$

where d is an integer greater than 1 or a radial kernel, defined as

$$K(\boldsymbol{x_k}, \boldsymbol{x_j}) = exp(-\gamma \sum_{k=1}^{p} (x_{tk} - x_{jk})^2) \tag{95}$$

where γ is a positive constant. Some examples for radial kernels with different values for γ are provided in Figure B.2.3 on page 75.

These and other kernels then lead to much more flexible decision boundaries. Therefore, SVMs can be applied to many highly non-linear learning tasks (Campbell, 2011, p. 1-7).

2.2.2.3. Neural Networks

Artificial NNs are statistical learning models which are inspired by biological neural networks (i.e. central nervous systems and, in particular, the brain[44]). NNs are used to estimate generally unknown functions with a large number of features. ***Two advantages*** of NNs are that they ***a)*** can model highly complex relationships between features and responses (especially non-linear relationships) and that ***b)*** no underlying assumptions are needed to create and evaluate the models. ***Three disadvantages*** are that ***a)*** NNs are black box techniques, meaning that their results are almost not interpretable, ***b)*** that their results can differ depending on the initialization of the weights, and ***c)*** that NNs are computationally expensive.

NNs are frequently applied to practical problems, such as speech and handwriting recognition, self-driving cars and self-piloting drones and an uncountable amount of scientific, social and economic phenomena. In short, NNs are versatile learners that can be applied to nearly all learning tasks. To illustrate the concepts of NNs in the following, the descriptions are divided into ***five subparts***. Part ***A)***

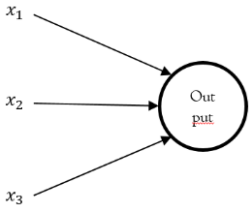
[44] NNs as well as biological neural networks uses interconnected neurons to process information. Some biological examples of neural networks can give a feeling for the number of connected neurons: the nervous system of a small worm consists of 302 neurons, 10^4 neurons make an ant, 10^5 a fly, $4 * 10^6$ neurons results in a mouse, $1{,}6 * 10^8$ neurons are necessary for a dog, $3 * 10^8$ for a cat, with $6 * 10^9$ neurons the nervous system of a chimpanzee can be constructed, the nervous system of humans are constructed with around 10^{11} neurons and for elephants and certain whales even $2 * 10^{11}$ neurons are necessary (Kriesel, 2005, p.28f).

first outlines some basic structures of NNs. Part ***B)*** then shows a commonly used learning (or optimization) algorithm called gradient descent. Part ***C)*** then describes how through backpropagation, the gradients for the gradient descent algorithms can be calculated. Part ***D)*** shows an alternative cost function called cross-entropy and the last part ***E)*** gives some ideas about more complex NNs and Deep Learning.

A) The basic structure of NNs:

A NN consists of many interconnected artificial neurons, whereby the connections between the neurons as well as the neurons themselfes can take on different forms. A neuron executes two consecutive calculations: a linear combination of its input followed by (in most cases) a non-linear transformation of the results to obtain the output value. To construct a NN, first an architecture needs to be established (where the number of neurons and layers is chosen), and then an algorithm is applied to find the weights of the connections between the neurons (Torgo, 2011, p. 123). To get a good understanding of how NN's work, it can be helpful to first depict the functionality of one single artificial neuron called a perceptron-neuron. A perceptron (as displayed in Figure 7) takes p binary inputs x_1, x_2,..., x_p and produces one single binary output – so far only one point in time is regarded). Thereby, depending on the importance, each input is weighted with a weight w_j and subsequently the sum over all weighted inputs, $\sum_{j=1}^{p} w_j x_j$, is computed. The neurons output then is determined by whether the weighted sum $\sum_{j=1}^{p} w_j x_j$ is more or less than a threshold (also called bias b), so that $\sum_{j=1}^{p} w_j x_j \leq b$ or $\sum_{j=1}^{p} w_j x_j > b$. Using the bias $b = -threshold$, this can be reformulated to

Figure 7: Percepton Neuron



Source: Own illustration

$$output \begin{cases} 1 & if \ \sum_{j=1}^{p} w_j x_j + b > 0 \\ 0 & if \ \sum_{j=1}^{p} w_j x_j + b \leq 0 \end{cases} \quad (96)$$

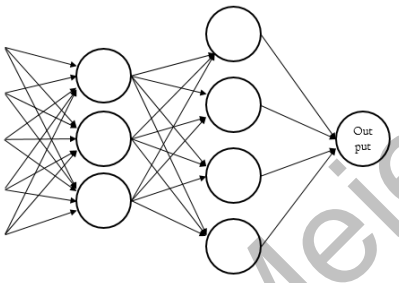
Thereby $\sum_{j=1}^{p} w_j x_j + b$ is represented by z and called weighted input in the following so that $\sum_{j=1}^{p} w_j x_j + b = z$ and the function, which defines the output for a given weighted input is called activation function - therefore, in the case of a perceptron, the activation function is a step function with a step at value zero.

A perceptron transforms several input signals into one output signal depending on the weights, the bias and the activation function. The weights and the bias can be thought of as a kind of decision rule in deciding to whether or not to signal an output[45]. Such perceptrons now can be combined and connected

[45] For example, a person could decide, given three circumstances, whether or not he or she wants to go to a concert. The circumstances could be A) whether the band is good or not, B) whether the weather is good or not and C) whether some friends are joining the concert or not. For the weights $w_1 = 6$, $w_2 = 2$, $w_3 = 2$ and the threshold $b = 5$, the person´s decision to go to the concert ($output = 1$) would be completely determined by the fact whether the band is good or not

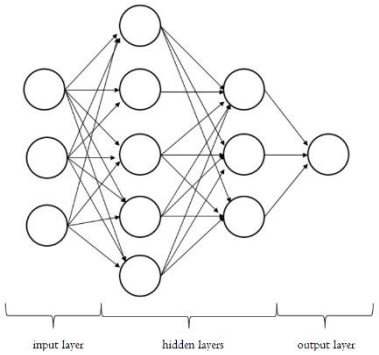
to an artificial network of perceptrons, as displayed in Figure 8 on the right. Each perceptron still has one single output, but the output is connected to several other perceptrons representing inputs for other perceptrons. Such a network then would transform several binary input signals into (not necessarily one) binary output signal depending on a complex decision rule based on all the weights and biases of the perceptrons. A network of perceptrons (or neurons in general) can be thought of as consisting of several layers or columns. The first column, consisting of the input signals, is called the input layer[46], the subsequent columns, consisting of neurons, are called hidden layers and the last column, consisting of the output neurons, is called output layer (as depicted in the Figure 9 on the left). Where there is always one input and one output layer, the number of hidden layers as well as the number of neurons in each layer has to be chosen manually - depending on the complexity and the aim of the task. It can be shown that with a network of neurons, it is not only possible to compute elementary logical functions like AND, OR and NAND functions, but in fact, it is possible to compute any logical function (which means that a network of neurons can theoretically compute anything). And even better, due to learning algorithms, which alter the weights and biases after rules which will be depicted later, a NN can learn to compute any logical gate without being explicitly programmed. Appendix B.2 provides some small examples how NNs can compute different logical gates.

Figure 8: Network of perceptrons



Source: Own illustration

Figure 9: Input, hidden and output layer



Source: Own illustration

Because learning to compute a desired output given some input values means changing the weights and biases of the network, perceptrons have some undesired properties. Because of the step function, a small change in a weight or bias of a perceptron can sometimes cause the output of that perceptron to completely flip around. This again can change the behavior of the network in complicated ways. In order to improve learning, it would be desirable that a small change in a weight or bias also corresponds to a small change in the output of the network. Therefore, perceptrons are rarely used in today´s NN`s. Instead of perceptrons, sigmoid neurons are the main neurons used in much modern NNs. Sigmoid neurons also take several inputs x_1, x_2,..., x_p and produce one single output, but instead of being zero or one, the output of a sigmoid neuron can be any number between zero and one. The activation function

(for the inputs $x_1 = 1$, $x_2 = 0$ and $x_3 = 0$ the output would be 1 because $\sum_{j=1}^{3} w_j x_j = 6 > b = 5$ and for the inputs $x_1 = 0$, $x_2 = 1$ and $x_3 = 1$ the output would be 0 because $\sum_{j=1}^{3} w_j x_j = 4 < b = 5$). In the case that the threshold would be set to 3, the person also would go to the concert if the band is bad if only the weather is good and some friends would join.

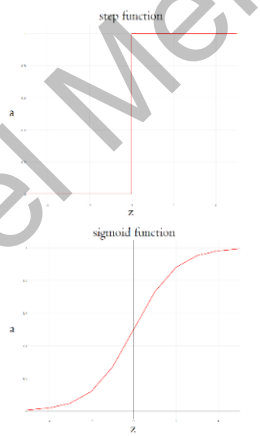
[46] The input signals are conventionally drawn as neurons (represented by circles). Therefore, they could be erroneously regarded as neurons without inputs, whereby technically they aren´t neurons, but just variables (a neuron without inputs would always output a constant and not the desired value of the input signal).

of a sigmoid neuron also computes the sum of the weighted inputs plus a bias (represented by z) and then uses this value as input for the activation function, which in the case of a sigmoid neuron is a sigmoid function defined as

$$\sigma(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+exp(-\sum_{j=1}^{p} w_j x_j - b)} \tag{97}$$

Thereby $\sigma(z)$ is also often referred to as the output or activation a of the neuron, so that $\sigma(z) = a$. The sigmoid neurons are not completely different from perceptrons because for very large positive or very large negative z values, the sigmoid neuron would output a one or a zero respectively, just like a perceptron[47]. It is only when z is of modest values around zero when sigmoid neurons behave differently from perceptrons. This can be seen best when the sigmoid function is compared to the step function of the perceptron, as displayed in Figure 10 on the rigth. In fact, the sigmoid function is just a flattened and smoothed step function and therefore small changes in the weights or the bias cause the desired small changes in the output of the sigmoid neuron. Although the sigmoid function is perhaps the most commonly used activation function, frequently used alternative functions would be linear functions, saturated linear functions, hyperbolic tangents or Gaussian functions. With linear activation functions, NNs very similar to linear regression models can be constructed. Typically, NNs need to restrict the range of the input signals because otherwise this would result in saturated activations, so that large-value features would dominate small-valued-features (because for a sigmoid function with inputs $< |5|$ the output will always be around 0 or 1). Therefore, the features are first standardized or normalized so that the activation function can have action on the entire range (Lantz, 2013, p. 224 & Nielson, 2017, chap. 1).

Figure 10: Step and sigmoid function



Source: Own illustration

B) Learning with gradient descent

In order to learn, there must be some measure for the quality of the model, which then can be improved while learning. Thereby, the quality is measured with a function of the predicted output a and the actual true output y (such functions are referred to as cost or loss functions). One of the most commonly used cost functions is the quadratic cost function[48], which is then averaged over all training examples to obtain the overall cost C, defined as

$$C(\boldsymbol{W}, \boldsymbol{B}) = \frac{1}{2n}\sum_{t=1}^{T}\|y_t - a_t^L\|^2 = \frac{1}{T}\sum_{t=1}^{T} C_t \tag{98}$$

[47] Because $e^{-z} \to 0$ and therefore $\sigma(z) \to 1$ as $z \to \infty$ and $e^{-z} \to \infty$ and therefore $\sigma(z) \to 0$ as $z \to -\infty$.

[48] The quadratic cost function for one training example t is defined as $C_t = \frac{1}{2}\|y_t - a_t^L\|^2$.

where $\boldsymbol{W}$ and $\boldsymbol{B}$ are matrices containing all weights and all biases of the network and y_t and a_t^L are scalars containing the actual true, as well as the predicted outputs for the training example t (L indicates the amount of layers in the network so that a_t^L represents the outputs of the neurons in the last layer of training example t). The cost function becomes small when the predicted output is equal to the actual true output, so that $y_t \approx a_t \; \forall \; t$, and therefore, the aim of the training algorithm is to find those weights and biases, that minimize the cost function $C(\boldsymbol{W}, \boldsymbol{B})$. This minimization is done by an algorithm called gradient descent. Thereby the weights and biases of the network are updated by a gradient descent update rule, defined as

$$\boldsymbol{W} \rightarrow \boldsymbol{W'} = \boldsymbol{W} - \eta \frac{\partial C}{\partial \boldsymbol{W}} \tag{99a}$$

$$\boldsymbol{B} \rightarrow \boldsymbol{B'} = \boldsymbol{B} - \eta \frac{\partial C}{\partial \boldsymbol{B}} \tag{99b}$$

whereby $\frac{\partial C}{\partial \boldsymbol{W}}$ and $\frac{\partial C}{\partial \boldsymbol{B}}$ are the gradient matrices containing the partial derivatives with respect to all weights and all biases of the network. η is a small positive parameter called the learning rate and it determines how fast the network is learning. For example, if the change of the cost for one training example due to a specific weight w is defined as $\Delta C_t = \frac{\partial C_t}{\partial w} \Delta w$, then $\frac{\partial C}{\partial w}$, the gradient, also written as ∇C_t^w, relates changes in the weight w to changes in the cost C_t. Therefore, if Δw is choosen as

$$\Delta w = -\eta \nabla C_t \tag{100}$$

then, the change in the cost becomes $\Delta C_t = -\eta \|\nabla C_t\|^2$. Because η and $\|\nabla C_t\|^2$ are positive, then, if the weight w is changed as in (100), the change in the cost will always be negative, i.e. the cost will be reduced. In this way, all weights and biases of a NN are changed to find the minimal cost for a training example C_t. This is repeated for each training example to find the minimal cost for each training example and subsequently, the mean is computed as in (98) to obtain the overall cost for all training examples C (if the data set is large, then there are methods like mini-batch or stochastic gradient descent, where only some training examples are used to compute C) (Nielson, 2017, chap. 2).

C) The backpropagation algorithm

As described above, for the gradient descent update rule, which minimizes the cost function, it is necessary to compute the gradients, i.e. the derivations of the cost with respect to the weights and biases. The computation of the gradients is reached by applying the backpropagation algorithm, whereby the following notation is used: w_{jk}^l denotes the weight for the connection from the kth neuron in the (l-1)th layer to the jth neuron in the lth layer[49], and b_j^l as well as a_j^l denotes the bias, respectively the activation of

[49] Thereby the order of j and k is somewhat against the more intuitive order of the layers, which is from left to right, because the layer with the neuron-reference j is right (and not left) of the layer with the neuron-reference k.

the j[th] neuron in the l[th] layer. With this notation, the activation of the j[th] neuron in the l[th] layer is defined as

$$a_j^l = \sigma\left(\sum_{k=1}^{n_{l-1}} w_{jk}^l a_k^{l-1} + b_j^l\right) = \sigma\left(z_j^l\right) \tag{101}$$

where n_{l-1} is the number of neurons in the (l-1)[th] layer. This can be rewritten in matrix notation as $\boldsymbol{a}^l = \sigma(\boldsymbol{W}^l \boldsymbol{a}^{l-1} + \boldsymbol{b}^l) = \sigma(\boldsymbol{z}^l)$, where $\boldsymbol{W}^l$ denotes the weight matrix of the connections to the l[th] layer of neurons (with w_{jk}^l as the entry of the k[th] row and j[th] column defined as above), and with $\boldsymbol{b}^l$, $\boldsymbol{a}^l$ and $\boldsymbol{z}^l$ as vectors of biases, activations and weighted inputs of the neurons in layer l. To compute the gradients, also an intermediate quantity δ_j^l, in the following referred to as the error – in this case the error in the j[th] neuron in the l[th] layer – is defined as $\delta_j^l = \frac{\partial C_t}{\partial z_j^l}$, whereby $\boldsymbol{\delta}^l$ refers to the vector of errors in the l[th] layer. The backpropagation algorithm can be applied in three steps: *(1)* First the input signal is fed forward through the network. Because the network contains no a priori knowledge, this is done by using randomly chosen starting values (Lantz, p. 230). *(2)* Thereafter the errors of every layer $\boldsymbol{\delta}^l$ are computed. Thereby first the error of the last layer L is computed as[50]

$$\delta_j^L = \frac{\partial C_t}{\partial a_j^L}\sigma'(z_j^L) = \frac{\partial C_t}{\partial a_j^L}\sigma(z_j^L)\sigma(1 - z_j^L) \tag{102}$$

The error of the last layer displays first, how strong the cost depends on a specific neuron, represented by the first term on the right side, and second, how fast the activation is changing with z_j^L. (102) can be expressed componentwise as $\boldsymbol{\delta}^L = \nabla_{\mathrm{a}} \boldsymbol{c}_t \odot \sigma'(\boldsymbol{z}^L)$, with $\nabla_{\mathrm{a}} \boldsymbol{c}_t$ as a vector of the partial derivatives $\frac{\partial C_t}{\partial a_j^L}$ for $j = 1, \ldots, n$, with n as the number of neurons in layer L and $\odot$ as the Hadamard product[51]. The error of the last layer L then can be propagated back through the network to compute the errors of the other layers defined as[52]

---

[50] (102) can be derived as follows: $\delta_j^L = \frac{\partial C_t}{\partial z_j^L} = \sum_{k=1}^{n_L} \frac{\partial C_t}{\partial a_k^L}\frac{\partial a_k^L}{\partial z_j^L}$ and $\frac{\partial a_k^L}{\partial z_j^L} = 0$ for $k \neq j$ because the activction only depends on the weighted input for $k = j$ and therefore $\delta_j^L = \frac{\partial C_t}{\partial a_j^L}\frac{\partial a_j^L}{\partial z_j^L}$. Because a_j^L is defined as $a_j^L = \sigma(z_j^L)$ this can be rewritten as $\delta_j^L = \frac{\partial C_t}{\partial a_j^L}\sigma'(z_j^L)$ and with the derivation of $\sigma'(z_j^L) = \frac{\partial \frac{1}{1+exp(-z_j^L)}}{\partial z_j^L} = -\left(1 + exp\left(-z_j^L\right)\right)^2(-1)\, exp\left(-z_j^L\right) = \frac{exp\left(-z_j^L\right)+1-1}{\left(1+exp\left(-z_j^L\right)\right)^2} = \frac{1}{1+exp\left(-z_j^L\right)} - \frac{1}{\left(1+exp\left(-z_j^L\right)\right)^2} = \frac{1}{1+exp\left(-z_j^L\right)}\left(1 - \frac{1}{1+exp\left(-z_j^L\right)}\right) = \sigma(z_j^L)\sigma(1 - z_j^L)$ this can be written as in (102).

[51] The Hadamard product, also called Schur product, is an operator for elementwise multiplications. If for example $\boldsymbol{s}$ and $\boldsymbol{t}$ are two vectors, defined as $\boldsymbol{s} = \begin{bmatrix}1\\2\end{bmatrix}$ and $\boldsymbol{t} = \begin{bmatrix}3\\4\end{bmatrix}$, then the Hadamard product $\boldsymbol{s} \odot \boldsymbol{t}$ is defined as $\begin{bmatrix}1\\2\end{bmatrix} \odot \begin{bmatrix}3\\4\end{bmatrix} = \begin{bmatrix}1 * 3\\2 * 4\end{bmatrix} = \begin{bmatrix}3\\8\end{bmatrix}$.

[52] (103) can be derived as follows: δ_j^l was defined as $\delta_j^l = \frac{\partial C_t}{\partial z_j^l} = \sum_{k=1}^{n_{l+1}} \frac{\partial C_t}{\partial z_k^{l+1}}\frac{\partial z_k^{l+1}}{\partial z_j^l} = \sum_{k=1}^{n_{l+1}} \delta_k^{l+1}\frac{\partial z_k^{l+1}}{\partial z_j^l}$ and z_k^{l+1} was defined as $z_k^{l+1} = \sum_{i=1}^{n_l} w_{ki}^{l+1} a_i^l + b_k^{l+1} = \sum_{i=1}^{n_l} w_{ki}^{l+1}\sigma(z_i^l) + b_k^{l+1}$. This implies, that $\frac{\partial z_k^{l+1}}{\partial z_j^l} = 0$ for $i \neq j$ such that $\frac{\partial z_k^{l+1}}{\partial z_j^l} =$

$$\boldsymbol{\delta}^l = ((\boldsymbol{W}^{l+1})'\boldsymbol{\delta}^{l+1}) \odot \sigma`(\boldsymbol{z}^l) = ((\boldsymbol{W}^{l+1})'\boldsymbol{\delta}^{l+1}) \odot \sigma(\boldsymbol{z}^l) \odot \sigma(\boldsymbol{\iota} - \boldsymbol{z}^l) \quad (103)$$

where $(\boldsymbol{W}^{l+1})'$ is the transpose of the weight matrix for the (l+1)th layer and $\boldsymbol{\iota}$ is a vector of ones with the same length as neurons in layer l. Thereby, the left term of the Hadamard product moves the error back to the output of the lth layer and the right term of the Hadamard product moves the error through the activation function to obtain the error for the weighted input to the lth layer $\boldsymbol{\delta}^l$. *(3)* After the errors for all layers are computed, the errors can be related to the gradients of the weights[53] and biases[54] as follows

$$\frac{\partial C_t}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l \quad (104a)$$

$$\frac{\partial C_t}{\partial b_j^l} = \delta_j^l \quad (104b)$$

From (104a) can be seen that, if a_k^{l-1} is low, then $\frac{\partial C_t}{\partial w_{jk}^l}$ is low and therefore a change of the cost provoked by a change of a weight is also low (a_k^{l-1} can be called the input activation for the weight w_{jk}^l). Similarly, (104a) shows that if δ_j^l is low, then $\frac{\partial C_t}{\partial w_{jk}^l}$ will also be low, whereby from (102) can be seen, that δ_j^l is low, when $\sigma`(z_j^l)$ is low (for the sigmoid function this is true when z_j^l is either very high or very low, meaning that the output activation is either very high or very low). Therefore, $\frac{\partial C_t}{\partial w_{jk}^l}$ is low, if either the input activation for the weight is very low, or, if the output activation for the weight is very high or very low. In such a case the weight is said to learn slowly.

The whole minimization of the cost function can be described in ***9 steps:*** *a)* The features for the test observation provide the inputs or activations $\boldsymbol{a}^1$ of the NNs input layer. *b)* For each layer $l = 2, \ldots, L$ $z_j^l = \sum_{k=1}^{n_{l-1}} w_{jk}^l a_k^{l-1} + b_j^l$ and with this $a_j^l = \sigma(z_j^l)$ are computed. *c)* The error vector of the last layer is computed as $\boldsymbol{\delta}^L = \nabla_a \boldsymbol{c}_t \odot \sigma`(\boldsymbol{z}^L)$. *d)* These errors are propagated back through the network with

$w_{kj}^{l+1}\sigma'(z_j^l)$. Therefore δ_j^l can be reformulated to $\delta_j^l = \sum_{k=1}^{n_{l+1}} \delta_k^{l+1} w_{kj}^{l+1}\sigma'(z_j^l)$. Written in component form this results in equation (103).

[53] (104a) can be derived as follows: $\frac{\partial C_t}{\partial w_{jk}^l} = \frac{\partial C_t}{\partial a_j^l}\frac{\partial a_j^l}{\partial w_{jk}^l}$ and with $a_j^l = \sigma\left(\sum_{i=1}^{n_{l-1}} w_{ji}^l a_i^{l-1} + b_j^l\right)$ the derivation $\frac{\partial a_j^l}{\partial w_{jk}^l}$ gets zero for $i \neq k$ and therefore $\frac{\partial a_j^l}{\partial w_{jk}^l} = a_k^{l-1}\sigma'\left(\sum_{i=1}^{n_{l-1}} w_{ji}^l a_i^{l-1} + b_j^l\right) = a_k^{l-1}\sigma'(z_j^l)$. The derivation $\frac{\partial C_t}{\partial w_{jk}^l}$ then becomes $\frac{\partial C_t}{\partial w_{jk}^l} = a_k^{l-1}\frac{\partial C_t}{\partial a_j^l}\sigma'(z_j^l)$, and with $\frac{\partial C_t}{\partial a_j^l}\sigma'(z_j^l) = \delta_j^l$ as defined in (102) this can be reformulated to $\frac{\partial C_t}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l$ as defined in (104a).

[54] (104b) can be derived as follows: $\frac{\partial C_t}{\partial b_j^l} = \frac{\partial C_t}{\partial a_j^l}\frac{\partial a_j^l}{\partial b_j^l}$ and with $a_j^l = \sigma\left(\sum_{k=1}^{n_{l-1}} w_{jk}^l a_k^{l-1} + b_j^l\right)$ the derivation $\frac{\partial a_j^l}{\partial b_j^l}$ gets $\frac{\partial a_j^l}{\partial b_j^l} = \sigma`\left(\sum_{k=1}^{n_{l-1}} w_{jk}^l a_k^{l-1} + b_j^l\right) = \sigma`(z_j^l)$. Therefore $\frac{\partial C_t}{\partial b_j^l} = \frac{\partial C_t}{\partial a_j^l}\sigma`(z_j^l)$ what is equal to $\frac{\partial C_t}{\partial b_j^l} = \delta_j^l$ as defined in (104b).

$\boldsymbol{\delta^l = ((W^{l+1})^T \delta^{l+1}) \odot \sigma`(z^l)}$. ***e)*** The gradients of the cost function are computed with $\frac{\partial C_t}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l$ and $\frac{\partial C_t}{\partial b_j^l} = \delta_j^l$. ***f)*** a) – e) is repeated for all training examples. ***g)*** The mean of the gradients over all training examples are computed so that $\frac{\partial C}{\partial \boldsymbol{W}} = \frac{1}{n}\sum_{t=1}^{n} a_k^{t,l-1}\delta_j^{t,l}$ and $\frac{\partial C}{\partial \boldsymbol{B}} = \frac{1}{n}\sum_{t=1}^{n} \delta_j^{t,l}$, where the subscript $\boldsymbol{t, l}$ denotes observation t and layer l. ***h)*** The weights and biases are updated according to the rules $\boldsymbol{W \rightarrow W' = W - \eta \frac{\partial C}{\partial W}}$ and $\boldsymbol{B \rightarrow B' = B - \eta \frac{\partial C}{\partial B}}$. ***i)*** a) – h) is repeated either a predefined number of times or it is repeated until some threshold value for the cost or the gradients is reached (Nielson, 2017, chap. 2).

D) The cross-entropy cost function

As described above, weights are said to learn slowly, if either the input activation for the weight is very low, or if the output activation for the weight is very high or very low. This can be a problem if the input or output activations are far away from their true value, i.e. an activation is near zero when it should be one or vice versa. In such a case the error would be very high, but the learning very slow. A desired property of learning in general is, however, that - just like humans – NNs should learn fast, i.e. change the weights and biases fast – when they make big mistakes, and learn slower and more tedious when the mistakes get smaller so that they can get a better fine tuning. From (104a) and (104b) can be seen that the changes of the gradients are low, i.e. learning is slow, when δ_j^l is small, and then again δ_j^l is small when $\sigma`(z_j^l)$ is small, because

$$\delta_j^l = \frac{\partial C_t}{\partial a_j^l}\sigma`(z_j^l) \tag{105}$$

whereby the first derivation of the sigmoid function is defined as $\sigma`(z_j^l) = \sigma(z_j^l)(1 - \sigma(z_j^l)) = a_j^l(1 - a_j^l)$. With the quadratic cost function $C_t = \sum_{j=1}^{n_L} \frac{1}{2}\left\|y_j - a_j^L\right\|^2$ the error $\delta_j^L$ will be $\delta_j^L = (y_j - a_j^l)\sigma`(z_j^l)$ - as described, the problematic term is $\sigma`(z_j^l)$. Now, with the cross-entropy cost function defined as $C_t = -\sum_{j=1}^{n_L} y_j \ln(a_j^L) + (1 - y_j)\ln(1 - a_j^L)$, the derivation $\frac{\partial C_t}{\partial a_j^L}$ will be $\frac{\partial C_t}{\partial a_j^L} = \frac{y_j}{a_j^L} - \frac{1-y_j}{1-a_j^L}$ and therefore the error δ_j^L will be

$$\delta_j^L = \left(\frac{y_j}{a_j^L} - \frac{1-y_j}{1-a_j^L}\right)\sigma`(z_j^l) = \left(\frac{y_j}{a_j^L} - \frac{1-y_j}{1-a_j^L}\right)a_j^L(1 - a_j^L) = y_j(1 - a_j^L) - (1 - y_j)a_j^L \tag{106a}$$

$$\delta_j^L = y_j - a_j^L \tag{106b}$$

which means that the problematic term $\sigma`(z_j^l)$ disappears from the definition of δ_j^l. With (106b), now the gradients in (104a) and (104b) will be large, when the error is large, meaning that the weights and biases are changing fast, and it will be small, when the error is small, meaning that the weights and biases are changing slowly – which is the desired property (Nielson, 2017, chap. 3).

E) Some insights into more complex NNs and Deep Learning

Most NNs contain one or two hidden layers, because the amount of parameters is already very high so that the model tends to overfitting (thereby, of course, also the number of neurons in each layer matters). But depending on the amount of available observations and the complexity of the task, also NNs with many hidden layers can be constructed. Such networks are called Deep Neural Networks and the practice of training such Deep Neural Networks is referred to as Deep Learning. Another type of variety lies in the direction of the information flow. In the NNs described above, the input signals were passed from left to right towards the output neurons, so that signals were only travelling in one direction. Such NNs are called feedforward networks. In contrast, feedback networks (or recurrent networks) allow signals to travel in both directions using loops. Such networks have a kind of short-term memory, because events can be modelled with their time structure. Such recurrent networks are closer to how biological neural networks work and are capable of learning extremely complex patterns. Recurrent neural networks are nevertheless rarely used in practice and still largely theoretical. Especially for time series data, however, they could have great potential (Nielson, 2017, chap. 6).

3. Empirical results

The following section provides some empirical results for the described time series as well as ML models applied to daily data of the USD/GBP exchange rate. Before the results are presented in part 3.2, part 3.1 displays how the performance of the empirical results has been measured.

3.1. Assessing performance

Features and responses may always also reflect noise, which means that instead of observing the feature $\mathbf{x}$ or the response $\mathbf{y}$ we might observe some corrupted $\mathbf{x}'$ or $\mathbf{y}'$. Because of noise in the data, it is generally not advisable to try to match the model exactly to the data, as this would also lead to incorporating the noise into the model. In such a case, the model is said to be overfitted. Hence, measuring the fit of a model to the data can be misleading. To account for this, the data set is split into a training and a testing data set and the performance of the models is measured by calculating predictions for newly unseen responses – observation in the testing data set – and then compare this predictions to the actual true responses. Because time series observations are chronologically ordered, they cannot be seen as drawn randomly from an independent and identically distributed underlying population. In addition, it is reasonable to assume that for the prediction of the response y_t, all previous responses $y_{t-1}, y_{t-2}, \dots, y_1$ and all previous features $x_{t-1}, x_{t-2}, \dots, x_1$ can be used. Therefore, the training data set is updated after each prediction in this way, that the first observation from the training data is dropped, and the first testing observation is included in the updated training data set. The size of the training data size thus

stays constant – this is called sliding window approach in the following.[55] (Torgo, 2011, p. 121f). For each prediction in the testing data set the ***performance then is measured in two ways***, where for the first way ***A)***, only the direction of the prediction and for the second way ***B)***, the distinct value of the prediction is assessed.

A) Assessing the performance of the direction of the predictions

To asses the performance of the direction of the predictions, the response was modified in this way, that positive changes of the USD/GBP exchange rate have been labeled with "True+" and negative changes of the USD/GBP exchange rate have been labeled with "True –". Then, in case of a classification task, the classification algorithm was trained to predict a positive or a negative class (labeled as "Predicted+" or "Predicted-"). But also when the task was a regression task or when classical time series models were applied to forecast non-discrete values of the USD/GBP exchange rate, the value of the prediction have been labeled as "Predicted +" and "Predicted-". In this way, the predicted sign can be compared to the actual change of the exchange rate in the testing data set (labeled as "True+" and "True-"). The performance of predicting the right sign can then be assessed using contingency tables, in which the number of true and false predictions are displayed. Such a contingency table is shown in Table 1, where true predictions are marked in grey and false predictions are marked in red. True predictions are all "True positives" and "True negatives", while false predictions are all "False negatives" and "False positives"[56]. From a contingency table, a number of indicators can be calculated, with the most important one being the accuracy defined as

Table 1: Contingency table

	Predicted +	Predicted –	
Actual +	True positives T_p	False negatives	# of actual positives
Actual –	False positives	True negatives T_n	# of actual negatives
	# of positive predictions	# of negative predictions	# of all observations N

Annotation: # = number Source: Own illustration

$$accuracy = \frac{T_p + T_n}{N} \tag{107}$$

B) Measuring the performance of the value of the predictions

Because of the serial correlation of time series data, it is not possible to simply use the correlation between the predicted and the actual response as a measure for the performance of the value of the predictions. Additionally, because predictions of different responses should be made comparable, instead of using the commonly applied Mean Absolute Error[57], the Mean Absolute Scaled Error (MASE) will be used for testing the performance of the predictions. The MASE is defined as

$$MASE = \frac{1}{T}\sum_{j=1}^{T} \frac{|\hat{y}_j - y_j|}{\frac{1}{T}\sum_{t=1}^{T} |y_{t-1} - y_t|} \tag{108}$$

[55] For example, the first model uses the training data with time labels $t = 1$ to $t = j - 1$ to predict the response in $t = j$. Then the subsequent model uses the training data with time label $t = 2$ to $t = j$ to predict the response in $t = j + 1$.

[56] True/false refers to whether the prediction is correct or not and positive/negative refers to the actual true change of the response.

[57] The Mean Absolute Error (MAE) is defined as $MAE = \frac{1}{T}\sum_{t=1}^{T} |\hat{y}_t - y_t|$.

where T is the number of tested predictions in the testing data. The nominator of the MASE is equal to the absolute error (where the error is simply the difference between the prediction $\hat{y}_t$ and the actual response y_t) and the denominator scales each absolute error with the mean absolute error of a naïve prediction[58] over all test observations. A MASE greater than one signals the predictions have been worse compared to naïve forecasts for the training data set (Hyndman, 2012, chap. 2 and 5).

3.2. Text mining and the exchange rate

In order to test the relationship between news announcements (NAs) and the USD/GBP exchange rate, ***two methods*** where the Multivariate Bernoulli Naïve Bayes Classifier has been applied are described in the following, whereby the first method is described in subpart ***A)*** and the second method in subpart ***B)***.

A) First method

For the following ***first method***, ***1)*** the transformation of the NA`s into a term document matrix and ***2)*** the labeling of the NAs, as well as the final results are described.

1) Transformation of data: In a first step 536.309 short text NAs with relevance to foreign exchange rates were crawled from the webpage www.fxstreet.com. The NAs cover the period between 2008/01/02 and 2016/10/27 and are tagged with minutely time designations. Out of these 536.309 NAs, 82.476 NAs were filtered[59] in order to obtain only the NAs with relevance to the USD/GPB exchange rate. In a next step, all the minutely NAs for one day were merged to one long NA, so that for every day 20-40 NAs were concatenated to a long string containing all the NAs of one day. This results in 2330 combined NAs – one for each day (whereby each NA contained between about 300 to 3000 words, depending on the amount of relevant NAs for each day). Subsequently, the NAs were transformed so that firstly, numbers, punctuations and stop words[60] were removed, and secondly, each word was stemmed to its root[61]. The remaining word stems then were transformed into a term document matrix containing 10713 unique words (rows) and 2330 documents (columns).

2) Labeling of NAs and results: In order to train a classifier to detect whether a NA is positive or negative, the NAs need an already labeled data set. Therefore, the NAs were labeled similarly to the approach applied by Aaese (2011, p. 45f). Thereby, a NA was labeled as positive (negative), if the subsequent change in the exchange rate was positive (negative). With the features (the term document matrix) and the corresponding labels, a Multinomial Naïve Bayes Classifier was trained with 1500 of the 2330 NAs.

[58] The naïve prediction simply uses the last observed value in t-1 as a predictor for period t, such that $\hat{y}_t = y_{t-1}$.

[59] Thereby, each NA was examined with if-loops, thereby checking, if a word in a specific list is present, while at the same time a word of another specific list is not present. For example, if one of the words in {gbp, pound, uk, united kingdom, usa, us dollar, fed, redbook, …} was present in the NA, while no word of the list {cad, jpy, Canada, ecb, euro, yen, …} was present in the NA, then the NA was considered as relevant for the USD/GBP exchange rate.

[60] Stop words are common word like "and", "or", "the" or "it".

[61] Stemming words to its root means cutting word ends like "ed", "ing" or "er". For example the words "fished", "fishing" or "fisher" all would have been stemmed to the root "fish".

According to the probabilities of the trained classifier, the remaining 830 NAs were then classified as either positive or negative. In a final step, the 830 predictions were compared to the actual changes of the USD/GBP exchange rate (whereby a positively (negatively) classified NA was taken as a prediction, that the subsequent change in the exchange rate would be positive (negative)). Out of this 830 predictions, only 434 were true predictions, leading to an accuracy of only 52.3%.[62] Therefore, with the applied technique, no subsequent improvement in predicting the direction of the USD/GBP exchange rate could be achieved. Table E.1.1 on page 96 shows the contingency table of the prediction for this first method.

B) Second method

The implicit assumption of the Naïve Bayes Classifier that some words occur more often in a specific kind of class may work for classifying emails as spam or non-spam (where words like "buy", "cheap" or "offer" occur more often in spam emails), but it might be problematic in classifying NAs as positive or negative, because the frequency of the occurrence of words like "raising", "growth" or "high" alone are not able to classify a NA as positive or negative. Rather the meaning of the words are contextual. Adjectives are depending on the object and the country they are referring to, so that for example a NA containing the words "uk *raising* unemployment" must be classified as negative, whereas a NA containing the words "uk *raising* interest rates" must be classified as positive. The word *raising* alone cannot classify NAs, so that a higher frequency of the words alone also cannot classify NAs (it is important if the unemployment or if the interest rate is rising). Additionally, a NA containing the words "*us* unemployment rising" and a NA containing the words "*uk* unemployment rising" also needed to be classified differently (here also the context, which country the NA is refering to, is important, leading to an opposite effect on the NA classification). For the following ***second method***, again ***1)*** the transformation of the NA`s into a term document matrix and ***2)*** the labeling of the NAs, as well as the final results are described.

1) Transformation of data: Therefore, to account for the context (country, noun, adjective) a second approach were applied, where the NAs were first separated into NAs with relevance to United States and NAs with relevance to the United Kingdom[63], leading again to 82.476 NAs in total. In a next step, all words but specific nouns and adjectives were removed from the NAs[64]. The remaining words in a NA, as well as the marker for the country (us or uk), were concatenated to one word, so that for example the NA "US unemployment was rising in June" is transformed to the concatenated artificial word

[62] Other examinations to measure the correlation between the NAs predicted label and the same (not the next) day also did not lead to significant results. Similar examinations with a different timely structure (1, 3, 6 and 12 hours) did also not lead to significant results.

[63] Similar to footnote 55, a NA was for example marked as relevant for the United States if one of the words in the list {usa, us dollar, fed, redbook, …} was present in the NA, while no word of the list {cad, jpy, Canada, ecb, euro, yen, …} was present in the NA.

[64] Only words in a list containing words like {cut, edges down, fall, contraction, rising, gain, …, manufacturing pmi, housing prices, consumer credit, …} have not been removed from the NAs.

"usunemploymentrising", so that each NA was represented by one concatenated word. Subsequently, similarly to the first approach, the minutely NAs for each day were merged to one NA per day, resulting in 1890 combined NAs (containing several concatenated artificial words). The combined NAs then were transformed to a term document matrix.

2) Labeling of NAs and results: To provide the labels for the training algorithm, the NAs were classified similarly to the first approach (NAs followed by positive (negative) changes in the exchange rate were classified as positive (negative) NAs). With the features (the term document matrix) and the corresponding labels, a Multinomial Naïve Bayes Classifier was trained with 1300 of the 1890 NAs. According to the probabilities of the trained classifier, the remaining 590 NAs were then classified as either positive or negative. In a final step, the 590 predictions were compared to the actual changes of the USD/GBP exchange rate (whereby a positively (negatively) classified NA was taken as a prediction that the subsequent change in the exchange rate would be positive (negative)). Out of this 590 predictions, only 283 were true predictions, leading to an accuracy of only 48.0%[65]. Therefore, also with the second technique applied, no improvement in predicting the direction of the USD/GBP exchange rate could be achieved. Table E.1.2 on page 96 shows the contingency table of the predictions for this second method.

3.3. Data mining and the exchange rate

For the following data mining approaches of the USD/GBP exchange rate, technical (3.3.1) as well as fundamental analyses (3.3.2) were applied.

3.3.1. Technical analysis

The technical analysis uses technical indicators to predict the USD/GBP exchange rate. Thereby the univariate approach in 3.3.1.1 only uses historical values, whereas the technical multivariate analysis in 3.3.1.2 uses further technical indicators, which were constructed out of the historical open, high, low and close prices as well as of the volume of the exchange rate.

3.3.1.1. Univariate analysis

In the ***technical univariate*** approach ***A)*** Trend Adjusted ES models, as well as ***B)*** ARIMA models were applied to fit the best model and forecast the USD/GBP exchange rate.

A) Trend Adjusted ES models

In order to test the performance of predictions of Trend Adjusted ES models, the USD/GBP exchange rate between 1989-01-03 and 2017-02-05 was downloaded from www.quandl.com in daily frequency

[65] Also for the second method other examinations to measure the correlation between the NAs predicted label and the same (not the next) day did not lead to significant results. Similar examinations with a different timely structure (1, 3, 6 and 12 hours) did also not lead to significant results.

(leading to 7170 observations). Out of these observations, the days 1 to 600 were taken to automatically fit a Trend Adjusted ES model as described in 2.1.2.1 (thereby the RMSE of this model was recorded). Subsequently, with the fitted model, a forecast of the exchange rate of day 601 was made. In a next loop, a Trend Adjusted ES model was fitted to the observations 2 to 601 and a forecast was made for the exchange rate of day 602. This rolling window procedure was repeated until the forecast for observation 7170 was made, so that 6570 Trend Adjusted ES models were fitted and 6570 forecasts were made. For these forecasts, the MASE was calculated to measure the performance of the predictions compared to naïve forecasts. Thereby, the benchmark of the naïve forecasts (a MASE of exactly 1) wasn´t beaten by the Trend Adjusted ES models – the MASE was calculated to 1.019, signifying that the Trend Adjusted ES models performed slightly worse than simple naïve forecasts. Additionally, out of the 6570 forecasts, only 3337 could predict the right sign of the changes of the exchange rate, leading to an accuracy of 51%. Therefore, with Trend Adjusted ES models no improvement in predicting the value or the direction of the USD/GBP exchange rate could be achieved. Table E.2.1 and Figure E.2.1 on page 97 show the contingency tables, the RMSE as well as the α and β values of the fitted Trend Adjusted ES models.

B) ARIMA models

In order to test the performance of the predictions of ARIMA models, the same time series and the same rolling window approach was applied, thereby fitting 6570 different ARIMA models and making 6570 daily predictions out of these models[66] (Hyndman, 2012, chap. 7 and 8). For the so calculated 6570 forecasts, the benchmark of naïve forecasts wasn´t beaten by the ARIMA models – the MASE was calculated to 1.024, signifying, that the ARIMA models performed slightly worse than simple naïve forecasts and also as the Trend Adjusted ES models. Out of the 6570 forecasts, only 3288 could predict the right sign of the changes of the exchange rate, leading to an accuracy of 50.0%. In Table E.2.2 the contingency table, and in Figure E.2.2 and Figure E.2.3 on pages 97f the RMSE over time, as well as a bar chart with all fitted ARIMA models are shown.

[66] Thereby, the fitting of the ARIMA models was achieved by a function from the R-package "Forecast". This so called auto.arima-function uses the Hyndman-Khandakar algorithm and tries to fit the best model according to the following rules:

1. Repeated KPSS tests are used to determine the order of differencing d
2. After differencing the data d times, the model with the smallest AIC is chosen from the following four models: ARIMA(2,d,2), ARIMA(0,d,0), ARIMA(1,d,0) and ARIMA(0,d,1), whereby if $d = 0$, then $c = nzc$, and when $d > 0$, then $c = zc$ (this is called the current model).
3. The current model will be varied in p and/or q by ± 1 and in including or excluding c (depending on the current model). If the AIC can be minimized, this model will be the current model.
4. Step 3. Is repeated until no lower AIC can be found.

3.3.1.2. Multivariate analysis

The multivariate technical analysis uses technical indicators generated out of the open, high, low, close and volume of the USD/GBP exchange rate. To get an idea of such technical indicators, some are described in Appendix C on pages 81ff. However, the following analysis uses more than the technical indicators described in Appendix C (a list of all used technical indicator can be found in Table E.3.1 on page 99). The open, high, low, close price and the volume of the USD/GBP exchange rate from 1986-02-10 to 2016-10-29 was downloaded from www.ducascopy.com in daily frequency and the indicators have been calculated using the R-package "TTR" (leading to 8537 observations and 22 technical indicators as features). For this multivariate technical analysis the response variable were transformed into a dummy variable with values 0 and 1 (0 if the daily change in the exchange rate was negative, and 1 if the daily change in the exchange rate was positive). Therefore, the task for the following two machine learning algorithms (SVM and NN) was to predict the direction (i.e. the right sign) of the changes of the USD/GBP exchange rate (thus, these are classification tasks).

A) SVM classification

With the described data set, the same rolling window approach as in 3.3.1.1 was applied whereby a radial kernel with $\gamma = 0.1$ a cost C of 1 have been chosen. Because of the classification task, no MASE was calculated. Out of the 7937 forecasts only 3972 could predict the right class (positive or negative) of the changes of the exchange rate, leading to an accuracy of 50.0%. Table E.3.2 shows the contingency table of the SVM predictions and Figure E.3.1 (both on page 99) shows the accuracy of all fitted models over time.

B) NN classification

For the NN, the same rolling window approach as for the SVM classification was applied, leading again to 7937 models and forecasts. Thereby, the model parameter of the fitted NN were chosen as follows: the network architecture were two hidden layers with 20 and 10 neurons per layer. A hyperbolic tangent activation function as well as a learning rate of 0.3 were chosen. Out of the 7937 forecasts, only 4041 could predict the right sign of the changes of the exchange rate, leading to an accuracy of 50.9%. Table E.3.3 shows the contingency table and Figure E.3.2 shows the accuracy of the fitted NNs over time (both on page 100).

3.3.2. Fundamental analysis

For the following fundamental analysis, ***three different techniques*** have been applied, where in ***A)***, VAR/VEC models were fitted to predict the *values* and in ***B)*** and ***C)*** SVMs as well as NN`s have been fitted to predict the right *sign* of the USD/GBP exchange rate. The collection of the data sets was based on the most common theoretical fundamental explanations for the determination of exchange rates (as

described in Appendix D on pages 90ff). Because these fundamental explanations very often use quarterly or yearly national economic data, only some were incorporated in making daily predictions of the USD/GBP exchange rate. In order to obtain more features in daily frequency, other financial time series (like other exchange rates and global stock indices) were used as additional features. Thereby, the predictions of the values (part A) and the predictions of the signs (part B and C) use different data sets as described in each part.

A) VAR/VEC models

In order to test the performance of predictions of VAR/VECM models with regard to the USD/GBP, the two strongest short time influences assumed on the exchange rates, i.e. the interest rates of both countries, as well as other exchange rates (to obtain potential co-integration relationships) were chosen as features. Therefore, the 2 month LIBOR based on USD and the 3 month LIBOR based on GBP, as well as the USD/GBP, CAD/GBP and CHF/GBP exchange rates between 1987-01-02 and 2016-11-07 were downloaded from www.quandl.com in daily frequency (leading to 6935 observations). Because for VAR and VEM models, the order of integration should be the same for all variables, ADF tests were applied to each feature following the approach of Pfaff (Pfaff, 2008, p. 92f), thereby assuming, that the test results for the complete time span are representative for the sub samples of the sliding window approach (i.e. the test was only applied once, and then it was assumed, that the results were valid for all sub periods of the sliding window approach). The final test results were similar for all variables so that they are explained here only for the USD/GBP exchange rate (all test statistics are however shown in Table E.4.1 on page 100). The tests were conducted as follows: As a first step, a regression with a constant, trend and lagged time series (coefficients β_1, β_2, π) has been estimated. Next, the hypothesis, that the trend and the lagged values are zero, i.e. $H_0: (\beta_1, \beta_2, \pi) = (\beta_1, 0, 0)$ was tested with a F type test. The test statistics for the USD/GBP exchange rate has a value of 2.45 with a corresponding 1% critical value of 8.27. Therefore, the null hypothesis, i.e. a unit root cannot be rejected. Next, it is tested, whether the USD/GBP exchange rate is a random walk with or without drift. The test statistic for the hypothesis, that the time series contains no drift, i.e. $H_0: (\beta_1, \beta_2, \pi) = (0, \beta_1, \pi)$, has a value of 2.05. Because of the 1% critical value of 6.09, this null hypothesis cannot be rejected, which implies, that the USD/GBP is a random walk with no drift. To verify these results, in a final step a regression with a constant only has been estimated and it is tested, whether the drift term is absent or not (i.e. $H_0: \beta_1 = 0$ is tested). With a test statistic for the USD/GBP exchange rate of 0.71 and a 1% critical value of 6.43 the null hypothesis cannot be rejected. Therefore, the USD/GBP (as well as the other variables) is considered to contain a unit root, but neither a linear trend nor a drift.

For the VAR/VECM approach the same sliding window approach is applied as before. In order to decide, whether a VAR or a VECM model should be fitted, the Johansen eigenvalue test is executed as

described in 2.1.2.4. The test is applied every 100th model building process. Therefore, it is assumed, that after the Johansen test was applied, the measured co-integration relationships stays constant for the next 100 model building processes[67]. To choose the lag order of the VAR/VEC models, the lag order with the minimal AIC in a range between 1 and 5 lags were automatically chosen. For the so calculated 6570 forecasts, the MASE was calculated to measure the performance of the predictions compared to naïve forecasts. Thereby, the benchmark of naïve forecasts wasn´t beaten by the VAR/VEC models where the MASE was calculated to 1.024, signifying, that the VAR/VECM models performed slightly worse than simple naïve forecasts and also as the Trend Adjusted ES models. Out of the 6935 forecasts, only 3487 could predict the right sign of the changes of the exchange rate, leading to an accuracy of 50.3%. In Table E.4.2 the contingency table, in Figure E.4.1, in Figure E.4.2 and in Figure E.4.3 the RMSE over time, the chosen lag orders as well as the chosen models (VAR vs VEC) are shown (on pages 101f).

B) SVM classification

The fundamental analysis uses the USD/GBP as dummy variable (positive, negative changes) as response as well as 498 features[68] for the period between 2001-01-23 and 2016-11-01.

With this features and responses the same rolling window approach as in 3.3.1.1 was applied to fit SVM models, whereby a radial kernel with an γ of 0.1 and a cost of 1 were chosen. Because of the classification task no MASE was calculated. Out of the 2581 forecasts only 1311 could predict the right sign (positive or negative) of the changes of the exchange rate, leading to an accuracy of 50.8%. Table E.4.4 shows the contingency table and Figure E.4.4 the in sample accuracy of the fitted SVM`s over time (on pages 103).

C) NN classification

For the NN, the same rolling window approach as for the SVM classification was applied, leading again to 2581 models and forecasts. Thereby, the model parameter of the fitted NNs were chosen as follows: the network architecture were three hidden layers with 50, 30 and 10 neurons per layer. A hyperbolic tangent activation function as well as a learning rate of 0.8 were chosen. Out of the 2581 forecasts, only 1344 could predict the right sign of the changes of the exchange rate, leading to an accuracy of 52.1%. Table E.4.5 show the contingency table and Figure E.4.5 the in sample accuracy of the fitted NNs over time (on page 103f).

[67] This means for example, that the Johansen test was applied for the time window of observation 1 to 600. If the test signals co-integration relationships, then a VEC model is fitted for the windows 1 to 600, 2 to 601, …, up to the window with observations 100 to 699. Then, the Johansen test is repeated for the window 101 to 700 and the resulting model (VAR or VEC) is fitted to the next 100 model building processes.

[68] The 498 features are described in Table E.4.3 on page 102. To construct these features, 57 time series have been downloaded from www.quandl.com and https://de.finance.yahoo.com/ in daily frequency. For 49 of the 57 features 10 differences have been calculated for each time series, such that the first differences (for variable one) have been calculated from the observations at time t and time t-1, the second differences have been calculated from the observations at time t and t-2 and so forth (up to the 10th differences). This has been done for 49 of the 57 time series. Therefore, the fundamental analysis uses 49*10 differences plus 8 time series (normal) = 49*10+8 = 498 features.

4. Conclusion

In this thesis, data and text mining approaches have been described analyzing classical time series, as well as modern ML techniques. Regarding the classical time series techniques, it was shown how Trend Adjusted ES models use exponentially smoothed levels, as well as exponentially smoothed trend components in order to fit models and make predictions. Next, the widely used ARIMA models were described in an intuitive, as well as in a more formal and mathematical way. ARIMA models combine the concept of integration with ARMA models. ARMA models use AR parts of the time series itself, as well as MA parts of the estimated error terms (plus possible deterministic trends) in order to fit models and make predictions. While the subsequently described VAR models only use AR components (and no MA parts), the AR parts are multivariate (in contrast to the univariate ARIMA approach), so that the model fits, as well as the predictions are obtained by AR parts of multiple time series (plus possible deterministic trends of these time series). VEC models further extend VAR models in the sense that they additionally correct for deviations from common trends of two or more time series (by modelling the speed of adjustment of the time series to one or more co-integration relationships). Regarding the ML techniques, first a Multinomial Naïve Bayes Classifier was introduced and it was shown how this classifier can be used to train (or learn) how to separate different texts into different categories. In a next part, the ideas and mathematical formulations of SVMs were outlined and it was described how with this technique, data can be divided into two or more groups using separating hyperplanes and support vectors. The last ML technique showed how a network of neurons can be interpreted as a complex decision making unit and how the training (or learning) process is executed by changing the weights and biases of the network in order to minimize the cost (in this way the fit of the model is improved successively).

In applying these techniques to the USD/GBP exchange rate, none of them were able to substantially improve the prediction performance for daily forecasts executed in rolling window approaches (when the values were predicted in regression tasks, no model could beat the benchmark of naïve forecasts and when the sign was predicted in classification tasks, no method could reach more than 53% in accuracy)[69].

There are ***five possible reasons, why no proper forecasts*** were obtained: ***a)*** the USD/GBP exchange rate follows an unpredictable random walk (the best prediction would then be a naïve forecast). In this sense,

[69] For the classical time series techniques this means, that the Trend Adjusted ES models (where the forecast was made up out of a weighted average plus a trend component), the ARIMA models (where the forecast was made up out of the lagged AR and MA parts plus the concept of integration), the multidimensional VAR models (where the forecast was made up out of multidimensional lagged AR and MA parts) as well as the VEC models (where the forecast was made up out of a VAR model plus a correcting term for co-integration relationships) could not produce proper forecasts in predicting the right values for the USD/GBP exchange rate. For the ML techniques, the Multinomial Naïve Bayes Classifier could not relate NAs to the changes of the USD/GBP exchange rate (what would be a sign against the efficient market hypothesis) and the classification tasks using SVMs and NNs could also not produce proper forecasts in predicting the right sign of the USD/GBP exchange rate.

there would be no patterns to detect. ***b)*** If there should be patterns in exchange rates, the available data base in daily frequency may not be sufficient. There could be complex patterns which evolve only within a day or stay for just some days. Such patterns, if present, could only be detected with a higher frequency (like hourly or even minutely data[70]). ***c)*** The time window of 600 days was chosen more or less arbitrarily. Should there be patterns in the USD/GBP exchange rate, then these patterns could be constant different time windows. The time window of 600 days (around two years) was chosen as a compromise between the goal to reach enough daily data to build complex models on the one hand and the goal to reach a time window which is not too big (so that a pattern could be constant in this period) on the other hand. Additionally, the window is changing each step by only one day, such that a lot of predictions have been made by very similar features. Nevertheless, this approach was applied to reach enough forecasts to verify the validity of the prediction performance of the applied models. ***d)*** With regard to the text mining analysis, the results could also be due to a wrong timing, meaning that in an efficient foreign exchange market – the very fluently traded USD/GBP exchange rate could be assumed as such – new information would be immediately captured and incorporated into the prices. Even more problematic could be the reverse labeling of the NAs as positive or negative depending on the subsequent price change of the USD/GBP exchange rate. All the positive and negative intraday messages of one day were labeled the same, although there are very different kind of information with a presumably different strength and duration of the impact on the exchange rate. ***e)*** Due to the sliding window approach, the hyperparameter of the different models needed to be chosen automatically[71] or in advance[72] for all models, whereby these parameters are best chosen individually due to the peculiarities of the underlying data set.

[70] For the handwritten digits recognition competition, which reached very high accuracy levels of up to 99.9%, 765 features were used with a training data set of around 50.000 pictures containing handwritten digits. Therefore, for the task to classify the digits 0 to 9, a matrix of 765 x 50.000 was necessary to train the classifiers. In this thesis, a feature vector of no more than 500 features were trained with a training data set of around 600 days. Whereby the matrix was with the dimensions 500 x 600 a lot smaller than the matrix to classify the digits from 0 to 9, the task for predicting the USD/GBP exchange rate is tremendously more complex (with a huge amount of global political as well as economical factors influencing the exchange rate).

[71] Like in the case of the Hyndman-Khandakar algorithm.

[72] Like the γ parameter or the cost for the SVMs or the network architecture, the learning rate or the cost function for the NNs.

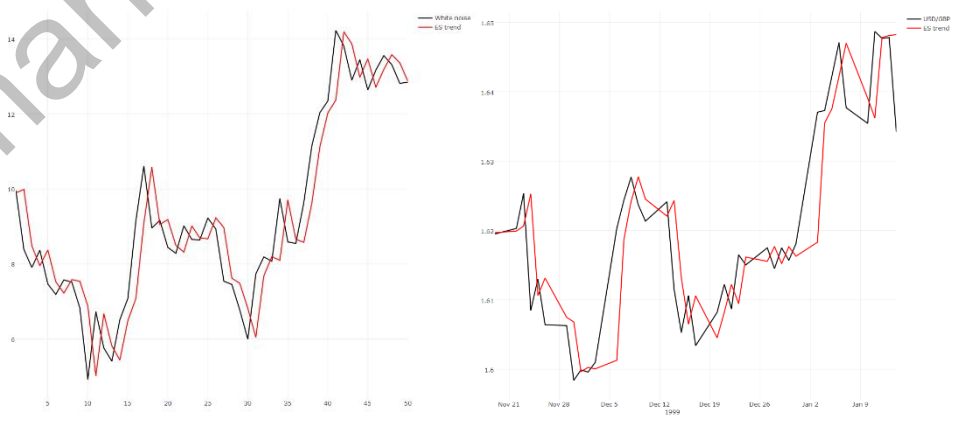
Appendix A: Classical time series techniques

A.1 Trend adjusted ES models

Figure A.1.1: 50 observations for a white noise process and a time series with a quadratic trend (each in black) as well as the correspondent trend adjusted exponentially smoothed time series (in red)

Annotation: The left figure shows a white noise process around the mean 5 for 50 observations and the corresponding trend adjusted exponentially smoothed time series in red. For the smoothed time series (also called the level) an α of 0.017 and a β of 0.017 was estimated, meaning, that current levels are almost not affected by current observations (so that the level stays quite constant) and that also the current trend is almost not affected by current level changes (so that the level is changing slowly). Thereby, a fast or slow change in the trend does not mean, that the trend adjustment is strong or weak. The right figure shows a stochastic time series around a quadratic trend in black and the correspondent trend adjusted exponentially smoothed time series in red. The levels have been estimated with an α of 0.25 and a β of 0.13. Therefore, current observations have a bigger impact on the current level and the trend is changing faster than for the white noise process on the left. *Source: Own simulation*

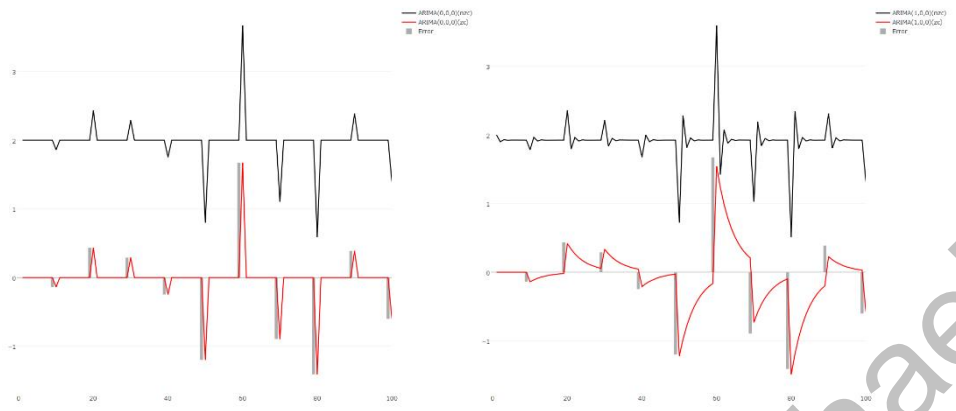
Figure A.1.2: 50 observations for a random walk process and the USD/GBP exchange rate between 1999-11-19 and 2000-01-26 (each in black) as well as the correspondent trend adjusted exponentially smoothed time series (in red)



Annotation: The left figure shows a random walk process with a starting value of 10 in black as well as the corresponding trend adjusted exponentially smoothed time series in red. For the levels an α of 0.96 and for the trends a β of 0.00 was estimated, meaning, that the one step ahead forecast of the levels are almost completely equal to the last observation and there is no trend adjustment. On the right side, the USD/GBP exchange rate between 1999-11-19 and 2000-01-26 as well as the corresponding trend adjusted exponentially smoothed time series in red is shown. For the levels an α of 0.90 and for the trends a β of 0.00 was estimated, meaning that the levels at time t are almost completely made up of the levels at time t-1 and the trend is not changing at all (just like for a random walk). *Source: Own simulation*

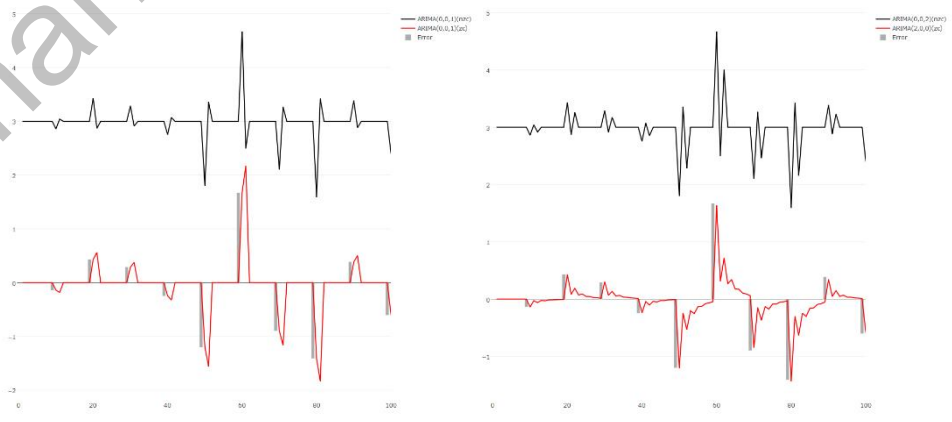
A.2 ARIMA models

Figure A.2.1: ARIMA(0,0,0)(zc) (red left) and ARIMA(0,0,0)(nzc) (black left) as well as ARIMA(1,0,0)(zc) (red right) and ARIMA(1,0,0)(nzc) (black right) with the according error terms (grey bars)



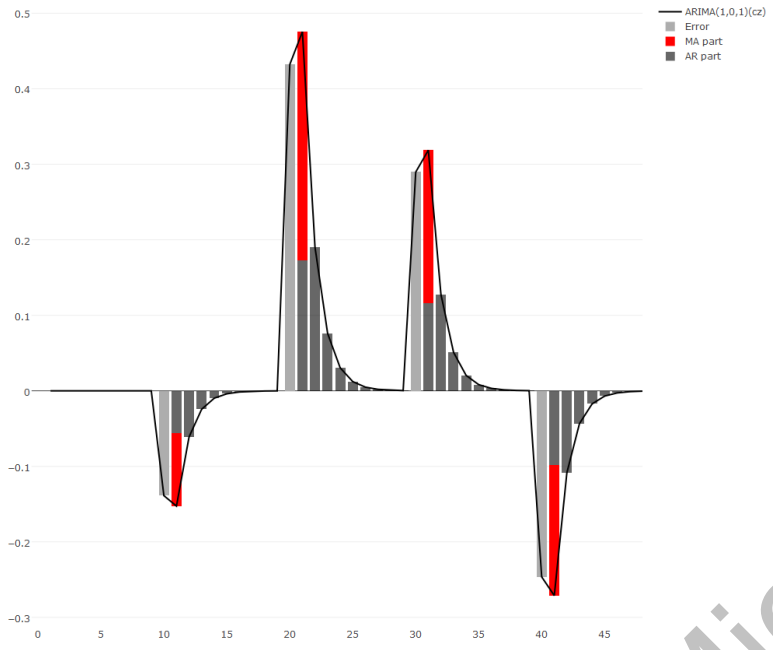
Annotation: The left figure shows an ARIMA(0,0,0)(zc) with $c = 0$ in red and an ARIMA(0,0,0)(nzc) with $c = 2$ in black. In grey, the corresponding error terms are shown. The error terms have been generated in this way, that it only differs from zero for every 10th observation. The ARIMA(0,0,0)s are just white noise processes. They are mean stable with an immediate reversion to the mean. The value of c just shifts the mean of the process up or down horizontally (here $c = 2$ leads to a mean $\mu = 2$ for the left time series in black). The right figure shows an ARIMA(1,0,0)(zc) where $c = 0$ and $\alpha = 0.8$ in red and an ARIMA(1,0,0)(nzc) with $c = 2.5$ and $\alpha = -0.3$ in black. ARIMA(1,0,0)s are just AR(1) processes with reversion to the mean depending on α. For series with higher and positive αs, like $\alpha = 0.8$ for the time series in red, the impact is declining and the reversion to the mean is slow. For series with smaller and negative αs, like $\alpha = -0.3$ for the time series in black, the impact is oscillating and the reversion to the mean is faster. The constant c again just shifts the mean of the process up or down horizontally (here $c = 2.5$ leads to a mean $\mu = \frac{c}{1-\alpha} = \frac{2.5}{1.3} = 1.92$ for the time series in black). *Source: Own simulation*

Figure A.2.2: ARIMA(0,0,1)(zc) (red left) and ARIMA(0,0,1)(nzc) (black left) as well as ARIMA(2,0,0)(zc) (red right) and ARIMA(0,0,2)(nzc) (black right) with the according error terms (grey bars)



Annotation: The left figure shows an ARIMA(0,0,1)(zc) with $c = 0$ and $\beta = 1.3$ (in red). In this case, the effect of the error term was enhanced for the subsequent period, but in contrast to AR processes, the effect only persisted for one more subsequent period. In general, the effect of an error term for MA(q) models will only last q subsequent periods. The black time series on the left corresponds to an ARIMA(0,0,1)(nzc) with $c = 3$ and $\beta = -0.3$. In this case, the sign of the effect was reversed and the strength was dampened (while still, the mean reversion was completed one subsequent period after the error occurred). The figure on the right shows an ARIMA(2,0,0)(zc) with $c = 0$, $\alpha_1 = 0.2$ and $\alpha_2 = 0.2$ (in red) where the mean reversion takes on a more complex form while the series is converging to the mean. In black an ARIMA(0,0,2)(nzc) with $c = 3$, $\beta_1 = -0.3$ and $\beta_2 = 0.6$ is shown. The effect of the error term lasted for two subsequent periods, while after that, it was completely gone. *Source: Own simulation*

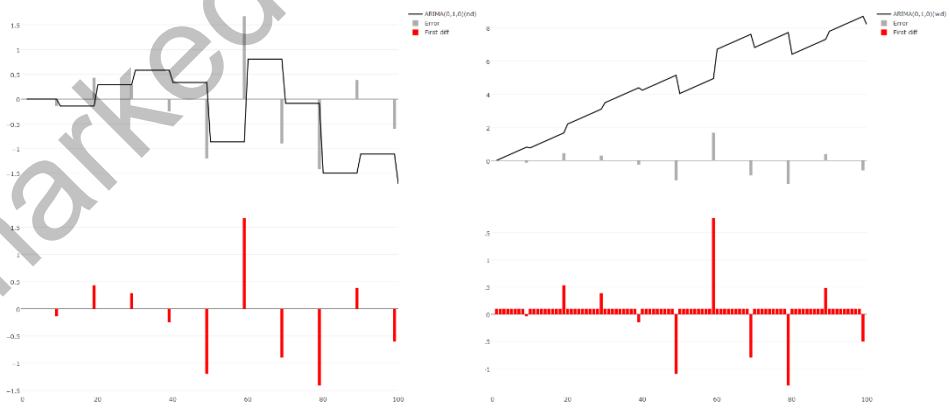
Figure A.2.3: ARIMA(1,0,1)(zc) (black line) with the corresponding AR (black bars) and MA parts (red bars) and with the corresponding error terms (grey bars)



Annotation: The figure shows an ARIMA(1,0,1)(zc) in black. Thereby an AR and a MA process have been combined. For stationary series, the subsequent mean reversion of the process depends on the AR as well as the MA part. While the MA part only contributes to the levels of the time series one subsequent period after an error occurred, the AR contributions are theoretically infinite (but converging to zero depending on α). The process was simulated with $\alpha = 0.4$ and $\beta = 0.7$. Therefore, the subsequent level after an error occurred is made up of 40% of the last level (AR part, black bars) plus 70% of the last error (MA part, red bars). The MA part only contributes to q periods after the error occurred (here q equals 1).

Source: Own illustration

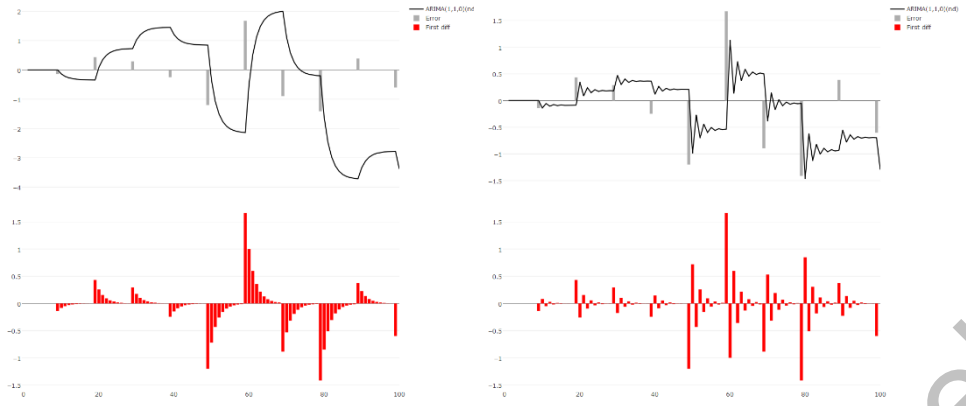
Figure A.2.4: ARIMA(0,1,0)(nd) on the left and ARIMA(0,1,0)(wd) on the right (each in black), each with the correspondent error terms (grey bars) and the first differences (red bars)



Annotation: The left figure shows an ARIMA(0,1,0)(nd) (in black). Thereby, the time series is first differenced and subsequently an ARMA(0,0)(zc) is applied. This random walk with no drift only follows a stochastic trend. The first differences (in red) follow a white noise process around zero. On the right figure, an ARIMA(0,1,0)(wd) (in black) is displayed. This random walk with drift follows a stochastic as well as deterministic trend because the first differences are white noise with a non-zero constant c. Therefore, each period, the series is growing with the slope c. Around this slope, the series follows a stochastic trend.

Source: Own simulation

Figure A.2.5: ARIMA(1,1,0)(nd) with $\alpha = 0.6$ on the left and ARIMA(1,1,0)(nd) with $\alpha = -0.6$ on the right (each in black), each with the correspondent error terms (grey bars) and the first differences (red bars)



Annotation: The left figure shows an ARIMA(1,1,0)(nd) with $c = 0$ and $\alpha = 0.6$. Through differencing, the non-stationary process is converted into stationary first differences in red. The first differences then follow an AR(1)(zc) process with $\alpha = 0.6$. This displays, that changes caused by an error term produces further changes in the same direction but with declining strength, leading to a stronger perpetuation of the original error (and a non-oscillating series). Thereby, the subsequent changes are converging to zero, because $c = 0$. The right figure shows an ARIMA(1,1,0)(nd) with $c = 0$ and $\alpha = -0.6$. In this case, changes caused by the error term produces further changes with decreasing strength in the opposite direction, leading to a weaker perpetuation of the original error (and a oscillating series). *Source: Own simulation*

Figure A.2.6: ARIMA(1,1,0)(wd) on the left and ARIMA(0,1,1)(wd) on the right (both in black), each with the correspondent error terms (grey bars) and their first differences (red bars)

Annotation: The left figure shows an ARIMA(1,1,0)(wd) with $c = 0.075$ and $\alpha = -0.6$. Through the non-zero constant in the first differences, the first differences are converging to the mean $\mu = \frac{c}{1-\alpha}$, leading to a drift in the original series. Because of this, the time series is twisted counter-clockwise compared to the same series with no drift (ARIMA(1,1,0)(nd) in Figure A.2.5 on the right side). The figure on the right shows an ARIMA(0,1,1)(wd) with $c = 0.075$ and $\alpha = -0.6$, which is just an twisted ARIMA(0,1,1)(nd). Here, the first differences are following a MA(1) process with mean $c = 0.075$. For a MA(1) process in differences, the first differences are not converging to the mean (like for a AR process), but they are breaking up and are equal to the mean in the second period after an error occurred. *Source: Own simulation*

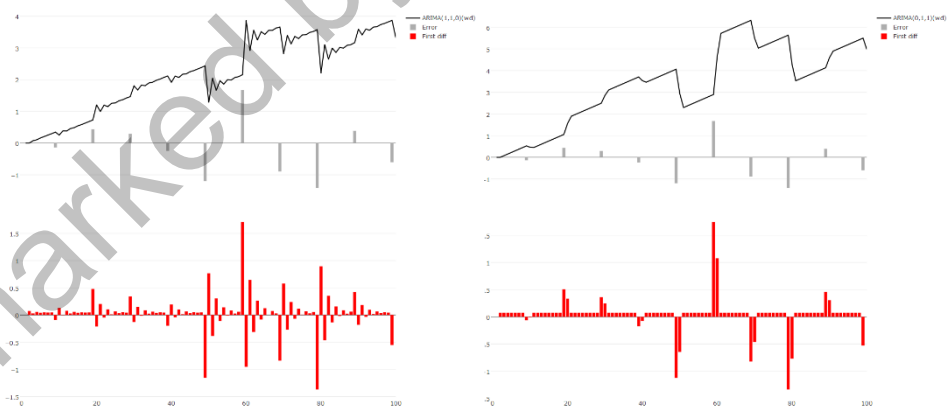


Figure A.2.7: ARIMA(1,1,1)(nd) (black line) with correspondent error terms (grey bars), AR part (black bars), MA part (red bars) and first differences (red line)

Annotation: The right figure shows an ARIMA(1,1,1)(nd) with $c = 0$, $\alpha = 0.4$ and $\beta = 0.3$ (black line). In this case, the differenced time series in red equals an ARMA(1,01)(zc) with a reversion to the zero mean depending on p and q. Thereby, the subsequent changes after an error occurred are made up of 40% of the last change (AR part, black bars) plus 30% of the last error (MA part, red bars). The MA part only contributes to q periods after the error occurred (here q equals 1). Depending on the values of α and β, the effect of an error to the original series can be enhanced and non-oscillating or decreased and oscillating (as described in Figure A.2.5).

Source: Own simulation

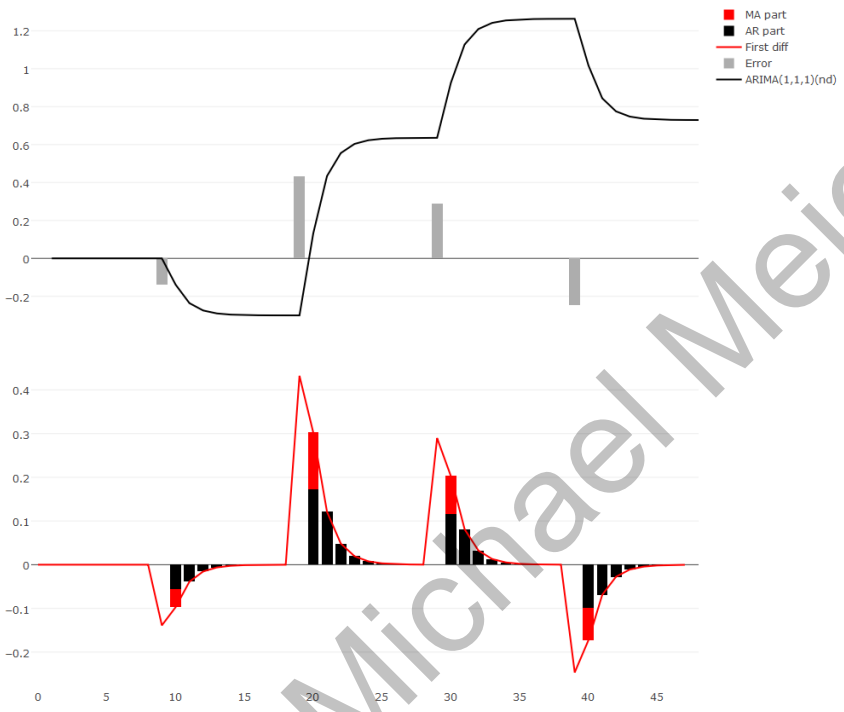


Figure A.2.8: ARIMA(0,0,0)(lt) on the left and ARIMA(1,0,0)(lt) on the right (both in black), each with the correspondent error terms (grey bars) and their de-trended time series (red)

Annotation: On the left, an ARIMA(0,0,0)(lt) is shown in black with the correspondent de-trended time series in red and the error terms in grey. De-trending transforms the non-stationary time series (with a time-dependent mean) into a stationary white noise process around a zero mean. Thereby, de-trended time series also could equal a time series with AR and/or MA components. This is shown on the right figure, where an ARIMA(1,0,0)(lt) is show. For this process, the de trended time series displays a AR(1) with an oscillating reversion to the mean.

Source: Own simulation

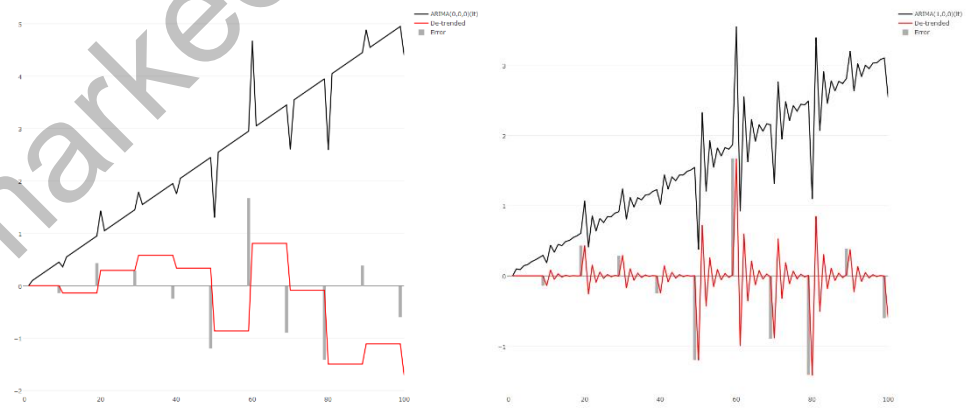
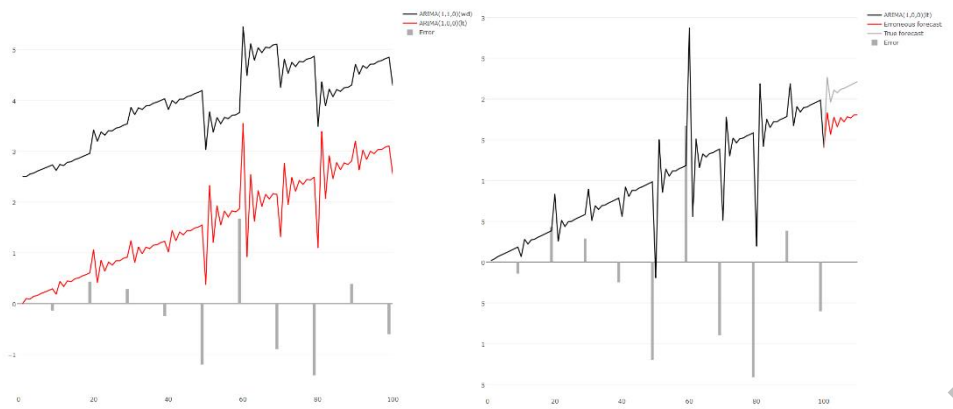


Figure A.2.9: ARIMA(1,0,0)(lt) (in red) and ARIMA(1,1,0)(wd) (in black) on the left, and ARIMA(1,0,0)(lt) (in black) with an erroneous (red) and an right forecast (grey) on the right

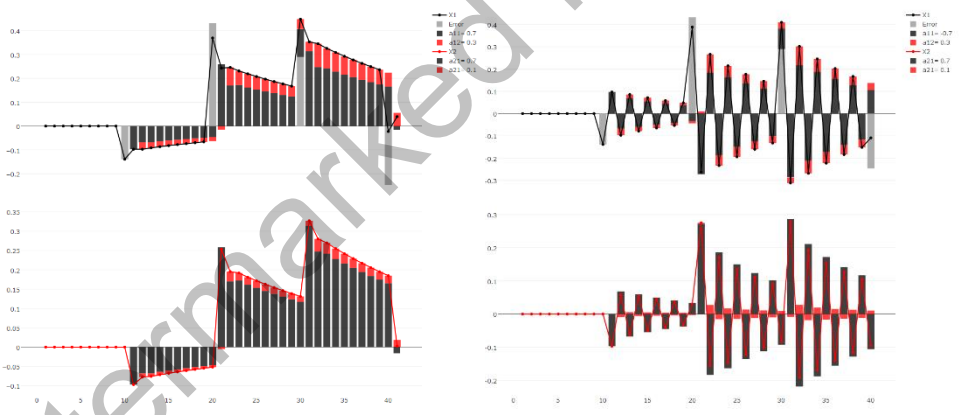


Annotation: The left figure shows an ARIMA(1,0,0)(lt) with a linear trend $lt = 3.5 + 0.05t$ (in red) and an ARIMA(1,1,0)(wd) with $c = 0.05$ and $\alpha = -0.6$ (in black). The time series with the linear trend shows no materialization of occurring error terms on the long run, while the time series with a drift materializes occurring errors to some extend on the long run. On the right side, an ARIMA(1,0,0,)(lt) is shown with an erroneous forecast (in red) and a true forecast (in grey). The erroneous forecast results from a fit with an ARIMA(1,1,0)(wd) to the time series with linear trend. The true forecast results from first de-trending the time series and then applying an AR(1) model to the de-trended time series.

Source: Own simulation

A.3 VAR Models

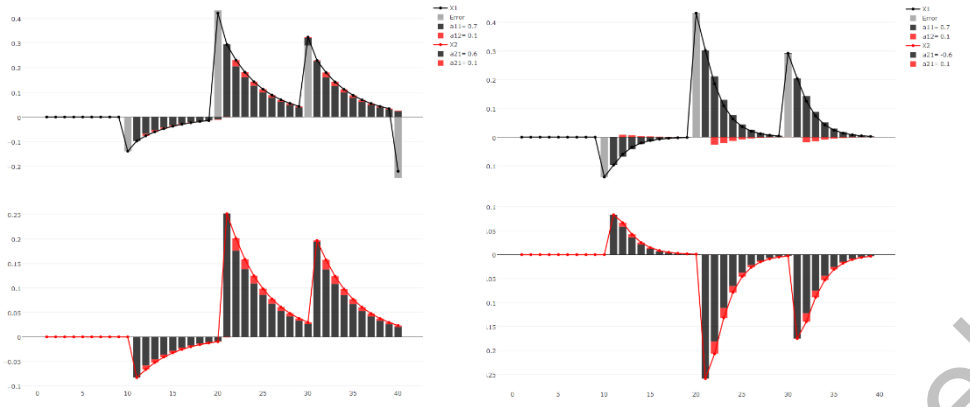
Figure A.3.1: VAR(1) with $\alpha_{11} = 0.7, \alpha_{12} = 0.3, \alpha_{21} = 0.7, \alpha_{22} = 0.1$ on the left side and VAR(1) with $\alpha_{11} = -0.7, \alpha_{12} = 0.3, \alpha_{21} = 0.7, \alpha_{22} = 0.1$ on the right, both with x_1 and its auto- and cross-regressive parts in black and with x_2 and its auto- and cross-regressive parts in red



Annotation: For illustrative reasons, the error terms (grey bars) have been constructed in this way, that e_1 is different from zero for every 10th observation while e_2 is zero for all t. On the left side a VAR(1) is shown, where α_{11} is big compared to α_{12} (0.7 vs 0.3). Therefore, the subsequent values of x_1 are more dependent on its own lagged values (black bars on the top left figure) than on the lagged values of x_2 (red bars on the top left figure). In contrast, for x_2 (line in red), α_{22} is small compared to α_{21} (0.7 vs 0.1). In this case, x_2 is more dependent on the lagged values of x_1 (black bars on the bottom left figure) than on its own lagged values (red bars on the bottom left figure). In this sense, x_2 is driven by x_1. On the right side, the same VAR(1) is shown, just that $\alpha_{11} = -0.7$ instead of $\alpha_{11} = 0.7$. Because x_1 is more dependent on its own lagged values and because α_{11} is smaller than zero, the mean reversion of the process is alternating. The mean reversion of x_2 is also alternating because x_2 is more dependent on the (alternating) lagged values of x_1 (x_2 is still driven by x_1).

Source: Own simulation

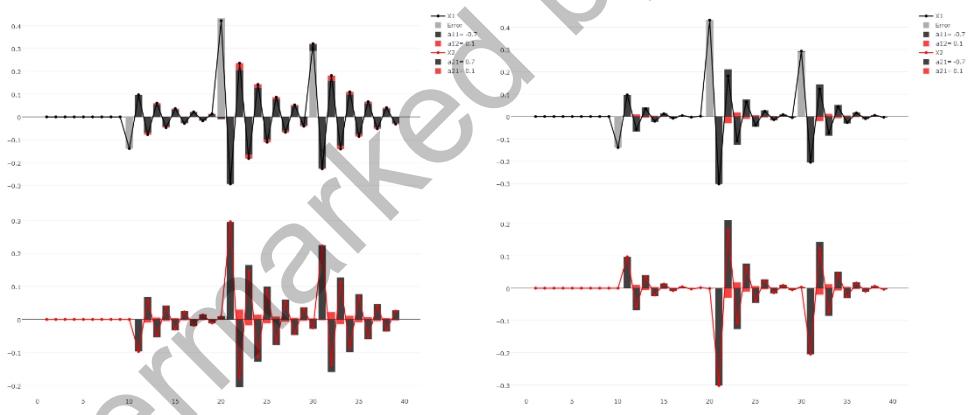
Figure A.3.2: VAR(1) with $\alpha_{11} = 0.5, \alpha_{12} = 0.1, \alpha_{21} = 0.5, \alpha_{22} = 0.1$ on the left side and VAR(1) with $\alpha_{11} = 0.7, \alpha_{12} = 0.1, \alpha_{21} = -0.6, \alpha_{22} = 0.1$ on the right, both with x_1 and its auto- and cross-regressive parts in black and with x_2 and its auto- and cross-regressive parts in red



Annotation: On the left side a VAR(1) is shown, where α_{11} is big compared to α_{12} as well as α_{21} is big compared to α_{22} (what means, that x_1 is almost an simple AR(1) and x_2 is driven by x_1). On the same time, α_{11} and α_{21} are positive, meaning that a positive shock has a positive effect on both x_1 and x_2. This leads to a correlation of 0.71 between x_1 and x_2. On the right side a VAR(1) is shown where again α_{11} and α_{21} are big compared to α_{12} and α_{22}, but this time with a negative α_{21}. Therefore, a shock e_t will have reverse effects on the levels of x_2, leading to a negative correlation of -0.64.

Source: Own simulation

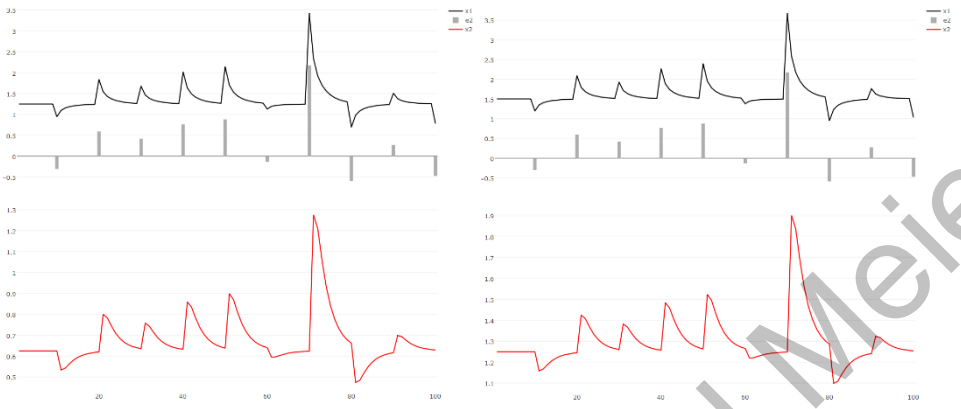
Figure A.3.3: VAR(1) with $\alpha_{11} = -0.7, \alpha_{12} = 0.1, \alpha_{21} = 0.7, \alpha_{22} = 0.1$ on the left side and VAR(1) with $\alpha_{11} = -0.7, \alpha_{12} = 0.1, \alpha_{21} = -0.7, \alpha_{22} = 0.1$ on the right, both with x_1 and its auto- and cross-regressive parts in black and with x_2 and its auto- and cross-regressive parts in red



Annotation: On the left side a VAR(1) is shown, where $|\alpha_{11}|$ is big compared to $|\alpha_{12}|$ and where $|\alpha_{21}|$ is big compared to $|\alpha_{22}|$ (x_2 is following x_1). As opposed to the example in Figure A.3.2 α_{11} is now negative (making the mean reversion of x_1 alternating). Because α_{21} is positive, a change of x_1 caused by an error will provoke a change in x_2 in the same direction one period after the error occurred. This leads to a timely contrarian alternation of x_1 and x_2 and to a negative correlation. On the right figure, the same process is shown besides that $\alpha_{21}$is negative, leading to a timely contemporaneous alternation and a positive correlation.

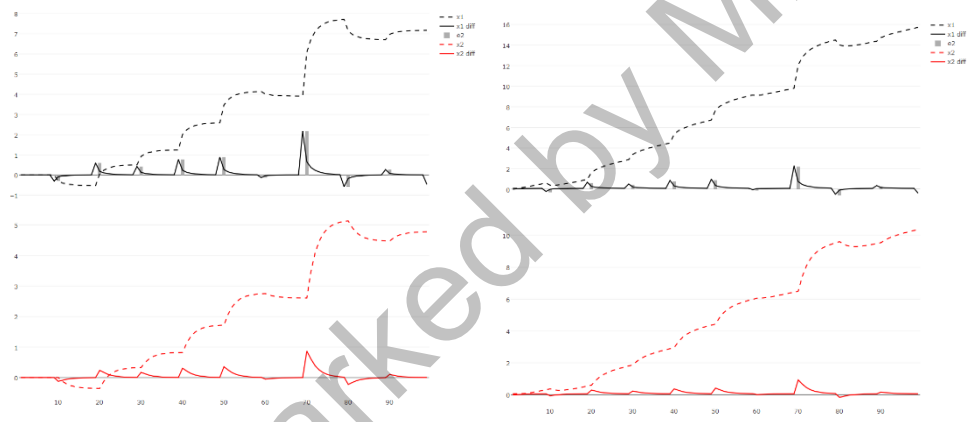
Source: Own simulation

Figure A.3.4: Two VAR(1) with $\alpha_{11} = 0.5, \alpha_{12} = 0.2, \alpha_{21} = 0.3, \alpha_{22} = 0.4$ where $c_1 = 0.5$ and $c_2 = 0$ on the left side and where $c_1 = 0.5$ and $c_2 = 0.3$ on the right side



Annotation: On the left figure a VAR(1) is shown where only one constant differs from zero ($c_1 = 0.5$ and $c_2 = 0$). Thereby, this is sufficient to shift both time series upwards (because the level of $\boldsymbol{x_2}$ also depends on the (shifted) level of $\boldsymbol{x_1}$). On the right side the same VAR(1) is shown but now also the constant of the time series $\boldsymbol{x_2}$ is different from zero ($c_1 = 0.5$ and $c_2 = 0.3$). This again not only shifts $\boldsymbol{x_2}$ but also x_1 up even further. *Source: Own simulation*

Figure A.3.5: VAR(1) in differences with $\alpha_{11} = 0.3, \alpha_{12} = 0.2, \alpha_{21} = 0.4, \alpha_{22} = 0.4$ for the differenced time series (where the levels contained a random walk component without drift on the left and with drift on the right)



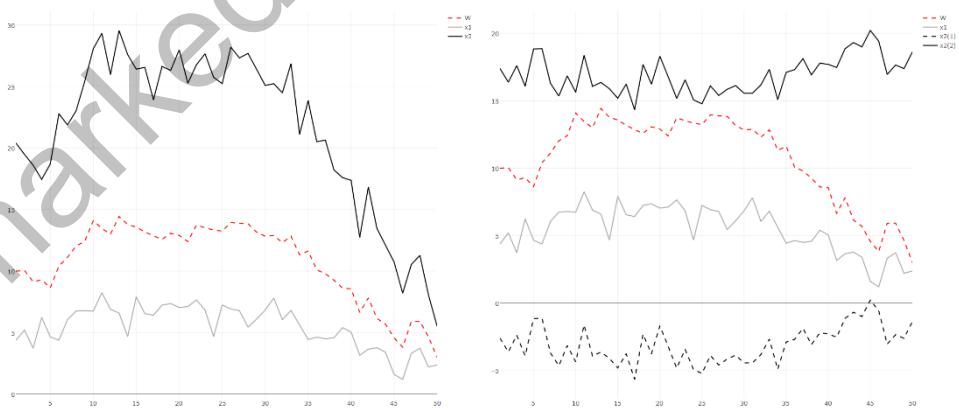
Annotation: On the left figure a VAR(1) in differences is shown (solid lines) where the original time series (dashed lines) have been differenced one time to remove the random walk component. Because the random walk component did not contain a drift, the differenced time series are reverting back to a mean equal to zero. On contrary, the right figure shows a VAR(1) in differences, where the original time series contained a random walk with drift. Therefore, the differenced time series are reverting to a slightly positive mean. *Source: Own simulation*

Figure A.3.6: Two VAR(1) processes with linear trends, whereby on the left, both trends emerge from the trend of only one time series, and on the right, both trends emerge due to trends in both time series (when considered univariate)

Annotation: On the left figure a VAR(1) in levels with $\alpha_{11} = 0.5, \alpha_{12} = 0.3, \alpha_{21} = 0.2, \alpha_{22} = 0.4$ and with $c_1 = 0.05t$ and $c_2 = 0t$ is shown. Simulated like this, only $\boldsymbol{x}_1$ contains a linear trend. Similar to the arguments of Figure A.3.4, this also provokes a linear trend in the levels of $\boldsymbol{x}_2$. The VAR(1) on the right side is simulated analogously to the VAR on the left, but this time with $c_1 = 0.05t$ and $c_2 = -0.033t$. The negative trend of $\boldsymbol{x}_2$ counteracts the trend of $\boldsymbol{x}_1$ such that the trend of $\boldsymbol{x}_2$ gets negative and the trend of $\boldsymbol{x}_1$ is dampened. *Source: Own simulation*

A.4 VEC Models

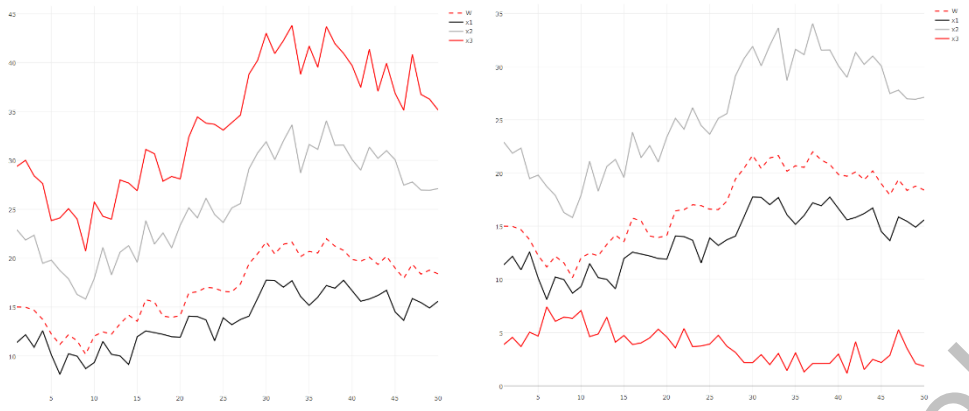
Figure A.4.1 Two variables (in black and grey) and one common trend (in red dashed) with $\lambda_1 = 0.5$ and $\lambda_2 = 2$ on the left and $\lambda_1 = 0.5$ and $\lambda_2 = -0.3$ on the right (whereby the process $x_2(2)$ in grey solid is just the grey dashed process $x_2(1)$ shifted up by 20 units)



Annotation: In both figures, the same random walk (w_t in red dashed, with $w_{1t} = w_{1t-1} + e_{3t}$, e_{3t} as white noise and a starting value of $w_0 = 10$) is driving the two processes x_1 and x_2, but each time with different weights. On the left side, the weight of the process x_1 is $\lambda_1 = 0.5$. Therefore, x_1 is just the half of the random walk plus an error term (such that $x_{1t} = 0.5W_t + e_{1t}$). For the process x_2, $\lambda_2 = 2$ and therefore $x_{2t} = 2w_t + e_{2t}$, so that the process is just twice as much than the driving random walk. In both cases, changes in the random walk leads to changes in the processes x_1 and x_2 in the same direction. The figure on the left side displays the most obvious and simple example. On the right side, the co-integration is a bit harder to detect. While the weight of the process x_1 is still $\lambda_1 = 0.5$, the weight for the process $x_2(1)$ in black dashed is now negative ($\lambda_2 = -0.3$). Through the negative weight, changes in the underlying trend now results in changes with opposite direction for the process $x_2(1)$. Because of the negative weight, the time series $x_2(1)$ would be negative (which would not be plausible for many practical cases). The process $x_2(2)$ in black solid was shifted up by 20 units, so that now, the changes of $x_2(2)$ are still reverse to the changes in the underlying trend, but now the values of the time series $x_2(2)$ are positive.

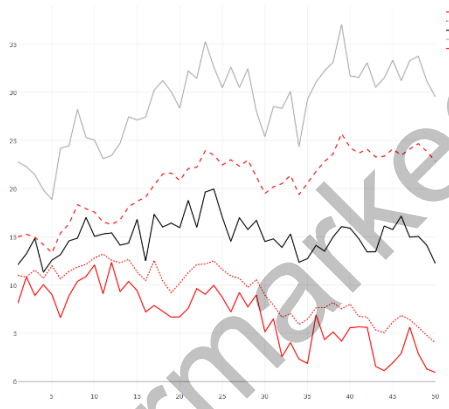
Source: Own simulation

Figure A.4.2 Three variables (in black, red and grey solid) and one common trend (in red dashed) with $\lambda_1 = 0.8, \lambda_2 = 1.5$ *and* $\lambda_3 = 2$ *on the left, and* $\lambda_1 = 0.8, \lambda_2 = 1.5$ *and* $\lambda_3 = -0.3$ *on the right.*



Annotation: In both figures, the same random walk (w_t in red, with $w_{1t} = w_{1t-1} + e_{3t}$, e_{3t} as white noise and a starting value $w_0 = 15$) is driving the three processes x_1, x_2 and x_3, but each time with different weights. This three-variate case is just a simple extension of the bivariate case shown in Figure A.4.1, where all weights are positive on the left side and one weight is negative on the right side ($\lambda_3 = -0.3$). The negative sign again leads to changes for x_3 (in red) which have the opposite direction than the changes in the driving random walk. Due to the negative sign in the weight, the process was shifted up by 9 units to ensure positive values (the series with the negative signs is not shown). *Source: Own simulation*

Figure A.4.3: Three variables (in black, red and grey solid) and two common trend (in red dashed and dotted) with $\lambda_{11} = \lambda_{12} = 0.5, \lambda_{21} = 1.3, \lambda_{22} = 0.2, \lambda_{31} = -0.1$ *and* $\lambda_{32} = 0.95$



Annotation: On the left figure, two random walks (w_{1t} in red dashed and w_{2t} in red dotted) are driving the three processes $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$, each time with different weights. Thereby, the values have been simulated in this way, that the process $\boldsymbol{x}_1$ in black is sharing both random walks to 50%, whereas the processes $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ are almost connected to just one random walk ($\boldsymbol{x}_2$ in grey to $\boldsymbol{w}_1$ and $\boldsymbol{x}_3$ in grey to $\boldsymbol{w}_2$). In this case with three variables and two common trends, the detection of co-integration relationships would be very hard visually. *Source: Own simulation*

Appendix B: Machine Learning techniques

B.1 Naïve Bayes Classifier

In the following, a complete ***example of the Multinomial Naïve Bayes Classifier*** is provided with three spam (S) and two non-spam (NS) emails, whereby ***(A)*** the transformation of the emails into a multinomial term document matrix and ***(B)*** the application of the Naïve Bayes Classifier to this term document matrix is shown. The example is demonstrated on the following five emails:

1. This is a special offer and a special price. (S)
2. 50 percent discount for sport shoes and other sport articles. (S)
3. We offer you a big discount for free. (S)
4. I will be at home in 3 hours (NS)
5. Please call me back. (NS)

(A) Multivariate term document matrix: To transform text into numerical data, there is normally some kind of preprocessing necessary. In the example, the text is first converted to lower cases and then punctuations, the stopwords "this", "is", "a", "and", "for", "other", "we", "you", "i", "will", "at","in" and "me" as well as the numbers "50" and "3" are removed. This then leads to the emails

1. special offer special price (S)
2. percent discount sport shoes sport articles (S)
3. offer big discount free (S)
4. home hours (NS)
5. please call back (NS)

Table B.1: Term document matrix

	Email 1 (S)	Email 2 (S)	Email 3 (S)	Email 4 (NS)	Email 5 (NS)
special	**2**	0	0	0	0
offer	**1**	0	**1**	0	0
price	**1**	0	0	0	0
percent	0	**1**	0	0	0
discount	0	**1**	**1**	0	0
sport	0	**2**	0	0	0
shoes	0	**1**	0	0	0
articles	0	**1**	0	0	0
big	0	0	1	0	0
free	0	0	1	0	0
home	0	0	0	**1**	0
hours	0	0	0	**1**	0
please	0	0	0	0	**1**
call	0	0	0	0	**1**
back	0	0	0	0	**1**

Source: Own illustration

The five emails now contain 19 words, whereby only 15 words are unique. This unique words now form the rows, and the email numbers form the column names of the term document matrix as shown on the left. All in all, the three spam emails consist of 14 and the two non-spam emails of 5 (not unique) words, so that $N_s = 14$ and $N_{Ns} = 5$. The number of distinct words are 15, so that $m = 15$.

(B) Naïve Bayes classifier: To classify the following new spam email

Cheap products, we offer free home deliveries.

the email is first preprocessed like above to receive the transformed email

cheap products offer free home deliveries

which are 6 remaining words, such that $k = 6$. To classify this new email, it is necessary to compute $P(E|C = S)P(C = S)$ as well as $P(E|C = NS)P(C = NS)$ (as described in equation (77)). $P(C = S)$ and $P(C = NS)$ are the probabilities, that any email is spam or not, which is just the amount of spam or non-spam emails to the amount of all emails, such that $P(C = S) = \frac{3}{5} = 0.6$ and $P(C = NS) = \frac{2}{5} = 0.4$. To calculate $P(E|C = S)$ and $P(E|C = NS)$ it is necessary to obtain $\hat{P}(w_j|C = S)$ as well as $\hat{P}(w_j|C = NS)$ for each of the 6 words of the new email via $\hat{P}(w_j|C = S) = \frac{1+n_S(w_j)}{m+N_S}$ and $\hat{P}(w_j|C = NS) = \frac{1+n_{NS}(w_j)}{m+N_{NS}}$ respectively. For example, the probabilities, that the word "cheap" belongs to the class of spam or non-spam emails is calculated as $\hat{P}(cheap|C = S) = \frac{1+0}{15+14} = 0.03$ and $\hat{P}(cheap|C = NS) = \frac{1+0}{15+5} = 0.05$, because the word "cheap" does not occur in either of the spam or non-spam emails, where all spam emails are made up of 14, and all non-spam emails are made up of 5 words (the terms 1 and 15 normalizes the relative frequencies). Because the word "offer" occurs two times in all spam and zero times in all non-spam emails, $\hat{P}(offer|C = S) = \frac{1+2}{15+14} = 0.16$ and $\hat{P}(offer|C = NS) = \frac{1+0}{15+5} = 0.05$. The other words can be calculated analogously, such that the probabilities for all 6 words of the new email are

$\hat{P}(cheap|C = S) = \frac{1+1}{15+14} = 0.069$ $\quad$ $\hat{P}(cheap|C = NS) = \frac{1+0}{15+5} = 0.050$

$\hat{P}(products|C = S) = \frac{1+1}{15+14} = 0.069$ $\quad$ $\hat{P}(products|C = NS) = \frac{1+0}{15+5} = 0.050$

$\hat{P}(offer|C = S) = \frac{1+2}{15+14} = 0.103$ $\quad$ $\hat{P}(offer|C = NS) = \frac{1+0}{15+5} = 0.050$

$\hat{P}(free|C = S) = \frac{1+1}{15+14} = 0.069$ $\quad$ $\hat{P}(free|C = NS) = \frac{1+0}{15+5} = 0.050$

$\hat{P}(home|C = S) = \frac{1+0}{15+14} = 0.034$ $\quad$ $\hat{P}(home|C = NS) = \frac{1+1}{15+5} = 0.100$

$\hat{P}(deliveries|C = S) = \frac{1+0}{15+14} = 0.034$ $\quad$ $\hat{P}(deliveries|C = NS) = \frac{1+0}{15+5} = 0.050$

$P(E|C = S)$ as well as $P(E|C = NS)$ can now be calculated like in formula (77) as $P(E|C = S) = \prod_{j=1}^{k} P(w_j|C = S)$ and $P(E|C = NS) = \prod_{j=1}^{k} P(w_j|C = NS)$, which gives for the new email

$$P(E|C = S) = \hat{P}(cheap|C = S) * \hat{P}(products|C = S) * \hat{P}(offer|C = S) * \ldots$$
$$\ldots \hat{P}(free|C = S) * \hat{P}(home|C = S) * \hat{P}(deliveries|C = S)$$
$$P(E|C = S) = 0.069 * 0.069 * 0.16 * 0.069 * 0.03 * 0.03 = 3.9 * 10^{-8}$$

and

$$P(E|C = \mathrm{NS}) = \hat{P}(cheap|C = NS) * \hat{P}(products|C = NS) * \hat{P}(offer|C = NS) * \ldots$$

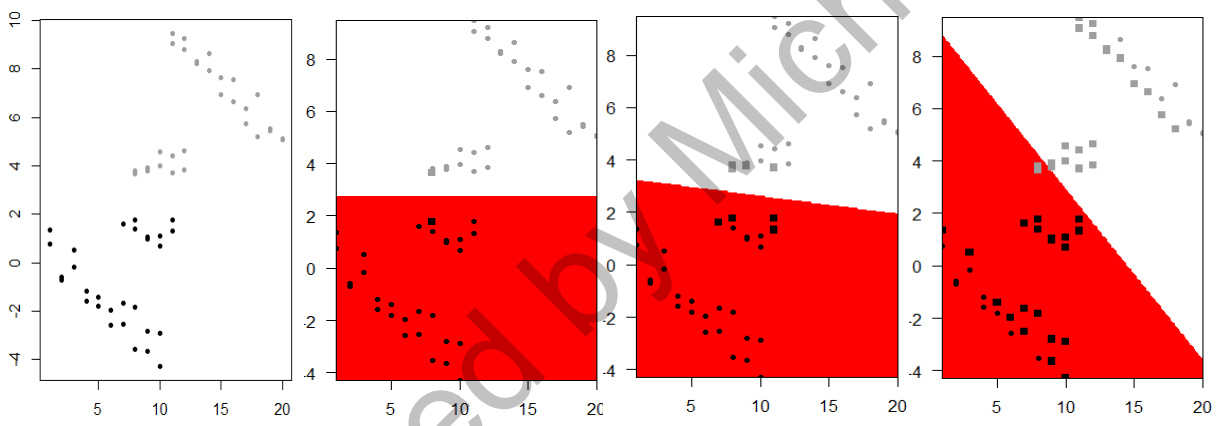
$$\dots \hat{P}(free|C = NS) * \hat{P}(home|C = NS) * \hat{P}(deliveries|C = NS)$$

$$P(E|C = S) = 0.05 * 0.05 * 0.05 * 0.05 * 0.10 * 0.05 = 3.1 * 10^{-8}$$

Finally, the email is classified as in (76) as belonging to the class with the highest probability of $P(C = S|E)$ and $P(E|C = \mathrm{NS})$, calculated as in formula (76) such that if $P(C = S|E) = P(E|S)P(C = S) > P(C = NS|E) = P(E|NS)P(C = NS)$, then the new email is classified as spam – or as non-spam if $P(C = S|E) < P(C = NS|E)$. In the example, $P(C = S|E) = 3.9 * 10^{-8} * 0.6 = 2.3 * 10^{-8}$ and $P(C = NS|E) = 3.1 * 10^{-8} * 0.4 = 1.3 * 10^{-8}$, and because $2.3 * 10^{-8} > 1.3 * 10^{-8}$, the email would be correctly classified as spam.

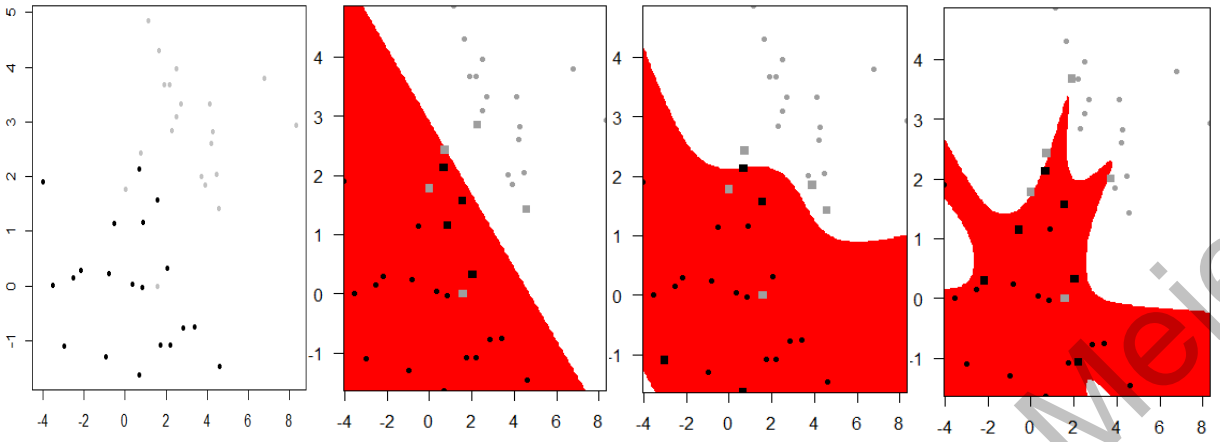
B.2 Support Vector Machines

Figure B.2.1: Linear class boundaries for different values of C applied to a linear separable data set



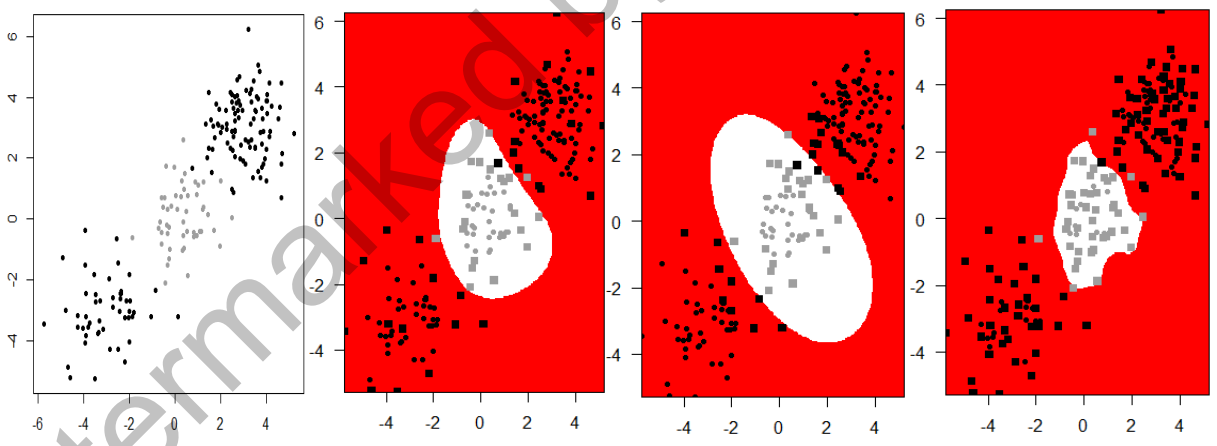
Annotation: On the first figure from left, two classes of data points are shown, which are linearly separable (in grey class 1 and in black class 2). For the classification in the second figure, a low value for C was chosen, such that not much cost (violations of the margin) is allowed. This leads to a narrow margin. Therefore, only two support vectors are taking part in defining the separating hyperplane (the support vectors are marked as squares). Despite the fact, that the overall data would be better classified by a more diagonal line, the hyperplane in the upper right figure is horizontal, because only the two support vectors are defining the hyperplane. In the third figure, the data was classified with a larger value for C. Therefore, the margin widens and the number of support vectors increases, such that the separating hyperplane was defined by 8 support vectors. The hyperplane in the fourth figure was defined by even a higher number of support vectors (42). This is due to an even higher value for the cost C, which leads to a very wide margin. Other than this example might suggest, a wider margin does not always lead to better results, i.e. a better generalization, because it may tend to overfitting the data. *Source: Own simulation*

Figure B.2.2: Linear and different polynomial kernels applied to a non-linear separable data set



Annotation: In the first figure from the left, two classes of data points are shown, which are not-linearly separable (in grey class 1 and in black class 2). In the second figure, a linear kernel (no kernel) was applied to the data, leading to a linear separable hyperplane (defined by 9 support vectors). On the third figure, a polynomial kernel of order 4 was applied. Thereby, the amount of support vectors did not change (the hyperplane was still defined by 9 support vectors). But the transformation of the feature space (due to the polynomial kernel) leads to a much more flexible hyperplane. Due to the more flexible class boundary, the error of misclassification could be reduced from 3 to 2 (on the second figure, 3 grey observations have been misclassified whereby on the third figure, only two grey observations have been misclassified). The fourth figure shows a polynomial kernel of a higher order (7), whereby the amount of support vectors only was increased slightly (from 9 to 11). Due to the even more flexible hyperplane, the amount of misclassified observations even could be reduced to one. Similar to a wider margin, a more flexible hyperplane does not necessarily lead to better results, i.e. a better generalization, because it may lead to overfitting the data. *Source: Own simulation*

Figure B.2.3: Radial kernels with different values for γ applied to a non-linear separable data set



Annotation: In the first figure from left, two classes of data points are shown, which are not-linearly separable (in grey class 1 and in black class 2). In the second figure, a radial kernel with the default value for γ, which is $\gamma = \frac{1}{number\ of\ features}$, was applied to the data. γ thereby can be interpreted as how much observations, that lie far away from the center of the circle, influence the decision boundary. In the third figure, a value for γ was chosen, which is *smaller* than the default value, meaning, that observations, which lie further away from the center of the grey observations have a stronger influence on the separating hyperplane – what leads to a less flexible hyperplane. In the fourth figure, a value for γ was chose, which is *bigger* than the default value, meaning, that observations, which lie more away from the center of the grey observations have now not that much influence on the separating hyperplane – this leads to a more flexible hyperplane. *Source: Own simulation*

B.3 Neural Networks

B.3.1: Logical gates

To understand NNs better, a short description is given about how the ***four logical gates***, ***(A)*** NOT, ***(B)*** OR, ***(C)*** AND and ***(D)*** XOR gate can be interpreted as small decision making units and how these units can be solved with simple NN`s containing only some neurons. Thereby, the gates can be interpreted and visualized analogous to the decision boundaries of a SVM.

(A) NOT gate:

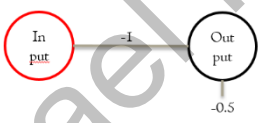
Signals:

Input	Output
0	1
1	0

Example:

If the change of the exchange rate in t-1 was **NOT** positive (1), then the prediction for the change of the exchange rate in t will be positive (1)

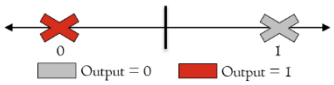
NN:



Calculation:

$Input = 0 \rightarrow 0 * -1 = 0 \rightarrow 0 > -0.5 \rightarrow Output = 1$
$Input = 1 \rightarrow 1 * -1 = -1 \rightarrow -1 < -0.5 \rightarrow Output = 0$

Visualisation:



(B) OR gate:

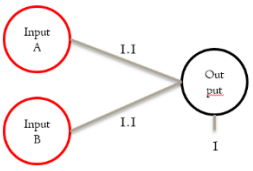
Signals:

Input A	Input B	Output
0	0	0
1	0	1
0	1	1
1	1	1

Example:

If the interest rate in t-1 **OR** the GDP growth in t-1 was high (1), then the prediction for the change of the exchange rate in t will be positive (1)

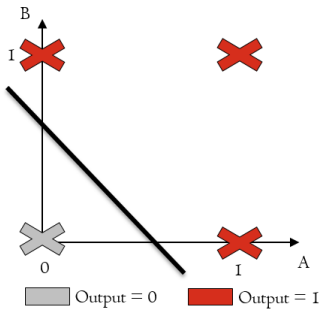
NN:



Calculation:

$A = 0 \,\&\, B = 0 \rightarrow 0 * 1.1 + 0 * 1.1 = 0 \rightarrow$
$0 < 1 \rightarrow Output = 0$
$A = 1 \,\&\, B = 0 \rightarrow 1 * 1.1 + 0 * 1.1 = 1.1 \rightarrow$
$1.1 > 1 \rightarrow Output = 1$
$A = 0 \,\&\, B = 1 \rightarrow 0 * 1.1 + 1 * 1.1 = 1.1 \rightarrow$
$1.1 > 1 \rightarrow Output = 1$
$A = 1 \,\&\, B = 1 \rightarrow 1 * 1.1 + 1 * 1.1 = 2.2 \rightarrow$
$2.2 > 1 \rightarrow Output = 1$

Visualisation:



(C) AND gate:

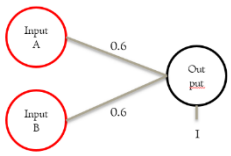
Signals:

Input A	Input B	Output
0	0	0
I	0	0
0	I	0
I	I	I

Example:

If the interest rate in t-I **AND** the GDP growth in t-I was high (I), then the prediction for the change of the exchange rate in t will be positive (I)

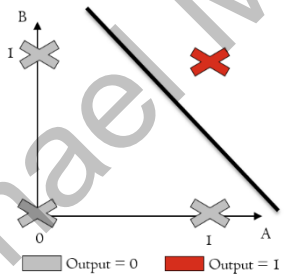
NN:



Calculation:

$A = 0 \,\&\, B = 0 \rightarrow 0 * 0.6 + 0 * 0.6 = 0 \rightarrow$
$0 < 1 \rightarrow Output = 0$
$A = 1 \,\&\, B = 0 \rightarrow 1 * 0.6 + 0 * 0.6 = 0.6 \rightarrow$
$0.6 < 1 \rightarrow Output = 0$
$A = 0 \,\&\, B = 1 \rightarrow 0 * 0.6 + 1 * 0.6 = 0.6 \rightarrow$
$0.6 < 1 \rightarrow Output = 0$
$A = 1 \,\&\, B = 1 \rightarrow 1 * 0.6 + 1 * 0.6 = 1.2 \rightarrow$
$1.2 > 1 \rightarrow Output = 1$

Visualisation:



(D) XOR gate:

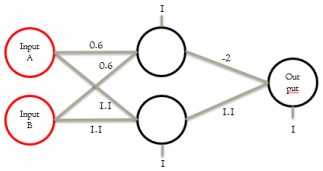
Signals:

Input A	Input B	Output
0	0	0
I	0	I
0	I	I
I	I	0

Example:

If the interest rate in t-I **OR** the GDP growth, but **NOT** both, were high (I) in t-I, then the prediction for the change of the exchange rate in t will be positive (I)

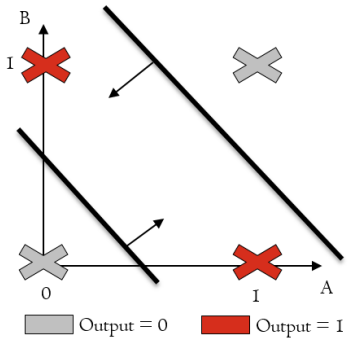
NN:



Calculation:

$A = 0 \,\&\, B = 0 \rightarrow 0 * 0.6 + 0 * 0.6 = 0 \rightarrow 0 < 1 \rightarrow$
$Hidden\ 1 = 0 \rightarrow A = 0 \,\&\, B = 0 \rightarrow 0 * 1.1 + 0 * 1.1 = 0 < 1 \rightarrow$
$Hidden\ 2 = 0 \rightarrow 0 * -2 + 0 * 1.1 = 0 \rightarrow 0 < 1 \rightarrow Output = 0$
$A = 1 \,\&\, B = 0 \rightarrow 1 * 0.6 + 0 * 0.6 = 0.6 \rightarrow 0.6 < 1 \rightarrow$
$Hidden\ 1 = 0 \rightarrow A = 1 \,\&\, B = 0 \rightarrow 1 * 1.1 + 0 * 1.1 = 1.1 > 1 \rightarrow$
$Hidden\ 2 = 1 \rightarrow 0 * -2 + 1 * 1.1 = 0 \rightarrow 1.1 > 1 \rightarrow Output = 1$
$A = 0 \,\&\, B = 1 \rightarrow 0 * 0.6 + 1 * 0.6 = 0.6 \rightarrow 0.6 < 1 \rightarrow$
$Hidden\ 1 = 0 \rightarrow A = 0 \,\&\, B = 1 \rightarrow 0 * 1.1 + 1 * 1.1 = 1.1 > 1 \rightarrow$
$Hidden\ 2 = 1 \rightarrow 0 * -2 + 1 * 1.1 = 1.1 \rightarrow 1.1 > 1 \rightarrow Output = 1$
$A = 1 \,\&\, B = 1 \rightarrow 1 * 0.6 + 1 * 0.6 = 1.2 \rightarrow 1.2 > 1 \rightarrow$
$Hidden\ 1 = 1 \rightarrow A = 1 \,\&\, B = 1 \rightarrow 1 * 1.1 + 1 * 1.1 = 2.2 > 1 \rightarrow$
$Hidden\ 2 = 1 \rightarrow 1 * -2 + 1 * 1.1 = -0.9 \rightarrow -0.9 < 1 \rightarrow Output = 0$

Visualisation:



Through the short description of the four logical gates, it can be imagined, how many different kind of gates (next to each other or behind each other) can be combined together in complex NNs with many hidden layers and many neurons in each layer to reproduce complicated high dimensional decision boundaries.

B.3.2: Feedforward, Cost and Backpropagation

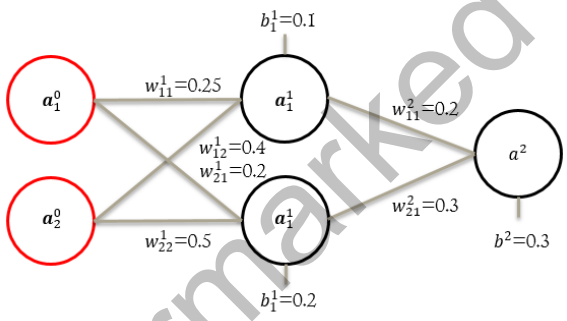
To illustrate the ***learning process*** of NNs, a small example is provided, where the network learns an OR gate. Therefore, ***(A)*** the inputs are fed forward through the network (with random initial weights), ***(B)*** the cost is calculated and ***(C)*** the loss is propagated back through the network in order to change the weights according to the gradient descent update rule.

(A) Feedforward: The inputs, outputs and the randomly chosen starting weights and biases are

$$\boldsymbol{A}^0 = \begin{bmatrix} t(\boldsymbol{a}_1^0) \\ t(\boldsymbol{a}_2^0) \end{bmatrix} = \begin{bmatrix} a_1^{01} & a_2^{01} \\ a_1^{02} & a_2^{02} \\ a_1^{03} & a_2^{03} \\ a_1^{04} & a_2^{04} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad \boldsymbol{W}^1 = \begin{bmatrix} w_{11}^1 & w_{21}^1 \\ w_{12}^1 & w_{22}^1 \end{bmatrix} = \begin{bmatrix} 0.25 & 0.2 \\ 0.4 & 0.5 \end{bmatrix}$$

$$\boldsymbol{w}^2 = \begin{bmatrix} w_{11}^2 \\ w_{21}^2 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.3 \end{bmatrix} \quad \boldsymbol{b}^1 = \begin{bmatrix} b_1^1 \\ b_2^1 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} \quad \boldsymbol{bb}^1 = \begin{bmatrix} \boldsymbol{b}^{1\prime} \\ \boldsymbol{b}^{1\prime} \\ \boldsymbol{b}^{1\prime} \\ \boldsymbol{b}^{1\prime} \end{bmatrix} \quad b^2 = 0.3 \quad \boldsymbol{bb}^2 = \begin{bmatrix} 0.3 \\ 0.3 \\ 0.3 \\ 0.3 \end{bmatrix}$$

where $\boldsymbol{bb}^1$ and $\boldsymbol{bb}^2$ represents vectors of repeated stacked biases of the neurons in the first (hidden) and the second (output) layer and $\boldsymbol{a}_1^0$ represents a vector of inputs for feature one (the activations of neuron one in the input layer zero). Therefore, the structure of the NN as well as the initial weights and biases looks like on the left. Feeding the inputs $\boldsymbol{A}^0$ through the weights to layer one and adding the biases $\boldsymbol{bb}^1$ leads to the weighted inputs $\boldsymbol{Z}^1$ calculated as



$$\boldsymbol{Z}^1 = \boldsymbol{A}^0 * \boldsymbol{W}^1 + \boldsymbol{bb}^1 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} * \begin{bmatrix} 0.25 & 0.2 \\ 0.4 & 0.5 \end{bmatrix} + \begin{bmatrix} 0.1 & 0.2 \\ 0.1 & 0.2 \\ 0.1 & 0.2 \\ 0.1 & 0.2 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.2 \\ 0.35 & 0.4 \\ 0.5 & 0.7 \\ 0.75 & 0.9 \end{bmatrix}$$

Each weighted input is then passed to the sigmoid activation function to receive the activations of the neurons in layer one $\boldsymbol{A}^1$ calculated as

$$\boldsymbol{A}^1 = \sigma(\boldsymbol{Z}^1) = \sigma\begin{bmatrix} 0.1 & 0.2 \\ 0.35 & 0.4 \\ 0.5 & 0.7 \\ 0.75 & 0.9 \end{bmatrix} = \begin{bmatrix} \sigma(0.1) & \sigma(0.2) \\ \sigma(0.35) & \sigma(0.4) \\ \sigma(0.5) & \sigma(0.7) \\ \sigma(0.75) & \sigma(0.9) \end{bmatrix} = \begin{bmatrix} \frac{1}{1+e^{-0.1}} & \frac{1}{1+e^{-0.2}} \\ \frac{1}{1+e^{-0.35}} & \frac{1}{1+e^{-0.4}} \\ \frac{1}{1+e^{-0.5}} & \frac{1}{1+e^{-0.7}} \\ \frac{1}{1+e^{-0.75}} & \frac{1}{1+e^{-0.9}} \end{bmatrix}$$

$$= \begin{bmatrix} 0.52 & 0.55 \\ 0.59 & 0.60 \\ 0.62 & 0.67 \\ 0.68 & 0.71 \end{bmatrix}$$

The activations are then again first fed through the weights to layer two, then the biases $\boldsymbol{bb}^2$ are added and finally the weighted inputs are passed to the sigmoid activation function the receive the activations of layer two (the outputs of the network)

$$\boldsymbol{a}^2 = \sigma(\boldsymbol{z}^2) = \sigma(\boldsymbol{A}^1 * \boldsymbol{w}^2 + \boldsymbol{bb}^2) = \sigma\left(\begin{bmatrix} 0.52 & 0.55 \\ 0.59 & 0.60 \\ 0.62 & 0.67 \\ 0.68 & 0.71 \end{bmatrix} * \begin{bmatrix} 0.2 \\ 0.3 \end{bmatrix} + \begin{bmatrix} 0.3 \\ 0.3 \\ 0.3 \\ 0.3 \end{bmatrix}\right)$$

$$\boldsymbol{a}^2 = \sigma\left(\begin{bmatrix} 0.27 \\ 0.30 \\ 0.32 \\ 0.35 \end{bmatrix} + \begin{bmatrix} 0.3 \\ 0.3 \\ 0.3 \\ 0.3 \end{bmatrix}\right) = \begin{bmatrix} \sigma(0.57) \\ \sigma(0.60) \\ \sigma(0.62) \\ \sigma(0.65) \end{bmatrix} = \begin{bmatrix} 0.64 \\ 0.64 \\ 0.65 \\ 0.66 \end{bmatrix}$$

(B) Calculating the cost: With the activations of the output neurons, the overall cost c over all inputs can be calculated as

$$c = \frac{1}{2}(\boldsymbol{y} - \boldsymbol{a}^2)^2 = \frac{1}{2} * \left(\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.64 \\ 0.64 \\ 0.65 \\ 0.66 \end{bmatrix}\right)^2 * \boldsymbol{I} * \frac{1}{4} = \frac{1}{2} * \begin{bmatrix} -0.64^2 \\ 0.36^2 \\ 0.35^2 \\ -0.66^2 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \frac{1}{4}$$

$$c = \begin{bmatrix} 0.20 \\ 0.06 \\ 0.06 \\ 0.22 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \frac{1}{4} = 0.136$$

where $\boldsymbol{I} * \frac{1}{4}$ was used to average the individual costs over all individual costs of each input.

(C) Backpropagation: As described in formula (102), the errors of layer two can be calculated as

$$\boldsymbol{\delta}^2 = -(\boldsymbol{y} - \boldsymbol{a}^2)\odot\sigma(\boldsymbol{a}^2)\odot\sigma(\boldsymbol{\iota} - \boldsymbol{a}^2) = -\begin{bmatrix}0\\1\\1\\0\end{bmatrix} - \begin{bmatrix}0.64\\0.64\\0.65\\0.66\end{bmatrix}\odot\sigma\begin{bmatrix}0.64\\0.64\\0.65\\0.66\end{bmatrix}\sigma\begin{bmatrix}1-0.64\\1-0.64\\1-0.65\\1-0.66\end{bmatrix}$$

$$\boldsymbol{\delta}^2 = -\begin{bmatrix}-0.64\\0.36\\0.35\\-0.66\end{bmatrix}\odot\begin{bmatrix}\sigma(0.64)\\\sigma(0.64)\\\sigma(0.65)\\\sigma(0.66)\end{bmatrix}\odot\begin{bmatrix}\sigma(0.36)\\\sigma(0.36)\\\sigma(0.35)\\\sigma(0.34)\end{bmatrix} = \begin{bmatrix}0.25\\-0.14\\-0.13\\0.25\end{bmatrix}$$

The gradient for the bias of the neuron in layer two is then calculated as the average of the individual gradients as $\nabla c^{b^2} = \frac{1}{4} * I * \frac{\partial C_t}{\partial b_j^l} = \frac{1}{4} * I * \nabla c_t^{b^2} = \frac{1}{4} * \boldsymbol{I} * \boldsymbol{\delta}^2 = \frac{0.25-0.14-0.13+0.25}{4} = 0.0575$

Next, the gradients with respect to the weights $\boldsymbol{w}^2$ are calculated as described in (104a) as

$$\nabla c^{w^2} = \boldsymbol{A}^{1\prime} * \boldsymbol{\delta}^2 = \begin{bmatrix}0.52 & 0.59 & 0.62 & 0.68\\0.55 & 0.60 & 0.67 & 0.71\end{bmatrix} * \begin{bmatrix}0.25\\-0.14\\-0.13\\0.25\end{bmatrix} = \begin{bmatrix}0.52 & 0.55\\0.59 & 0.60\\0.62 & 0.67\\0.68 & 0.71\end{bmatrix} = \begin{bmatrix}0.13\\0.14\end{bmatrix}$$

Next, the errors in layer I can be calculated as described in formula (103) as

$$\boldsymbol{\delta}^1 = \boldsymbol{\delta}^2 * \boldsymbol{w}^{2\prime}\odot\sigma(\boldsymbol{Z}^1)\odot\sigma(\boldsymbol{\iota} * \boldsymbol{\iota}' - \boldsymbol{Z}^1)$$

$$\boldsymbol{\delta}^1 = \begin{bmatrix}0.25\\-0.14\\-0.13\\0.25\end{bmatrix} * [0.2; 0.3]\odot\sigma\left(\begin{bmatrix}0.1 & 0.2\\0.35 & 0.4\\0.5 & 0.7\\0.75 & 0.9\end{bmatrix}\right)\odot\sigma\left(\begin{bmatrix}-0.9 & -0.8\\-0.65 & -0.6\\-0.5 & -0.3\\-0.25 & -0.1\end{bmatrix}\right) = \begin{bmatrix}0.02 & 0.03\\-0.01 & -0.02\\-0.01 & -0.02\\0.02 & 0.03\end{bmatrix}$$

The gradient with respect to the weights $\boldsymbol{W}^1$ can then be calculated as

$$\nabla c^{W^1} = \boldsymbol{A}^{0\prime} * \boldsymbol{\delta}^2 = \begin{bmatrix}0 & 1 & 0 & 1\\0 & 0 & 1 & 1\end{bmatrix} * \begin{bmatrix}0.02 & 0.03\\-0.01 & -0.02\\-0.01 & -0.02\\0.02 & 0.03\end{bmatrix} = \begin{bmatrix}0.0088 & 0.0124\\0.0089 & 0.0128\end{bmatrix}$$

Finally, the following update rules can be applied to change the weights and the biases where the learning rate was chosen to 1.5

$$\boldsymbol{w}^2 = \boldsymbol{w}^2 - \eta\nabla c^{w^2} = \begin{bmatrix}0.2\\0.3\end{bmatrix} - 0.5 * \begin{bmatrix}0.13\\0.14\end{bmatrix} = \begin{bmatrix}-0.006\\0.084\end{bmatrix}$$

$$\boldsymbol{W}^1 = \boldsymbol{W}^1 - \eta\nabla c^{W^1} = \begin{bmatrix}0.25 & 0.2\\0.4 & 0.5\end{bmatrix} - 0.5 * \begin{bmatrix}0.0088 & 0.0124\\0.0089 & 0.0128\end{bmatrix} = \begin{bmatrix}0.24 & 0.18\\0.39 & 0.48\end{bmatrix}$$

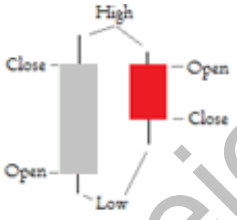
$$b^2 = b^2 - \eta\nabla c^{b^2} = 0.3 - 0.5 * 0.0575 = 0.27125.$$

Now, when the inputs are fed forward through the network again – with the changed weights and biases – and the cost is recalculated, this leads to a value of 0.127, which is 0.009 smaller than the previous cost of 0.136. Repeating (A)-(C) several times reduces the cost further and further until the desired approximation to the output is reached.

Appendix C: Technical analysis of the exchange rate

While there are almost an unlimited number of technical indicators, only a few are widely used. To get an intuition about technical indicators (also known as chart indicators) one of the most commonly used indicators are presented in the following. Most of the technical indicators use opening, high, low and closing prices as well as the volume of a financial time series (in the following revered to as short open, high, low, close and volume). Thereby typically candle chart are used. On the right in Figure C.1, a candle chart with two price changes is shown, where the first change was positive and the second change was negative. In the ***following10 of the most widely used technical indicator (A to J)*** are presented.

Figure C.1: Candle chart



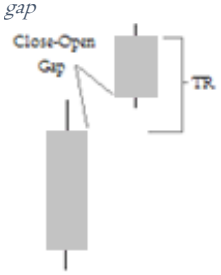
Source: Own illustration

A) Average True Range (ATR)

The ATR is a measure of volatility and is calculated as the arithmetic average of the past True Ranges (TR). The TR is defined as the greatest of 1) current high less current low, 2) current high minus the previous close (absolute value) or 3) current low less previous close (absolute value). As a first step as in case 1), it is easy to think about the ATR as just the average of the past differences between the high- and the low-prices. So for example for the last three periods

$$ATR = \frac{(H_t - L_t) + (H_{t-1} - L_{t-1}) + (H_{t-2} - L_{t-2})}{3} \tag{109}$$

Figure C.2: Open-close-gap



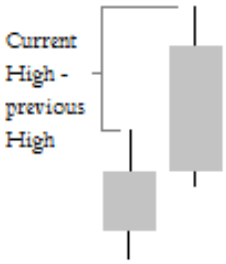
Source: Own illustration

If the range of the past highs and lows was high on average, then the ATR will be high, indicating high volatility. Normally, the close of the previous candle will be the open for the next candle. But this is not always be true. The cases 2) and 3) are in the event of gaps between the close- and open prices. If the current low was higher than the previous close, then the current high minus the previous close is used as the TR (case 2)). Case 3) is in the event of negative close-open-gaps. Typically, the ATR is based on 14 periods.

B) Average Directional Index (ADX)

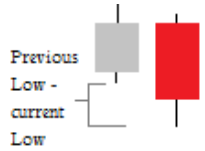
The ADX is a measure of trend strength. The ADX alone does not regard the direction of the trend, but it is complemented by the Plus and Minus Directional Indicator (+DI and -DI). Used together, chartists are able to determine the strength, as well as the direction of a trend. The ADX is based on the Plus and Minus Directional Movement (+DM and –DM). Let´s first look at these Directional Movements. In the right figure, the current high is higher than the previous high and therefore, the difference will be positive. On contrary, the previous low minus the current low will be negative, because the previous low is lower than the current low. In this case, the +DM will be the difference between the current high and the previous high and the –DM will

Figure C.3: Difference current previous high



Source: Own illustration

Figure C.4: Difference previous current



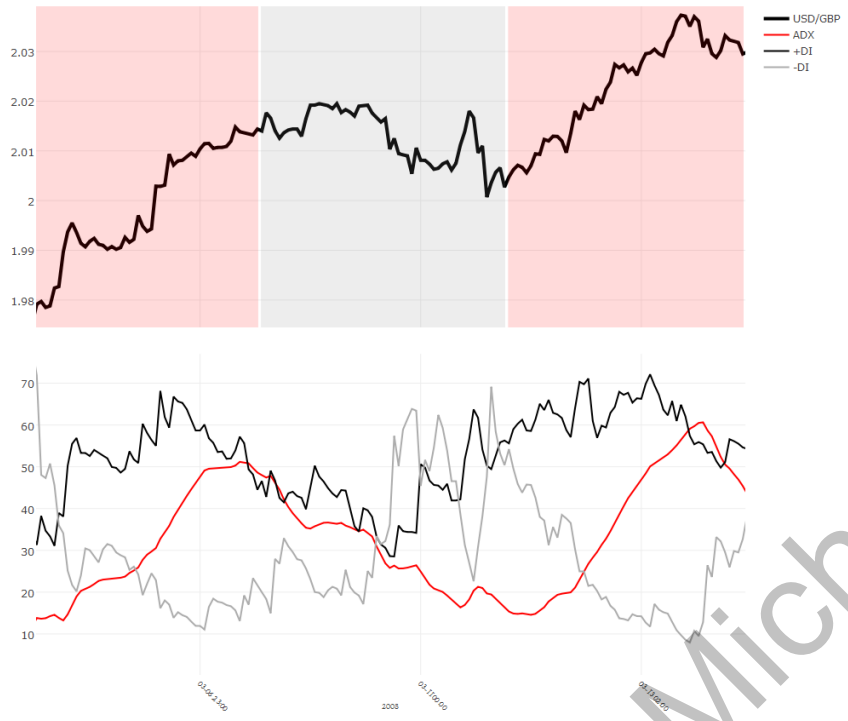
Source: Own illustration

be zero, indicating a positive movement. In Figure C.4 on the left, the Difference between the current high and the previous high is negative, but the difference between the previous low and the current low is positive. In such a case, +DM will be zero, and –DM will be the difference between the previous low and the current low, indicating a negative movement. In general, a directional movement is positive, when the current high minus the previous high is greater than the previous low minus the current low and when this difference is positive. A directional movement is negative, when the previous low minus the current low is greater than the current high minus the previous high and when this difference is positive. In the case of an inside day movement as shown in the right figure, both differences are negative. In this case, +DM and –DM will both be zero. Averaging the +DMs and –DMs (let´s say for 14 periods) and dividing this average through the 14 periods ATR results in the +DI and –DI, which then can be interpreted as a standardized sign of the strength of the last 14 periods movements. A high +DI indicates strong positive movements in the last 14 periods, and the high –DI strong negative movements. To obtain the ADX, the last step is to calculate the Directional Index (DX) and again, average this values for the last periods. The DX is calculated as follows: the absolute value of the difference of +DI and –DI is divided by the sum of +DI and –DI. Because absolute values are used for the difference of +DI and –DI, DX does not show the direction of the movement. The division through the sum of +DI and –DI can also be interpreted as a form of standardizing. If +DI and –DI are high, then there have been strong positive as well as strong negative movements in the last 14 periods. In this case, the DX will tend to be relatively small. In the case of a high +DI and a small –DI (which indicates, that there have been strong positive but just weak negative movements in the past 14 periods), the DX will tend to be high (the same holds for a high –DI and a small +DI). The ADX is then the average of the DX values in a time period (typically again 14 periods). ADX, +DI and –DI are percentage terms and vary between 0 and 100. It needs 28 periods to get the first ADX (14 to average the DIs and 14 to average the ADX). The ADX can be used for a trend following system. As a rule of thumb, an ADX over 25 is often said to indicate a trend, whereas an ADX below 25 indicates no trend. The following figure illustrates an ADX applied to the USD/GBP exchange rate.

Figure C.5 Inside day movement

Source: Own illustration

Figure C.6: USD/GBP exchange rate, +DI, -DI and ADX between 2008/03/05 17:00 and 2008/03/14 02:00 GMT



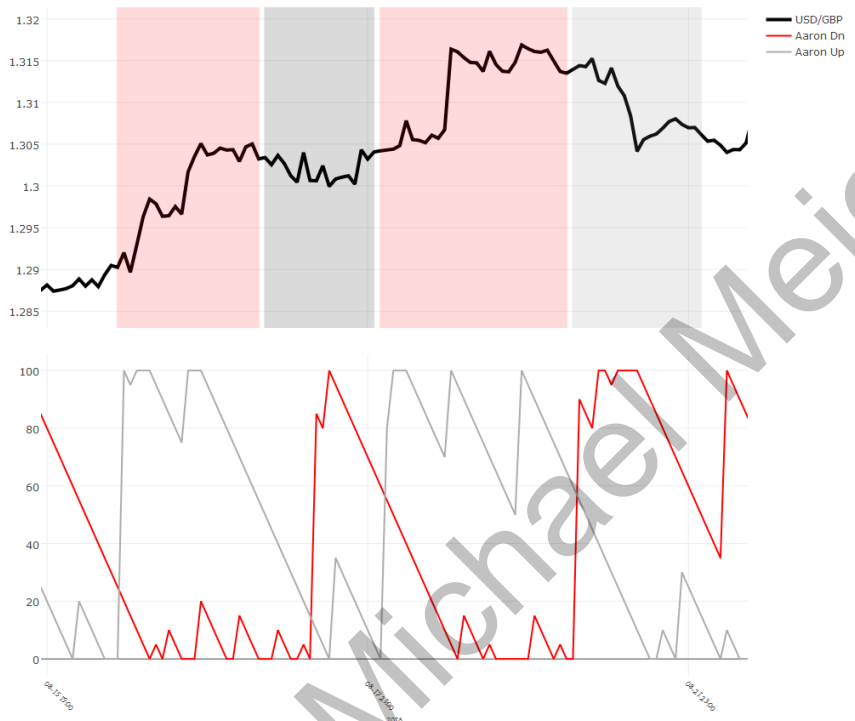
Annotation: The figure shows the USD/GBP exchange rate (black bold), as well as +DI (black), -DI (grey) and ADX (red) between 2008/03/05 17:00 GMT and 2008/03/14 02:00 GMT. Clearly, ADX indicates a strong positive trend (in which +DI is high and –DI is low, marked as red shaded area), a sideways movement (where +DI and –DI are either both high or both low, marked as grey shaded area) and again a strong positive trend (in which +DI is high and –DI is low again, marked as red shaded area).

Source: Own simulation

C) Aroon

ADX is high, when a trend already has been manifested. In contrast to ADX, the two Aroon indicator (Aroon Up and Aroon Down) try to indicate the beginning of a new trend. The simple idea behind the Aroon indicator assumes, that a new trends may arise, when new highs or lows (compared to the last n observations) arise. In this way, the Aroon indicators measures the number of periods since prices recorded an n-day high or low. A 25-period Aroon Up measures the number of periods, since a 25-period high occurred the last time. As a rule of thumb, 50 is often used as a threshold to indicate a bull or a bear market. The Aroon indicators are percentage terms and vary between 0 and 100. For a 25-period Aroon Up Indicator, a value over 50 means, that a new high was recorded within the last 12 periods. A new trend may be emerging, if the Aroon Up indicator surges to 100, and on the same time, the Aroon Down indicator is decreasing and staying at low levels (indicating that there did not emerge new lows in the recent period). A consolidation is present, when Aroon Up and Down move lower in parallel fashion or when both remain at lower levels (approximately under 30). The following figure illustrates the Aroon indicators applied to the USD/GBP exchange rate.

Figure C.7: USD/GBP exchange rate, Aroon Up and Aroon Down between 2016/08/15 18:00 and 2016/08/22 08:00 GMT



Annotation: The figure shows the USD/GBP exchange rate (blue), the Aroon Up (grey) and Aroon Down (red) between 2016/08/15 18:00 and 2016/08/15 08:00 GMT. The first red shaded area starts with a surge of the Aroon Up and a decrease of the Aroon Down to low levels (this can indicate a starting trend, because new highs are arriving and new lows did not arrive in the latest time). This signal can be regarded as successful. In the subsequent dark grey shaded area, first Arron Up falls and then Aroon Down surges to high levels, what would have given a (wrong) signal for an upcoming downward trend. Nevertheless, it can be seen, that Aroon Up does not stay at low and Aroon Down not at high levels, indicating that no strong trend is present (still in the dark grey shaded area). The subsequent red shaded area can again be regarded as a period, where the Aroon indicator would have given a proper signal, starting with a surge in the Aroon Up and a fall of the Aroon Down to low levels (indicating a starting upward trend). The last grey shaded area also can be regarded as a period, where the Aroon indicator would have given a proper signal, this time indicating an upcoming downward trend (light grey shaded area). *Source: Own simulation*

D) On Balance Volume (OBV)

The OBV indicator measures positive and negative volume flows, i.e. buying and selling pressure by adding volumes on up moves and subtracting volume on down moves. The OBV indicator is calculated as

$$Current\ OVB = Previous\ OVB + Current\ Volume \tag{110}$$

if the current closing price is above the prior closing price or as

$$Current\ OVB = Previous\ OVB - Current\ Volume \tag{111}$$

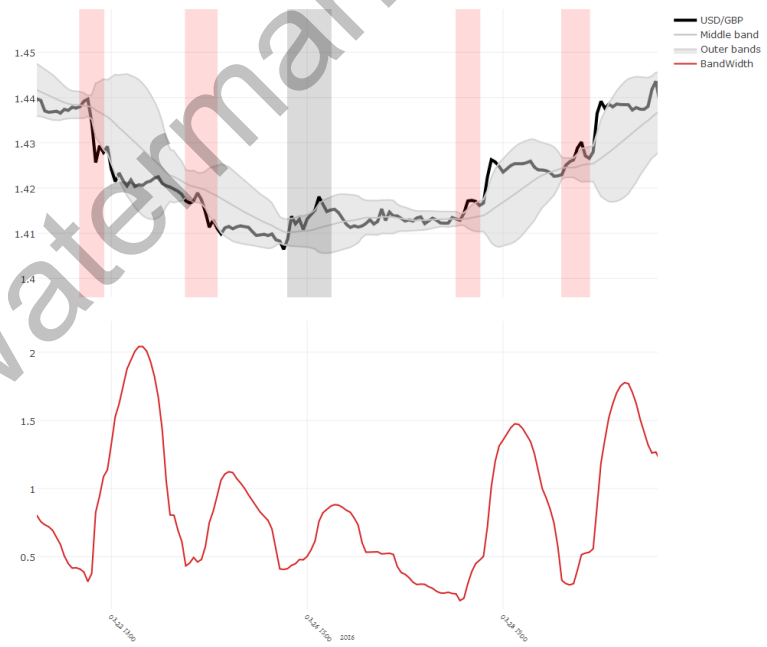
if the current closing price is below the prior closing price. Granville, who introduced the indicator in 1963, argued, that OVB often moves before prices.

This OVB indicator can be used by chartists to A) predict price reversals or B) confirm price trends. A) A positive trend reversal (turning from bearish into bullish markets) thereby is expected, when OBV moves higher or forms a higher low while at the same time prices move lower or forge a lower low. A negative trend reversal (turning from bullish into bearish markets) is expected, when OBV moves lower or forms a lower low while at the same time prices move higher or forge a higher high. B) Trends can be confirmed, when OVB and prices move in the same direction. Rising prices are then confirmed with rising demand (rising volumes) so that it can be argued, that the trend may continue.

E) Bollinger Bands®

Bollinger Bands are volatility bands placed above and below a middle band. The middle band is a simple moving average. The middle band and the outer bands often uses a lookback period of 20 periods. The outer bands are usually set to two standard deviations above and below the middle band. Because it needs a pretty strong price move to exceed the upper or lower band, this could signal the start of an up- or a downward trend. In contrast to this, because prices are relatively high (or low) when they are above (or below) the outer band, this could mean, that they are overpriced (or underpriced) and that they could revert back to the middle band soon. Bollinger Bands are not meant to be used as a standalone tool but are more commonly combined with other indicators. Another property of Bollinger Bands are the characteristic, that they widen when volatility increase and narrow when volatility decreases. This can indicate stormy high-volatile or quiet low-volatile times. To receive a measure for the volatility, the Bollinger Band Width can be calculated as: (Upper Band – Lower Band) / Middle Band. Dividing the difference through the middle band normalizes the Band Width and makes it comparable to other time series. It can be argued, that periods of low volatility may be followed by periods of high volatility. A narrowing of the band can alert chartists to prepare for a new move. A low Band Width in combination with a subsequent strong price move, which breaks and exceeds the outer band, can then be used as a signal for a new emerging trend (while the direction of the trend depends on whether the upper or lower band was breached). The following figure provides an illustration of the Bollinger Bands and the Bollinger Band Width applied to the USD/GBP exchange rate.

Figure C.8: USD/GBP exchange rate, Bollinger Middle Band with the corresponding outer bands as well as the Band Width between 2016/03/21 18:00 and 2016/03/30 11:00 GMT



Annotation: The figure shows the USD/GBP exchange rate (black bold line), the Bollinger Middle Band (grey line) with the corresponding outer bands (grey area around grey line) as well as the Band Width (red line) between 2016/03/21 18:00 and 2016/03/30 11:00 GMT. The red shaded areas show a narrowing of the Bollinger Band with a subsequent breakthrough through the outer bands which can be regarded as successful signals (because the subsequent price moves followed a small trend with the same direction of the breakthrough). The grey shaded area whereas can be regarded as a wrong signal (because after the narrowing of the Band Width, there is not trend arising and prices moved sideways, what is not in accordance to the assumptions). *Source: Own simulation*

F) Accumulation/Distribution Line (ADL)

The ADL is similar to the OVB indicator in the sense, that both measure the money flow into or out of a security. Thereby, the ADL is calculated in three steps. 1) The Money Flow Multiplier is calculated based on the relationship of the close to the high-low range, such that

$$Money\ Flow\ Multiplier = [(Close - Low) - (High - Close)]/(High - Low) \quad (112)$$

2) The Money Flow Multiplier is multiplied by the period´s volume to come up with the Money Flow Volume, i.e.

$$Money\ Flow\ Volume = Money\ Flow\ Multiplier * Volume\ for\ the\ period \quad (113)$$

3) The ADL is then finally calculated as the running total of the Money Flow Volumes, such that

$$ADL = Previous\ ADL + Current\ Period´s\ Money\ Flow\ Volume \quad (114)$$

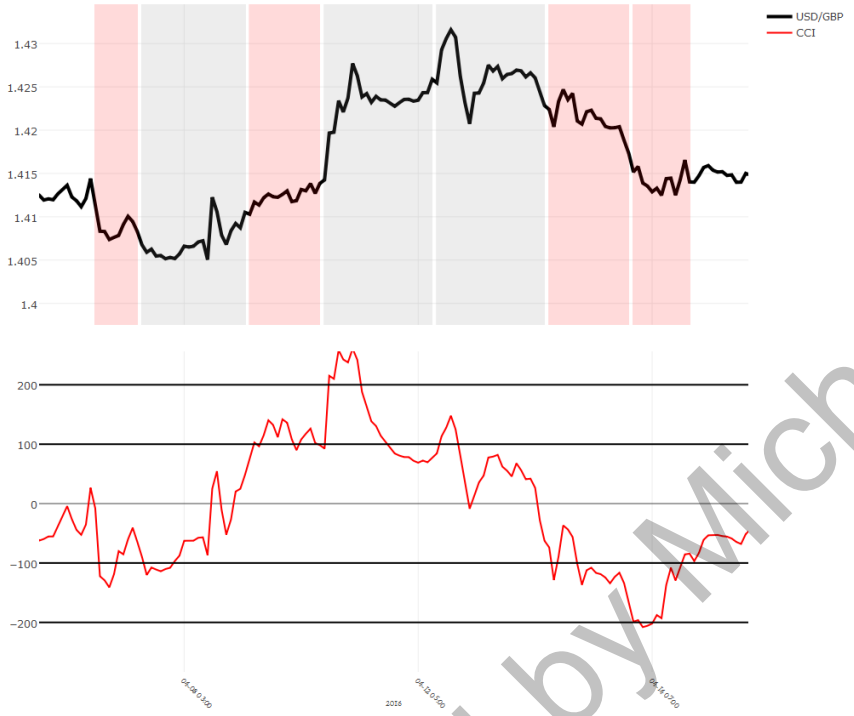
The ADL fluctuates between +1 and -1 and it rises (falls), when the Money Flow Multiplier is positive (negative). The Money Flow Multiplier is positive, when the closing price is in the upper half of the high-low range and is negative in the opposite case. Therefore, a high positive Money Flow Multiplier combined with a high Money Flow Volume will push the ADL up strongly, indicating a strong buying signal. Very similar to the arguments used by the OVB indicator, the ADL can be used to predict trend reversals or confirm trends.

G) Commodity Channel Index (CCI)

CCI is a versatile indicator that can be used to either warn of overbought or undersold levels or to identify new trends in time series. Because CCI measures the current price level relative to an average price level over a given period of time, CCI is relatively high when prices are above average. Correspondingly, CCI is relatively low when prices are below average. Usually 20 periods are used as the lookback horizon. CCI uses so called Typical Prices (TP`s), which are defined as: (High + Low + Close)/3. Simple moving averages (SMAs) subsequently smooth these TPs. CCI is calculated as follows: CCI = (TP – 20-period SMA of TPs)/(0.015 * Mean Deviation). Mean Deviation is calculated as the absolute sum of the 20 last differences between the TP and the 20-period SMA of the TPs divided by the total number of periods. The constant 0.015 was chosen by the developer Donald Lambert to ensure, that approximately 75% of the CCI values fall between -100 and +100. To indicate overbought or undersold levels, CCI values of ±200 are often used as relatively hard levels, meaning, that if CCI is above (below) +200 (-200), then prices tend to be overbought (undersold), indicating, that prices may soon reverse back to more modest levels. A starting trend can be indicated when CCI crosses a threshold of ±100, because if 75% of the prices lie between ±100, then a strong price move is necessary to break this line (often such lines are called resistant lines). A break through the +100 resistant line should therefore indicate a starting positive trend, and a break through the -100 resistant line should analogously indicate a starting negative

trend. The following figure illustrates the CCI and the resistant lines applied to the USD/GBP exchange rate.

Figure C.9: USD/GBP exchange rate, CCI, ±100 and ±200 resistant lines between 2016/04/06 20:00 and 2016/04/15 04:00 GMT



The figure shows the USD/GBP exchange rate (black bold line), CCI (red), ± 100 and ± 200 resistant lines (black horizontal lines) between 2016/04/06 20:00 and 2016/04/15 04:00 GMT. The first red shaded area corresponds to a successful trend signal triggered through a break through the -100 resistant line (successful because the exchange rate declined as in line with the assumption). The subsequent grey shaded area would be an example of a wrong signal, because there was a break through the -100 resistant line and subsequently, the exchange rate did not decline further. The second red shaded area is again an example of a successful trend signal (triggered through the break through the $+100$ resistant line). The second grey shaded area can be seen as an example of a wrong signal (because the $+200$ resistant line was breached and the prices did not reverse back to the mean – what is against the assumption, because a break through the $+200$ resistant line should signal overbought prices, which should then reverse back to the mean). The subsequent grey shaded area shows a wrong trend signal (break through $+100$ but no rise in prices). The last two red shaded areas are example of successful trend as well as trend reverse signals (break through -100 and -200 resistant line). *Source: Own simulation*

H) Relative Strength Index (RSI)

The RSI calculates a ratio of the recent upward price movement to the absolute price movement. RSI measures the speed and strength of price movements oscillating in a range between 0 and 100. The RSI is a normalized measure of the Relative Strength (RS). RSI and RS are calculated as

$$RSI = 100 - \frac{100}{1+RS} \tag{115a}$$

$$RS = \frac{Average\ Gain}{Average\ Loss} \tag{115b}$$

The RSI calculation is typically based on 14 periods. The first Average Gain is just a simple average over the past 14 periods, such that

$$Average\ Gain\ (Loss) = \frac{Sum\ of\ Gains\ (Losses)\ over\ past\ 14\ periods}{14} \tag{116}$$

The subsequent Average Gains (Loss) are calculated as

$$Average\ Gain\ (Loss) = \frac{Previous\ Average\ Gain\ (Loss)*13+Current\ Gain\ (Loss)}{14} \tag{117}$$

The RSI is zero, when the Average Gain is zero, indicating that the last 14 price movements all have been negative. Analogously, the RSI equals 100, when the last 14 price movements all have been positive. RSI levels of over 80 can indicate overbought securities and RSI levels of under 20 oversold securities. However, these signals would be wrong during strong up (down) trends, where the RSI can stay over 80 (under 20) for a long time. The RSI can also be used to indicate trend reversals. A positive trend reversal (turning from bearish into bullish markets) may occur, when the security records a lower low and the RSI on the same time forms a higher low. A negative trend reversal may occur for the opposite case where the security records a higher high and the RSI at the same time a lower high).

I) Moving Average Convergence/Divergence Oscillator (MACD)

A typical MACD line is calculated by the difference between a 12-period Exponential Moving Average (EMA) and a 26-period EMA. An EMA is calculated as

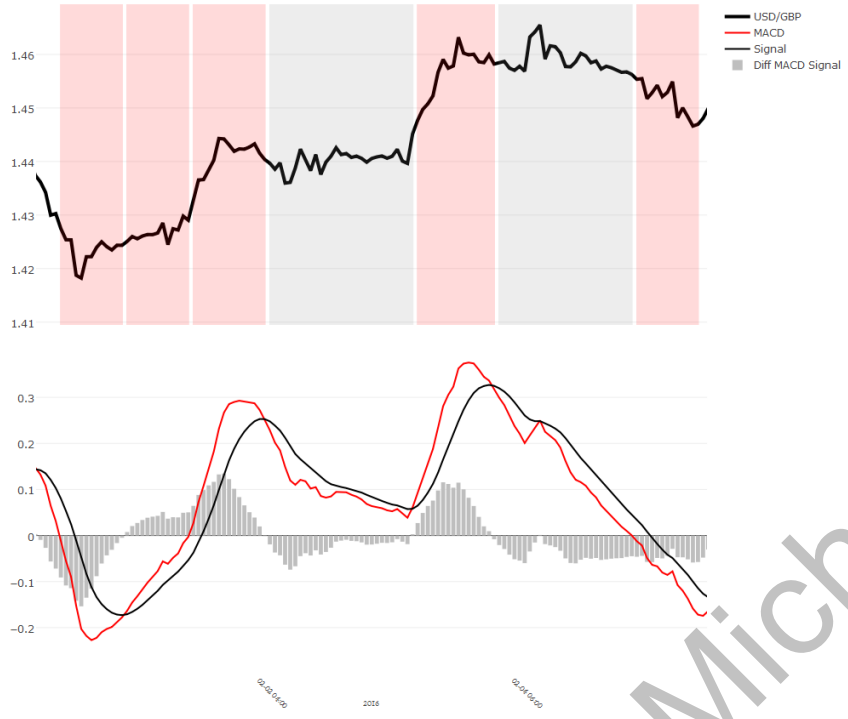
$$EMA_t = (Close_t - EMA_{t-1})\, x\, Multiplier + EMA_{t-1} \quad (118)$$

The Multiplier is calculated as

$$Multiplier = 2/(Time\ Period + 1) \quad (119)$$

The Multiplier weights the difference between the actual closing price and the previous EMA. If the multiplier is very low, then the actual EMA will be very similar to the previous EMA (and the EMA will be rigid). If the multiplier is very high, then the actual EMA will vary very strong from the previous EMA (and the EMA will be volatile). Because the Multiplier rises with a falling time period, a 12-period EMA will be more volatile than a 26-period EMA. As the MACD is the difference between a 12-period EMA and a 26-period EMA, the MACD will be positive, when the 12-period EMA is greater than the 26-period EMA, indicating that the most recent prices (smoothed on a shorter time horizon) have been higher than the smoothed prices on a longer time horizon. The MACD fluctuates around the zero line. In the following, the two most common signals, which are recovered from the MACD line, will be described shortly. A first signal can be derived, when the MACD crosses the zero line, for example from negative to positive, indicating, that prices are turning from having been lower on average to now beginning to be higher on average (on the short run). Consistently, MACD will remain positive (negative) as long as there is a sustainable upward (downward) trend. A second signal uses an additional Signal Line. The Signal Line is a 9-period EMA of the MACD line, which means that the Signal Line is a smoothed MACD line, which is lagging behind the original MACD line. In an ongoing trend, with a rising MACD line, the MACD line will not cross the Signal Line. If the MACD line turns from rising to falling, then MACD line will cross the lagging Signal Line if the change of the MACD reaches a substantial level. So the second signal derived from the MACD and the Signal Line is a crossing of the two lines, indicating, that the change in the MACD seems to be sustainable and could last longer. The following figure shows an illustration of the MACD and the Signal Line applied to the USD/GBP exchange rate.

Figure C.10: USD/GBP exchange rate, MACD and Signal Line between 2016/01/29 08:00 and 2016/02/05 20:00 GMT



Annotation: The figure shows the USD/GBP exchange rate (black bold line), the MACD line (red), the Signal Line (black) as well as the difference between the MACD and the Signal line (grey bars) between 2016/01/29 20:00 and 2016/02/05 20:00 GMT. The first red shaded region shows a successful signal triggered from a crossing of the MACD line with the zero line (successful because the following move has been negative if seen as a whole move until the next triggered signal). The second red shaded area as well as the third red shaded area also indicate successful signals (subsequent prices are rising) – the first signal triggered by a crossing of the MACD line with the Signal Line and the second signal triggered by a crossing of the MACD line with the zero line. The two grey shaded areas can be seen as wrong signals (subsequent prices are not falling but moving sideways). The last two red shaded area can be considered as a successful signal (subsequent prices are falling), triggered by a crossing of the MACD line with the Signal and the zero line. *Source: Own simulation*

J) Stochastic Oscillator (SO)

SO relates each period´s close price to the high-low range over the past n periods. The SO does not follow the price or the volume of a security, but the speed or momentum of the price changes. The basic assumption for the SO is, that momentum often changes direction before price. Therefore, the SO can be used to predict positive or negative trend reversals. Additionally, it also can be used to identify overbought or oversold price levels. The SO is calculated as

$$\%K = \frac{Current\ Close - Lowest\ Low}{Highest\ High - Lowest\ Low} * 100 \tag{120a}$$

$$\%D = Simple\ Moving\ Average\ of\ \%K\ over\ 3\ periods \tag{120b}$$

Where Lowest Low (Highest High) is the lowest (highest) value for the lookback period of typically 14 periods. The %D line is plotted alongside the %K line to act as a signal or trigger line. An SO above (below) 50 indicates, that the current close price is in the upper (lower) half of the Highest High – Lowest Low range of the last 14 periods. Accordingly, high (low) SO levels indicate, that the current close price is near its high (low) for the past 14 periods. Thereby, typically SO levels over 80 can indicate overbought securities, while SO levels below 20 can indicate oversold securities. However, similar to the RSI, these signals would be wrong during strong up (down) trends, where the SO can stay over 80 (under 20) for a longer time. A positive trend reversal may occur, when prices record a lower low but the SO

forms a higher low indicating less downside momentum (which may be a signal for a turning trend from bearish to bullish markets). Similarly, a negative trend reversal may occur, when prices record a higher high but the SO forms a lower high indicating less upside momentum (which may be signal for a turning trend from bullish to bearish markets) (StockCharts.com, 2017, see Technical Indicators)

Appendix D: Fundamental analysis of the exchange rate

In the following Appendix D, ***seven fundamental theories (A-G)*** about the determination of exchange rates are provided. Part ***A)*** deals with the Purchasing Power Parity (PPP), part ***B)*** with the Interest Parity (IP), part ***C)*** looks at the international trade, part ***D)*** describes the Mundell-Flemming model, part ***E)*** deals with a monetary approach with immediate price adjustments, part ***F)*** with a monetary approach with delayed price adjustments and the last part ***G)*** with an asset based approach to determine exchange rates. Thereby, the first and the second theory (PPP and IP) can be seen as basic concepts, which are also incorporated in the other theoretical explanations. In brackets, the variable(s) which determines the exchange rate according to the theories are given.

A) PPP (inflation)

There are ***two forms*** of the PPP, namely ***1)*** the absolute and ***2)*** the relative PPP.

1) Absolute PPP: The absolute PPP is based on the law of one price and it states, that in integrated world markets, every good should have the same price when exchange rates are taken into consideration. Therefore, due to arbitrage, the equation

$$P_t S_t = P_t^* \tag{121}$$

should hold, where P and P^* are national and foreign price indices and S is the exchange rate. Hence, if the absolute PPP holds, then the real exchange rate $\frac{P_t S_t}{P_t^*}$ should stay constant, such that $\frac{P_t S_t}{P_t^*} = 1$. In logarithmic terms, (121) becomes $s_t = p_t - p_t^*$.

2) Relative PPP: Because of transportation costs, import duties, different tax systems and trade barriers the law of one price may be violated for many goods. But in such cases, arbitrage still should provoke a relationship between the price changes, i.e. the inflation rates. Therefore, the relative PPP states that

$$\Delta P_t \Delta S_t = \Delta P_t^* \tag{122}$$

where ΔP_t and ΔP_t^* denotes the national and foreign inflation rates and ΔS_t the change of the exchange rate. For the relative PPP to be true, there must be a full substitution of the goods on the world markets[73].

In reality, there are several violations of the PPP, such that its assumptions are hard to measure and only may be detected in a weak form and on the long run. Such violations can be a changes in the

[73] Next to full substitution, the assumption of homogeneity of the monetary theory must hold. This assumption states, that monetary changes does not affect relative prices such that all prices are affected with the same proportion by the monetary change. Then, the terms of trades (the structures of import and export prices) are not affected.

trading relationships or changes in the economies which affects the terms of trades, an imperfect substitution of goods in the world markets, the so called Balassa-Samuelson-effect[74] or incomplete or delayed adjustments of prices (which again affects the terms of trades). The PPP is therefore not appropriate for short run predictions, whereas on the long run it can be a simple predictor (thereby, because of the persistence of the deviations from the PPP for many exchange rates, it is hard to determine, when exactly a correction to the PPP equilibrium may occur).

B) IP (interest rates)

The IP can be divided into ***two different considerations***, ***1)*** the covered IP (CIP) and ***2)*** the uncovered IP (UIP).

1) CIP: The CIP is based on the assumption, that an international financier has the choice to invest his money national, receiving an interest rate i_t, or to invest his money abroad, receiving an interest rate i_t^*. If the financier invests his money national, then his return would be $(1 + i_t)$ times the invested amount. If he would invest abroad, then, in time t, he would first need to change the money with the spot exchange rate S_t. In foreign currency, his investment would then return $\frac{(1+i_t^*)}{S_t}$ times the invested amount (denoted in foreign currency)[75]. If the investor covers his investment already in time t with a foreward exchange contract F_t (which gives him the right to change the foreign currency at time t+1 with a exchange rate F_t into his national currency), then his return, denoted in national currency, would be $\frac{(1+i_t^*)}{S_t} F_t$. In complete international capital markets, arbitrage should ensure, that the return of the national investment equals the return of the foreign investment (both denoted in national currency), such that

$$(1 + i_t) = \frac{(1+i_t^*)}{S_t} F_t \qquad (123)$$

In cases that the interest rates are small enough, the CIP is often expressed as[76] $i_t - i_t^* = f_t - s_t$, where small letters indicate logarithmic terms

[74] The Balassa-Samuelson-effect is best described with a small example: two countries A and B are producing tradable and non-tradable goods. A change of the productivity of the tradable goods in country A will have an affect on the wages of these workers. The prices of the tradable goods thereby stay unaffected, because the higher productivity has to be compensated with higher wages - such that there is no inflation and no changes in the exchange rates. But, if there are additionally uniform wage negotiations for the workers in the tradable and in the non-tradeable sector, then the increase in productivity in the tradable sector could also lead to higher wages in the non-tradable sector (without an increase in productivity, this would lead to higher prices, i,e. to inflation). Therefore, there would be inflation without a change of the exchange rate leading to a seemingly overvalued currency of country A and to a violation of the PPP.

[75] Where S_t is in direct quotation i.e. the unit is $\frac{foreign\ curreny}{national\ currency}$, meaning that the price for one unit of foreign currency is expressed directly, i.e. in national currency. For example, when Euroland = national and USA = foreign and an european financier would like to buy 1$, then he would have to pay S_t€. Therefore $S_t = \frac{\$}{€}$. This means at the same time, that if this investor would have 1€, he could change this to $\frac{1}{S_t}$$.

[76] $log\left((1 + i_t) = \frac{(1+i_t^*)}{S_t} F_t\right) \Leftrightarrow log(1 + i_t) = log(1 + i_t^*) + \log(F_t) - \log(S_t)$ and with $\log(1 + x) \approx x$ for small x, this becomes $i_t - i_t^* = f_t - s_t$.

2) UIP: For cases, that the international financier is not covering his investment with a foreward exchange contract, the financier bases his decision on the expected value of the exchange rate in the period t+1. Therefore, the UIP is defined as

$$(1 + i_t) = \frac{(1+i_t^*)}{S_t} S_{t+1}^e \qquad (124)$$

In logarithmic terms this becomes $i_t - i_t^* = s_{t+1}^e - s_t = \Delta s^e$. So far, national and foreign securities have been perfect substitutes. In case, that this assumption is not true, a risk premium l_t can be added to the expected vale to receive the foreward exchange rate, such that $s_{t+1}^e + l_t = f_t$. In this case, the logarithmic form of (124) will be $i_t - i_t^* = s_{t+1}^e + l_t - s_t$.

An inflow of foreign capital into a country (due to a higher national interest rate) may be to some extend damped in the future, because an increased amount of money will reduce the national interest rate. When considering the IP, it can be crucial to also regard the Fischer-equation $i_t = r_t + \pi_t$ in combination with the PPP. When the nominal interest rate is rising due to an increase in the real interest rate r_t, then the mechanism of the IP holds (predicting rising exchange rates) and the PPP stays disregarded. But if the increase in the nominal interest rate is due to inflation π_t, then the IP could be overcompensated by the PPP (which then predicts falling exchange rates at least on a longer run).

C) International trade (imports, exports)

One of the most fundamental and traditional approaches to determine exchange rates is based on the equilibriums in import and export markets. Thereby, the demand and the supply of a currency is determined by its demand and supply of its imports and exports. Hence, the demand of a currency is determined by the foreign demand for goods and services of that country (which in turn is determined by the income and the taste of the foreign consumers as well as the national factor prices). The supply of a currency is determined analogous. An increase in the foreign income for example, increases the demand for national products and this leads to an increase in the demand of the national currency, i.e. a decrease in the exchange rate (in direct quotation). In this case, the exports are rising relative to the imports, such that the net exports (exports minus imports) are increasing, i.e. the trade balance improves on the short run. On the long run however, the decreased exchange rate makes foreign imports cheaper and exports to the foreign country less valuable for national consumers and producers, such that the trade balance deteriorates (this is called the j-curve effect).

While there may be good intuitions in the international trade approach, there are several implicit assumptions which may not hold in reality. Complete competition with homogenous goods and services for example, or the assumption, that national and foreign goods are perfect substitutes. Furthermore, only the real economy matters, so that the approach misses to take international financial markets into account.

D) Mundell-Flemming model (current account)

The Mundell-Flemming model will only be depicted shortly for the sake of completeness, because it turns out, that it cannot explain most of the strong and seemingly random occurring changes of the exchange rate. Furthermore, the model does not take expectations into account, whereby financial markets in general as well as the FX market is heavily based on expectations. The Mundell-Flemming model is an extension of the IS-LM model for the open economy. Thereby, the goods, the money as well as the currency market are considered. The equilibrium of proposed savings and investments for an open economy is given by

$$Y = C(Y) + I(i) + G + X(R, Y^*) - R * IM(R, Y) \quad (125)$$

with $\frac{\partial C}{\partial Y} > 0, \frac{\partial I}{\partial i} < 0, \frac{\partial X}{\partial R} > 0, \frac{\partial X}{\partial Y^*} > 0, \frac{\partial X}{\partial IM} < 0, \frac{\partial IM}{\partial Y} > 0$ and where Y is output, C is consumption, I are investments, G stands for government expenditure, X are exports, IM are imports, I is the nominal interest rate and R is the real exchange rate. The money market equilibrium is given by the LM curve as

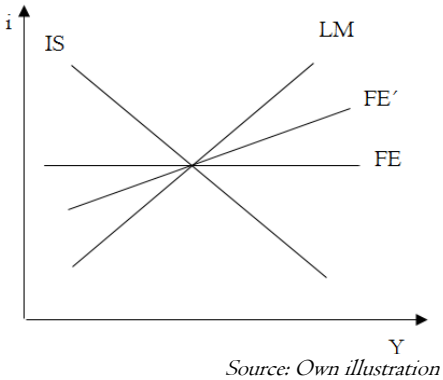
$$\frac{M}{L} = L(i, Y) \quad (126)$$

with $\frac{\partial L}{\partial Y} > 0$ and $\frac{\partial L}{\partial i} < 0$. At the exchange market, the balance of payment (BOP) consists of the trade balance (TB) and the capital account (CA), such that

$$BOP = TB + CA \quad (127)$$

with the dependencies $TB(R, Y, Y^*)$ and $CA(i, i^* + \Delta s^e)$. Further assumptions are that prices are fixed, i.e. $P = P^* = 1$, that capital is flowing into the country where the interest is higher and that national and foreign securities are perfect substitutes. With the assumption of full capital mobility, the UIP (without risk premium) $i_t = i_t^* + \Delta s^e$must hold. For static expectations $\Delta s^e = 0$, the curve of the exchange market (FE-curve) for a small open economy is then a horizontal line (because it is completely determined by the foreign interest rate i_t^*). In the case of incomplete capital mobility, the FE-curve (denoted with FE´ on the right figure) has a positive slope because for example, if the output growth (and the TB falls therefore), then the CA needs to compensate the decline of the TB (the CA needs to rise). In order for the CA to increase, i must rise to attract foreign money to flow into the country.

Figure D.1: IS-LM-FE equilibrium



Source: Own illustration

E) The monetary approach with immediate price adjustments (money demand, output and interest rates and their expectations)

This approach is based on the PPP as well as on a theory for the determination of the price levels (used by the PPP). The assumptions are: prices are completely flexible, national and foreign securities are perfect substitutes, complete mobility of capital, national currency is only held by national financiers, foreign

currency is only held by foreign financiers, the money supply is determined autonomously and real income is exogenous. The model is based on the absolute PPP

$$s_t = p_t - p_t^* \tag{128}$$

Because national (foreign) currency is only held by national (foreign) financiers, the real demand can be expressed in logarithmic terms as

$$m_t^d - p_t = k + \alpha_1 y_t - \alpha_2 i_t \tag{129a}$$

$$m_t^{d*} - p_t^* = k^* + \alpha_1^* y_t^* - \alpha_2^* i_t^* \tag{129b}$$

with m^d as nominal money demand, p as the price level, k as scaling variable, α_1, α_1^* as elasticity of the output, α_2, α_2^* as semi-elasticity of the money demand (whereby $\alpha_1, \alpha_1^*, \alpha_2, \alpha_2^* > 0$), y as output and i as nominal interest rate. Under the assumption, that $\alpha_1 = \alpha_1^*, \alpha_2 = \alpha_2^*$ and that the national and foreign nominal money demand is exogenous, (128), (129a) and (129b) can be combined to the fundamental equation of the model

$$s_t = (k - k^*) + (m_t - m_t^*) - \alpha_1(y_t - y_t^*) + \alpha_2(i_t - i_t^*) \tag{130}$$

In this model, the exchange rate is determined by the relative money demand, the relative output as well as the relative interest rates. Because such a model does not take expectations into account and therefore can not explain the volatility of most exchange rates, it can be extended with the UIP $i_t - i_t^* = s_{t+1}^e - s_t$ to

$$s_t = (k - k^*) + (m_t - m_t^*) - \alpha_1(y_t - y_t^*) + \alpha_2(s_{t+1}^e - s_t) \tag{131}$$

This then can be reformulated to

$$s_t = \frac{1}{1+\alpha_2} \sum_{i=0}^{\infty} \left(\frac{\alpha_2}{1+\alpha_2}\right)^i [(m_{t+i}^e - m_{t+i}^{e*}) - \alpha_1(y_{t+i}^e - y_{t+i}^{e*}) - (k - k^*)] \tag{132}$$

Such a model, in which the exchange rate is now not only determined by the relative money demand, the relative output as well as the relative interest rates, but also by their expected values, would be able to explain highly volatile exchange rates with changes in the expectations. Thereby it is decisive, if financiers regard changes of the money demand, the output or the interest rates as long-lasting and sustainable or as transitory and temporary.

F) The monetary approach with delayed price adjustments (lagged money demand, output and interest rates)

One main point of criticism of the aforementioned model is the assumption of fully flexible prices. Dornbusch therefore proposed a model, where at the short run violations of the PPP are possible, while on the long run, the PPP holds (due to price adjustment costs or incomplete information). The UIP however hold permanently – due to the fast adjustments in financial markets. The equations (128), (129a) and (129b) still holds in the model with delayed price adjustments whereas (130) is replaced with

$$\bar{s}_t = \bar{p}_t - \bar{p}_t^* \tag{133}$$

where $\bar{s}_t$ is the long term equilibrium exchange rate and $\bar{p}_t$ and $\bar{p}_t^*$ are the long term equilibrium price levels. Furthermore it is assumed, that the expected change of the exchange rate is proportional to the difference between the actual from the long term equilibrium exchange rate, such that

$$s_{t+1}^e = \beta(\bar{s}_t - s_t) \tag{134}$$

where $\beta > 0$. Regarding the price adjustment, it is assumed, that they react with a lag of one period to the conditions on the goods market

$$(p_t - p_t^*) - (p_{t-1} - p_{t-1}^*) = \delta\left[\left(y_{t-1}^d - y_{t-1}^{d*}\right) - (y_{t-1} - y_{t-1}^*)\right] \tag{135}$$

with $\delta > 0$ and

$$\left(y_t^d - y_t^{d*}\right) = c + \gamma(y_t - y_t^*) - \sigma(i - i^*) + \zeta(s_t - p_t + p_t^*) \tag{136}$$

what means that the demand at the goods market is dependent from the relative output, the relative interest rate as well as from the terms of trade.

(132), (133), (134), (135), and (136) can be combined to obtain the final equation of the model

$$s_t = c_0 + c_1 s_{t-1} + c_2(m_t - m_t^*) + c_3(m_{t-1} - m_{t-1}^*) + c_4(y_t - y_t^*) + \cdots$$
$$\ldots c_5(y_{t-1} - y_{t-1}^*) + \ldots c_6(p_{t-1} - p_{t-1}^*)^{77} \tag{137}$$

what shows that such a model would take the actual as well as the past relative money demand, output as well as the past relative price levels into account.

G) The asset base approach (demand and supply of foreign and national securities)

The asset base approach assumes, that the asset (W) of national financiers consists of national currency (M), national securities (B) and foreign securities (F), such that

$$W = M + B + SF \tag{138}$$

whereby foreign securities are multiplied by the exchange rate S to denominate them in national currency (the assumption that national financiers do not hold foreign currency still holds in this model). Additionally, the UIP with risk premium $i_t - i_t^* = s_{t+1}^e + l_t - s_t$ holds. For the assets, the following dependencies are assumed

$$M^d = m(i, i^* + \Delta s^e, W) \text{ with } m_i < 0, m_{i^*+\Delta s^e} < 0, m_W > 0 \tag{139a}$$

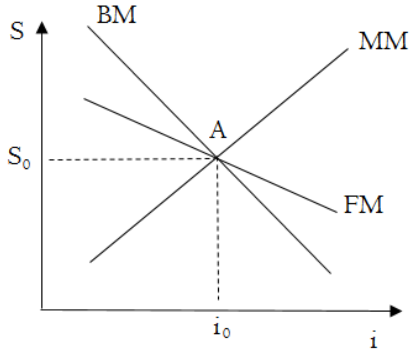
$$B^d = b(i, i^* + \Delta s^e, W) \text{ with } b_i > 0, b_{i^*+\Delta s^e} < 0, b_W > 0 \tag{139b}$$

$$SF^d = f(i, i^* + \Delta s^e, W) \text{ with } f_i < 0, f_{i^*+\Delta s^e} > 0, f_W > 0 \tag{139c}$$

[77] With $c_0 = -(k - k^*)\left(1 + \frac{1}{\alpha_2 \beta}\right) + \left(\frac{\delta\sigma(k-k^*)}{\alpha_2} - \delta c\right)\frac{1}{\alpha_2 \beta}$; $c_1 = -\frac{\zeta\delta}{\alpha_2 \beta}$; $c_2 = \left(1 + \frac{1}{\alpha_2}\right)$; $c_3 = -\frac{\sigma\delta}{(\alpha_2)^2 \beta}$
$c_4 = -\alpha_1\left(1 + \frac{1}{\alpha_2}\right)$; $c_5 = \frac{\delta}{\alpha_2 \beta}\left(1 - \gamma + \frac{\alpha_1 \sigma}{\alpha_2}\right)$; $c_6 = -\frac{1}{\alpha_2 \beta}\left(1 - \frac{\delta\sigma}{\alpha_2} - \delta\zeta\right)$

With the further assumptions, that the money supply M^S is exogenously determined by the central bank, that national securities are exclusively demanded by national financiers, that the amount of national securities is exogenous and the amount of foreign securities is fixed on the short run and that expectations are static, i.e. $s^e = 0$, the model can be displayed in a S-i-diagram as shown on the left. Thereby, the following intuitions can be derived: moving from point A to the right means an increase of the national interest rate. With an unchanged amount of assets this means, that the demand for money decreased. The basic equation $W = M + B + SF$, where M is now smaller, but W, B and F are unchanged, only can hold, when the exchange rate rises (this explains the rising slope of the money market curve (MM)). The falling slope for the market of national securities (BM) can be derived similarly. Moving from point A to the right, i.e. increasing the national interest rate, leads to a higher demand of national securities. Unchanged assets and increased national securities (B) in $W = M + B + SF$ are just possible with a falling exchange rate. The slope of the last curve, the curve for the equilibrium of foreign securities in combination with the exchange rate (FM), also can be derived argumentative with a move from point A to the right. An increase of the national interest rate means ceteris paribus a decrease in the demand for foreign securities (F). In order to keep the combined asset SF in $W = M + B + SF$ constant, because all assets are fixed on the short run per assumption, the exchange rate has to fall to enhance the demand for foreign securities again (Gerhards, 1994, p. 19-55).

Figure D.2: S-i-diagram



Source: Own illustration

Appendix E: Empirical Results

E.1 Text mining with Multinomial Naïve Bayes Classifier

Table E.1.1: Contingency table of Multinomial Naïve Bayes Classifier for Method 1

	Predicted +	Predicted -	Total
Actual +	69	338	407
Actual -	58	365	423
Total	127	703	830

Annotation: The contingency table shows that the sum of true positive and true negative predictions is just 434 out of 830 forecasts, leading to an overall accuracy of 52%. The predictions also seem to be biased, because negative predictions are more than five time more frequent than positive (703 to 127, while the actual positive and negative changes of the exchange rate seem to be quite equally distributed with 407 to 423).

Source: Own calculations

Table E.1.2: Contingency table of Multinomial Naïve Bayes Classifier for Method 2

	Predicted +	Predicted -	Total
Actual +	167	110	277
Actual -	197	116	313
Total	364	226	590

Annotation: The contingency table shows that the sum of true positive and true negative predictions is just 283 out of 590 forecasts, leading to an overall accuracy of 48%. This time, the predictions seem to be biased just a little bit in favour of positive predictions.

Source: Own calculations

E.2 Data mining and technical univariate analysis

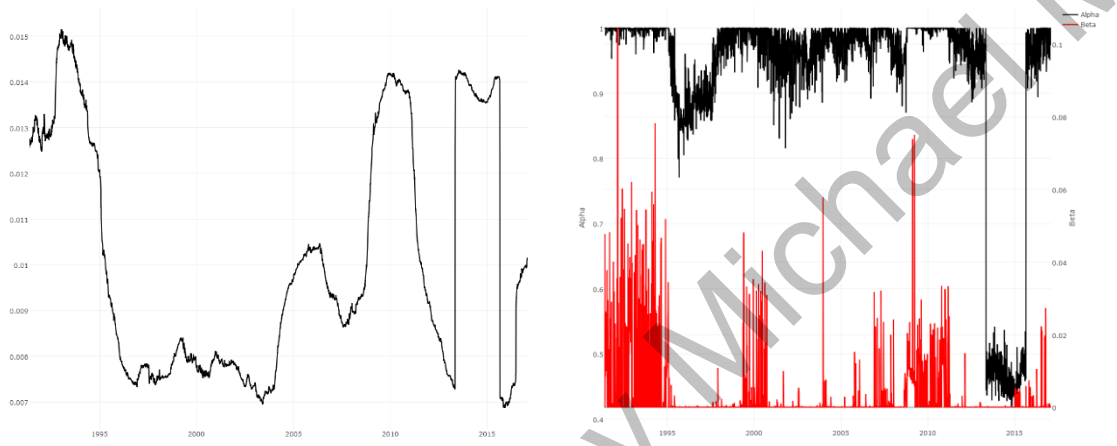
Table E.2.1: Contingency table of forecasts made with Trend Adjusted ES models

	Predicted +	Predicted -	Total
Actual +	1733	1693	3426
Actual -	1540	1604	3144
Total	3273	3297	6570

Annotation: The contingency table shows that the sum of true positive and true negative predictions is just 3337 out of 6570 forecasts, leading to an overall accuracy of 51%.

Source: Own calculations

Figure E.2.1: RMSE on the left as well as α and β values for the fitted Trend Adjusted ES models



Annotation: The left figure shows the RMSEs of the 6570 models over time. Thereby, the RMSE fluctuates between 0.007 and 0.015 with a relatively long period between 1996 and 2003, where the fit of the model was constantly better than in the remaining periods. The right figure shows the α and β values of the fitted models. Thereby, the α value of the model is in the most cases between 0.9 and 1, with a small period between 2013 and 2015, where the models seemed to change with α values around 0.5. Except of these small period, the forecast is made up almost totally out of the last observation, such that the exponentially smoothed time series has very often no substantial influence on the forecasts. The same is true for the trend component. The β values are very low and fluctuate for the complete series between around 0.1 and 0. Therefore, the Trend Adjusted ES models very often are similar to those fitted to a random walk with a slightly drift component.

Source: Own calculations

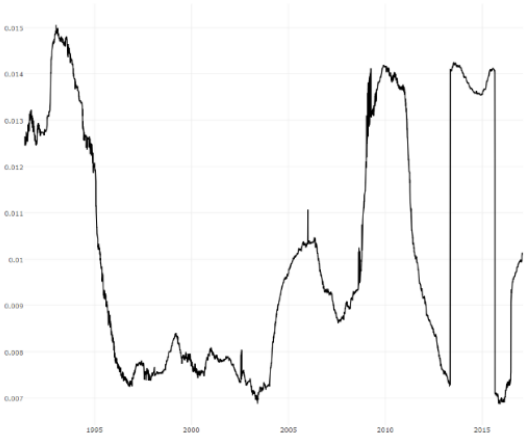
Table E.2.2: Contingency table of forecasts made with ARIMA models

	Predicted +	Predicted -	Total
Actual +	1693	1702	3395
Actual -	1580	1595	3175
Total	3273	3297	6570

Annotation: The contingency table shows that the sum of true positive and true negative predictions is just 3288 out of 6570 forecasts, leading to an overall accuracy of 50%. Therefore, also the forecasts made with ARIMA models could not improve the prediction performance in forecasting the USD/GBP exchange rate.

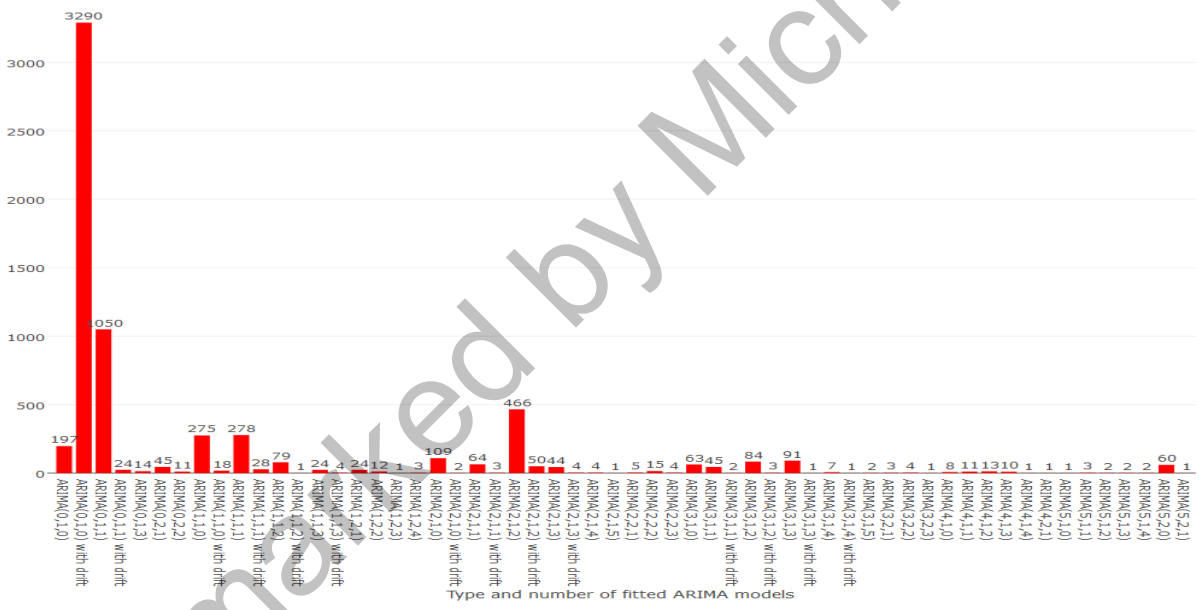
Source: Own calculations

Figure E.2.2: RMSE of fitted ARIMA models over time



Annotation: The left figure shows the RMSEs of the 6570 fitted ARIMA models. Thereby, the RMSE fluctuates between 0.007 and 0.015 with a relatively long period between 1996 and 2003, where the fit of the model was constantly better than in the remaining periods. This figure is very similar to the RMSEs over time for the Trend Adjusted ES models. As shown in the next Figure E.2.5, the fast majority of ARIMA models have been ARIMAs(0,1,0)(wd). As shown in Figure E.2.2, the fitted Trend Adjusted ES models are also very often similar to random walks with a slightly drift component. Therefore, the two figures are very similar. *Source: Own calculations*

Figure E.2.3: Type and number of fitted ARIMA models



Annotation: The bar chart shows the types and the numbers of the fitted ARIMA models. Thereby it can be seen, that all chosen time series have been differenced at least one time (i.e. the Hyndman-Khandakar algorithm did not detect one stationary time series in levels). Also, the algorithm mostly chose quite small models (out of 6570 models, 4537 have been ARIMAs(0,1,0)(nd), ARIMAs(0,1,0)(wd) or ARIMAs(0,1,1)). *Source: Own calculations*

E.3 Data mining and technical multivariate analysis

Table E.3.1: List of used features for SVM and NN in technical multivariate analysis

Features:	Used days for calculation:
1-day difference of exchange rate	-
2-day difference of exchange rate	-
3-day difference of exchange rate	-
Average True Range	14
Aroon	20
On Balance Volume	-
Bollinger Bands	20
Chaikin Accumulation/Distribution	-
Close Location Value	-
Chande Momentum Oscillator	14
Commodity Channel Index	20

Features:	Used days for calculation:
Relative Strength Index	14
MACD Oscillator	12
Stochastic Oscillator	14
Chaikin Volatility	10
Exponentially weighted mean	10
Volatility Indicator	10
Money Flow Index	14
Parabolic Stop and Reverse	-
Mean over 10 days	10
Standard deviation of last 10 days	10

Source: Own illustration

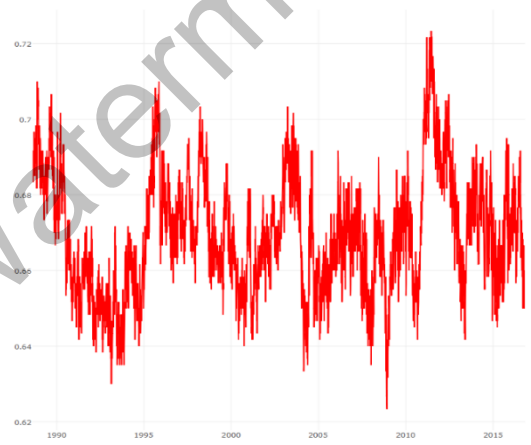
Table E.3.2: Contingency table of forecasts made with SVM

	Predicted +	Predicted -	Total
Actual +	2133	2125	4258
Actual -	1840	1839	3679
Total	3973	3964	7937

Annotation: The contingency table shows that the sum of true positive and true negative predictions is just 3972 out of 7937 forecasts, leading to an overall accuracy of 50%. For the SVMs a radial kernel with $\gamma = 0.1$ and a cost C of 1 have been chosen.

Source: Own calculations

Figure E.3.1: In sample accuracy of fitted SVMs over time



Annotation: The figure shows the in sample accuracy (sum of true positives and true negatives divided by the observations of the training data set) of all fitted SVMs over time. Thereby, the accuracy fluctuates between 0.62 and 0.73 meaning that around 60 to 70% of the training observations have been classified correctly.

Source: Own calculations

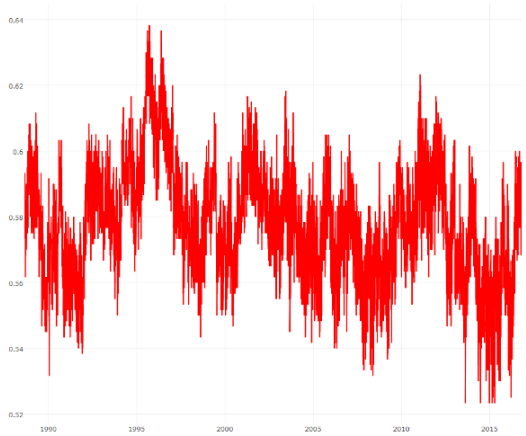
Table E.3.3: Contingency table of forecasts made with NN

	Predicted +	Predicted -	Total
Actual +	2031	1942	3973
Actual -	1954	2010	3964
Total	3985	3952	7937

Annotation: The contingency table shows that the sum of true positive and true negative predictions is just 4041 out of 7937 forecasts, leading to an overall accuracy of 51%. Thereby, the model parameter of the fitted NN have been chosen as follows: the network architecture have been two hidden layers with 20 and 10 neurons per layer. A hyperbolic tangent activation function as well as a learning rate of 0.3 has been chosen.

Source: Own calculations

Figure E.3.2: In sample accuracy of fitted NNs over time



Annotation: The figure shows the in sample accuracy (sum of true positives and true negatives divided by the observations of the training data set) of all fitted NNs over time. Thereby, the accuracy fluctuates between 0.52 and 0.64 meaning that around 50 to 65% of the training observations have been classified correctly. Therefore, the in sample fit for the fitted NNs is around 10% smaller than the in sample fit of the SVNs, while the out of sample predictions are with 50 and 51% for both techniques very similar.

Source: Own calculations

E.4 Data mining and fundamental analysis

Table E.4.1: ADF tests for 2 month LIBOR (USD), 3 month LIBOR (GBP) as well as the USD/GBP, CAD/GBP and CHF/GBP

	$H_0: (\beta_1, \beta_2, \pi) = (\beta_1, 0, 0)$		$H_0: (\beta_1, \beta_1, \pi) = (0, \beta_1, \pi)$		$H_0: \beta_1 = 0$	
	Test statistic	Critical value	Test statistic	Critical value	Test statistic	Critical value
2 month LIBOR (USD),	1.27	8.27	1.37	6.09	1.02	6.43
3 month LIBOR (GBP)	1.20	8.27	1.97	6.09	2.12	6.43
USD/GBP,	3.64	8.27	2.46	6.09	1.98	6.43
CAD/GBP	2.44	8.27	2.05	6.09	0.71	6.43
CHF/GBP	3.84	8.27	2.59	6.09	2.78	6.43

Source: Own calculations

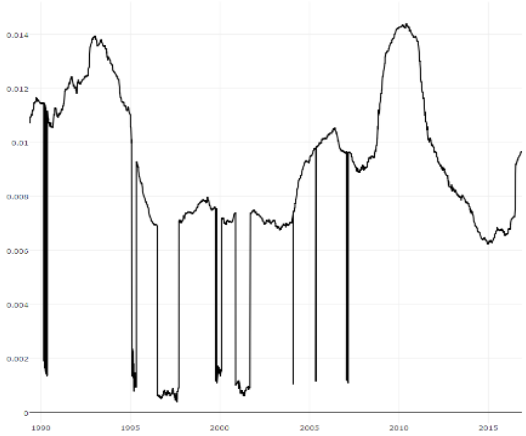
Table E.4.2: Contingency table of forecasts made with VAR/VEM model

	Predicted +	Predicted -	Total
Actual +	1829	1769	3598
Actual -	1679	1658	3337
Total	3508	3427	6935

Annotation: The contingency table shows that the sum of true positive and true negative predictions is just 3487 out of 6935 forecasts, leading to an overall accuracy of 50%.

Source: Own calculations

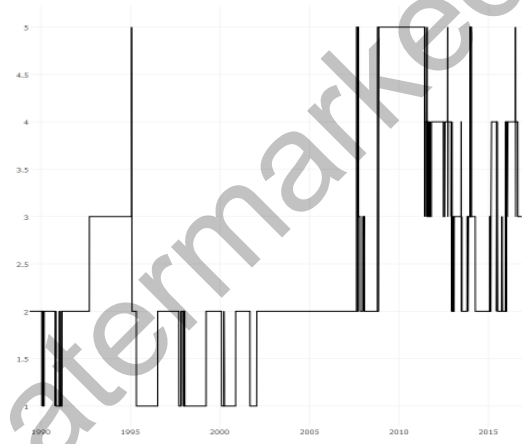
Figure E.4.1: RMSE of fitted VAR/VEC models over time



Annotation: The left figure shows the RMSE of the fitted VAR or VEC models over time. Thereby, the RMSE fluctuated between 0.001 and 0.014. Compared to the Trend Adjusted ES and ARIMA models, the RMSE for the fitted values reaches for the most time similar values. But for some periods the RMSE for the VAR/VEC models have been a lot smaller (reaching for the best models values around 0.001 compared to the values of the best Trend Adjusted ES and ARIMA models of 0.007).

Source: Own calculations

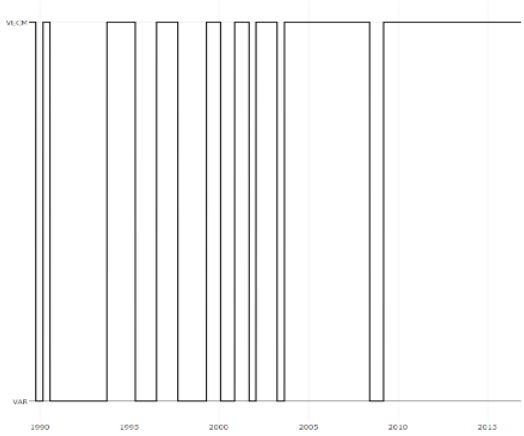
Figure E.4.2: Chosen lag orders of fitted VAR/VEC models over time



Annotation: The left figure shows the chosen lag orders of the fitted VAR/VEC models over time. Thereby, the first half of the test data set was almost completely fitted by models with lag orders one or two, whereas in the second half of the test data set, a lot of more complex models with lag orders between two and five have been fitted.

Source: Own calculations

Figure E.4.3: Chosen models (VAR vs VEC) over time



Annotation: The left figure shows the chosen models (VAR vs VEC) over time. Thereby, every 100th observation a Johansen Test for co-integration was applied to check, whether a VAR or a VEC model should be fitted. In the first half of the testing data set, there seem to be almost as much VAR as VEC models, whereas in the second half of the testing data set, almost only VEC models have been fitted.

Source: Own calculations

Table E.4.3: Features for fundamental analysis

CPI USA (m)*	Spot rate Canadian Dollar/GBP (d)	Overnight LIBOR based on GBP (d)	Close Price HSBC (d)	Close Price NYSE Amex Composite Index (d)
CPI UK (m)*	Spot rate Swiss Franc/USD (d)	1 week LIBOR based on USD (d)	Close Price Lloyds Banking Group (d)	Close Price Hang Seng Index (d)
Inflation USA (m)*	Spot rate Swiss Franc/GBP (d)	1 week LIBOR based on GBP (d)	Close Price Barclays (d)	Close Price IBOVESPA (d)
Inflation UK (m)*	Spot rate New Zealand Dollar/USD (d)	Overnight LIBOR based on USD (d)	Close Price Standard Chartered (d)	Close Price TA-125 (d)
Imports USA (m)*	Spot rate New Zealand Dollar/GBP (d)	US AAA rated Bond Index (d)	Close Price JPMorgan Chase (d)	Close Price SSE Composite Index (d)
Exports USA (m)*	Spot rate Australian Dollar/USD (d)	US BBB rated Bond Index (d)	Close Price Bank of America (d)	Close Price Russell 2000 Index (d)
Import UK (m)*	Spot rate Australian Dollar/GBP (d)	US CCC rated Bond Index (d)	Close Price Wells Fargo & Company (d)	Dow Jones Industrial Average (d)
Exports UK (m)*	Spot rate Euro/USD (d)	US High Yield Corporate Bond Index (d)	Close Price Citigroup (d)	Close Price Nasdaq Composite (d)
1 month LIBOR based on GBP (d)	Spot rate Euro/GBP (d)	Spot rate Japanese Yen/USD (d)	Close Price Goldman Sachs Group (d)	Close Price Stoxx 50 Europe (d)
12 month LIBOR based on GBP (d)	Close Price FTSE 100 (d)	Spot rate Japanese Yen/GBP (d)	Close Price Dax 30 (d)	Close Price NYSE Composite (d)
3 month LIBOR based on GBP (d)	Close Price S&P 500 (d)	Spot rate Canadian Dollar/USD (d)	Close Price Nikkei 225 (d)	Close Price CAC 40 (d)
2 month LIBOR based on USD (d)	1 day lagged spot rate USD/GBP (d)			

Annotation: d = daily, m = monthly data. The time series marked with * are taken in their untransformed form. Because this data are monthly data, the values of theses time series are constant for most days. Because of their importance according to the theoretical approaches for the determination of exchange rates, they have been nevertheless incorporated into the feature set. From the remaining 49 features, for every time series 10 differences have been calculated, such that the first differences have been calculated from observation at time t and time t-1, the second differences have been calculated from observations at time t and t-2 and so forth (up to the 10th differences). Therefore, the fundamental analysis used 49*10 differences plus 8 untransformed = 490+8=498 features.

Source: Own illustration

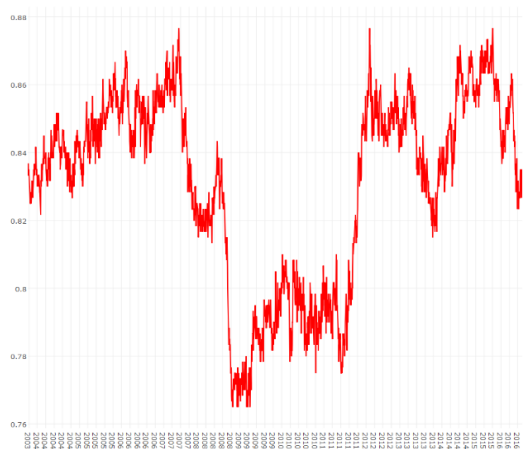
Table E.4.4: Contingency table of forecasts made with SVM models

	Predicted +	Predicted -	Total
Actual +	767	728	1495
Actual -	542	544	1086
Total	1309	1272	2581

Annotation: The contingency table shows that the sum of true positive and true negative predictions is just 1311 out of 2581 forecasts, leading to an overall accuracy of 51%.

Source: Own calculation

Figure E.4.4: In sample accuracy of fitted SVMs over time



Annotation: The figure shows the in sample accuracy (sum of true positives and true negatives divided by the observations of the training data set) of all fitted SVMs over time. Thereby, the accuracy fluctuates between 0.76 and 0.88 meaning that around 75 to 90% of the training observations have been classified correctly. Despite the fact, that the in sample fits reach very high levels, the out of sample predictions are with 51% not better than for the other fitted models. *Source: Own calculation*

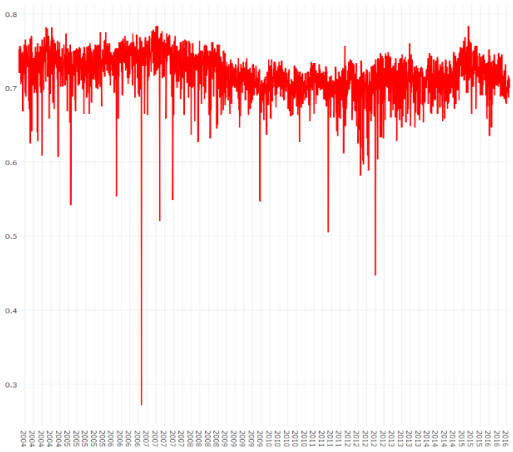
Table E.4.5: Contingency table of forecasts made with NN models

	Predicted +	Predicted -	Total
Actual +	727	582	1309
Actual -	655	617	1272
Total	1382	1199	2581

Annotation: The contingency table shows that the sum of true positive and true negative predictions is just 1344 out of 2581 forecasts, leading to an overall accuracy of 52%.

Source: Own calculation

Figure E.4.5: In sample accuracy of fitted NNs over time



Annotation: The figure shows the in sample accuracy (sum of true positives and true negatives divided by the observations of the training data set) of all fitted NNs over time. Thereby, the accuracy fluctuates for the most time between 0.60 and 0.80 meaning that around 60 to 80% of the training observations have been classified correctly. Despite the fact, that the in sample fits reach high levels, the out of sample predictions are with 52% not substantially better than for the other fitted models.

Source: Own calculation

Bibliography

Aaese, Kim-Georg (2011): Text Mining of News Articles for Stock Price Predictions. Norwegian University of Science and Technology. Master-Thesis. Available online at https://daim.idi.ntnu.no/masteroppgaver/006/6012/tittelside.pdf

Brown, Eric D. (2014): Drowning in Data, Starved for Information. Available online at http://ericbrown.com/drowning-in-data-starved-for-information.htm, checked on 1/18/2017.

Caffo, Brian (2016): Brian Machine Learning vs Traditional Statistics Part 1. Edited by www.youtube.com. Available online at https://www.youtube.com/watch?v=788WrMQo1wY, checked on 1/18/2017.

Campbell, Colin; Ying, Yiming (2011): Learning with support vector machines (Synthesis lectures on artificial intelligence and machine learning). Available online at http://dx.doi.org/10.2200/S00324ED1V01Y201102AIM010.

Cowpertwait, Paul S.P.; Metcalfe, Andrew V. (2009): Introductory time series with R. Dordrecht: Springer (Use R). Available online at http://dx.doi.org/10.1007/978-0-387-88698-5.

Fayyad, Usama M.; Gregory, Piatetsky-Shapiro; Padhraic, Smyth (1996): Knowledge discovery and data mining: towards a unifying framework. In *KDD. Vol. 96*.

Flach, Peter A.; Bie, Tijl de; Cristianini, Nello (2012): Machine learning and knowledge discovery in databases. European conference, ECML PKDD 2012, Bristol, UK, September 24 - 28, Berlin: Springer. Available online at http://dx.doi.org/10.1007/978-3-642-33486-3.

Gerhards, Tilmann (1994): Theorie und Empirie flexibler Wechselkurse. Pysica-Verlag Heidelberg.

Gruber, Antje Birgit (2011): Kointegration in Theorie und Praxis. Statistische Analyse gemeinsamer Entwicklungstrends in psychologischen Zeitreihensystemen.

Hendry, David F.; Jusélius, Katarina (2000): Explaining cointegration analysis, Part 1. In *The energy journal* 21 (1), pp. 1–42.

Hendry, David F.; Jusélius, Katarina (2001): Explaining cointegration analysis, Part 2. In *The energy journal* 22 (1), pp. 75–120.

Hyndman, Rob J. (2012): Forecasting: principles and practice. Edited by www.otexts.org. Available online at https://www.otexts.org/fpp/7, checked on 1/18/2017.

James, Gareth; Witten, Daniela; Hastie, Trevor; Tibshirani, Robert (2013): An Introduction to Statistical Learning. Springer Science+Business Media; Springer New York, Heidelberg, Dordrecht London

Kampakis, Stelios (2015): Statistics vs Machine Learning: The two worlds. Edited by www.skampakis.com. Available online at http://www.skampakis.com/statistics-vs-machine-learning-the-two-worlds/, checked on 1/18/2017.

Kriesel, David (2005): Ein kleiner Überblick über Neuronale Netze. Availabel online at http://www.dkriesel.com/_media/science/neuronalenetze-de-zeta2-2col-dkrieselcom.pdf

Lang, Carsten (2005): Theoretische und empirische Aspekte der Prognose wichtiger makroökonomischer Größen. Zugl.: Gießen, Univ., Diss., 2005. 1. Aufl. Göttingen: Cuvillier.

Lantz, Brett (2013): Machine learning with R. Learn how to use R to apply powerful machine learning methods and gain an insight into real-world applications. 1. publ. Birmingham u.a.: Packt Publ.

Lütkepohl, Helmut; Krätzig, Markus (2004): Applied Time Series Econometrics. Cambridge, New York, Melbourne: Cambridge University Press

Mazzoni, Thomas (2015): Zeitreihenanalyse. FernUniversität Hagen. Available online at http://www.fernuni-hagen.de/imperia/md/content/ls_statistik/zeitreihenskript_als_ke2.pdf, checked on 1/23/2017.

Mills, Terence C.; Markellos, Raphael N. (2008): The econometric modelling of financial time series. 3rd ed. New York, Cambridge, UK: Cambridge University Press. Available online at http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10221546.

Nagpaul, P. S. (2005): Time series analysis in WinIDAMS. In *www.portal.unesco.org*. Available online http://portal.unesco.org/ci/fr/files/18650/11133194701TimeSeriesAnal.pdf/TimeSeriesAnal.pdf.

Neusser, Klaus (2011): Zeitreihenanalyse in den Wirtschaftswissenschaften. 3., überarbeitete Auflage. Wiesbaden: Vieweg+Teubner Verlag / Springer Fachmedien Wiesbaden GmbH Wiesbaden. Available online at http://dx.doi.org/10.1007/978-3-8348-8653-8.

Nielsen, Michael (2017): Neural Networks and Deep Learning. Determination Press. Available online at http://neuralnetworksanddeeplearning.com/index.html

Pfaff, Bernhard; Gentleman, Robert; Hornik, Kurt; Parmigiani, Giovanni (2008): Analysis of Integrated and Cointegrated Time Series with R. 2nd ed. New York, NY, Heidelberg: Springer (Use R!). Available online at http://d-nb.info/990764168/34.

Schneider, Matti; Mentemeier, Sebastian (2010): Zeitreihenanalyse mit R. Available online at file:///C:/Users/DE-94010/Downloads/Zeitreihenanalyse.pdf

Shah, Aatash (2016): Machine Learning vs. Statistics. Edited by www.kdnuggets.com. Available online at http://www.kdnuggets.com/2016/11/machine-learning-vs-statistics.html, checked on 1/18/2017.

Sheppard, Kevin (2010): Financial econometrics notes. New York, NY: University of Oxford.

Shimodaira, Hiroshi (2015): Text Classification using Naïve Bayes. Availabe online at http://www.inf.ed.ac.uk/teaching/courses/inf2b/learnnotes/inf2b-learn-note07-2up.pdf

Srivastava, Tavish (2015): Difference between Machinie Learning & Statistical Modeling. Edited by www.analyticsvidhya.com. Available online at https://www.analyticsvidhya.com/blog/2015/07/difference-machine-learning-statistical-modeling/, checked on 1/18/2017.

Stier, Winfried (2001): Methoden der Zeitreihenanalyse. Berlin, Heidelberg: Springer (Springer-Lehrbuch). Available online at http://dx.doi.org/10.1007/978-3-642-56709-4.

StockCharts.Com (2017): Technical Indicators and Overlays. Available online at http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators

Torgo, Luis (2011): Data Mining with R. Boca Raton, London, New York: Chapman & Hall/CRC Press Book.

Vogel, Jürgen (2015): Prognose von Zeitreihen. Eine Einführung für Wirtschaftswissenschaftler. Wiesbaden: Springer Gabler.

Weiss, Sholom; Indurkhya, Nitin; Zhang, Tong (2010): Fundamentals of Predictive Text Mining. Springer-Verlag London.

Zivot, Eric; Wang, Jiahui (2006): Modeling Financial Time Series with S-PLUS®. 1. Aufl. s.l.: Springer-Verlag. Available online at http://gbv.eblib.com/patron/FullRecord.aspx?p=264861

Statement of Originality

I, Michael Meier, declare that I have authored this thesis independently, that I have not used other than the declared sources and that I have explicitly marked all material which has been quoted either literally or by content from the used sources. This work has not previously been submitted for a degree or diploma in any university.

_____

Würzburg, 20.04.2017