Deloitte.

Comparison of selected machine learning approaches- An analysis based on different economic application fields

Bachelor thesis

Start of studies 2017

Course A

Faculty of Economics

Course of studies Business Administration-Banking

Minor subject “Digital Finance”

COOPERATIVE STATE UNIVERSITY BADEN-WÜRTTEMBERG

VILLINGEN-SCHWENNINGEN

Author: Till Wagner

Supervising lecturer: Prof. Dr. André Kuck

Company: Deloitte, Munich

Table of contents

Page

List of abbreviations

AI	Artificial Intelligence
AUC	Area under curve
CART	Classification and regression trees
CORR	Correlation
DAX	Deutscher Aktien-Index
ELBS	Emergent law-based statistics
FICO	Fair Isaac Corporation
FPR	False positive rate
GBM	Gradient boosting models
GOSS	Gradient based one side sampling
IS	In-Sample
kNN	K-nearest neighbors approach
Lasso	Least absolute shrinkage selector operator
LightGBM	Light gradient boosting model
MAPE	Mean absolute prediction error
ML	Machine learning
MSPE	Mean squared prediction error
OLS	Ordinary least squares
OOS	Out-of-Sample
ROC	Receiver operating characteristic curve
SGD	Stochastic gradient descent
SVM	Support vector machine
TPR	True positive rate
XG-Boost	Extreme gradient boosting

Symbol directory

α	Mixing parameter for penalty terms
b	Intercept
Cov	Covariance
DiV	Degree of Inductive Verification
e	Euler`s number
$f(x)$	Function of x
i	Number of example
k	K-fold splitting parameter
(λ)	Lambda
N	Number of predictions
P	Probability
r	Correlation coefficient
s	Standard deviation
T	Set of Emergence
t	Time point
w	Slope
x	Independent variable
$\bar{x}$	Mean value for x
y	Dependent variable
$\bar{y}$	Mean value for y
$\hat{y}_i$	Predicted value
y_i	Actual value

Illustration directory

Page

1 Introduction

1.1 Aim and structure of the bachelor thesis

Using machine learning (ML) approaches to answer scientific questions has been an idea in computer science since the birth of information technology. After the two „Artificial Intelligence-Winters“ in the 1960’s and 1980’s, an increase in computer performance, cheaper memory technology and more available data, led to better results and the capability of analyzing more data than ever before.[1]

These developments led to a rise of data science techniques to investigate scientific questions. A widespread practice to answer these questions is the use of machine learning methods. The variety of answers to the best and most frequently used procedures in an annual survey on the data science platform Kaggle shows the complexity of finding the right approach to a specific problem.[2] The advantages of these new developments shall be exploited to answer the central question of this thesis, which is the following:

“Is there a machine learning approach that always leads to better results?”

To answer this research question, popular machine learning approaches are compared by measuring their performance with predefined comparison criteria in different economic application fields.

At first, a description of the considered application fields and datasets is given. To obtain a fair competition, three application fields with different data structures and a multitude of underlying datasets are used. The application fields are the following: cross-sectional data, times series data and time series panel data.

In chapter three, the metrics for the comparison are introduced. In this case the common performance metrics for machine learning tools: mean absolute prediction error, mean squared prediction error, area under curve and correlation are used. The fourth chapter presents concise descriptions of each approach, a short conclusion of all approaches as well as their different advantages and disadvantages.

[1] Cf. Döbel/Leis/Vogelsang, 2018, p. 21.
[2] Cf. Kaggle, 2019.

Afterwards, the application results of the various approaches on the mentioned application fields are analyzed. The results for each individual application field are presented in chapter five and a combined result overall application fields is demonstrated in chapter six.

This means the superiority of a method is not only checked overall datasets but also for each individual application field. Furthermore, it is analyzed whether it is possible to categorize the approaches using the following terms: performance, explainability and interpretability. In the end, the outcome and results of this thesis are concluded and critically appraised in chapter seven.

1.2 Definition and delimitation

The models and results are produced in the Python programming environment. They are mainly based on the Scikit machine learning library contained in Python. For the calculations of the ELBS-Tool, the center for emergent law-based statistics (ELBS) has kindly provided the access to its methods server.

The approaches are used with their default settings and standard architectures to obtain a fair competition. Therefore, no hyperparameter tuning is carried out. There are no common default hyperparameter for deep learning techniques like neural networks, they are usually set by experimenting with Grid-Search. Without hyperparameter tuning in this case, they are not able to achieve reliable results. Hence neural networks are not included, although they are commonly used tools in machine learning. The illustrations in this bachelor thesis are, unless otherwise explained, own depictions created with different visualization tools.

2 Application fields and selected datasets

To recognize the different strengths and weaknesses of the various approaches, three different applications fields are considered. The application fields vary in the data structure. They are categorized in three classes: cross-sectional datasets, time series datasets and time series panel datasets. Several, different economic datasets are contained in each application field. They were chosen since, almost all datasets are available for free access on the data-science platforms Kaggle and UCI. Furthermore, they consist of regression and classification problems, which allows to include both kinds of problems in the comparison.

All records are divided in two parts: training and test data frames. The quantitative relationship is 75 % for the training data, also called In-Sample (IS), to create and train the models. The test data, which is used for model evaluation, contains the remaining 25 %, also called Out-of-Sample (OOS). The division of the two parts is done by cutting the first 75 % from the dataset (chronological order), hence no random selection is carried out. The performance of the different approaches is only measured with the test data results.

In order to provide a better overview, each dataset will concisely be described, and the prediction goals are presented. Several visualizations of the datasets are used to show the most important relations and visualize the different prediction goals. They can be viewed in the appendix. In summary, there are 14 datasets used, which contain 88 different target variables and consist of 405.395 data entries.

2.1 Cross-sectional data

The category of cross-sectional datasets can be described by the following characteristics: They are the result of a data collection at one specific moment. Not the change over time, but the status at the time of the collection is important. To put it simply: One unit is observed at one specific time point.[3]

The category consists of the following datasets: IBM, Taiwan credit card, Polish companies' bankruptcies, bank marketing, health insurance claims, FICO-Heloc, Allstate insurance claims and Russian housing prices.

IBM-Dataset

The IBM dataset was artificially designed by the HR-Department of IBM to start a competition on the data-science platform Kaggle in 2017. It consists of various features of fictional employees working at IBM. Examples of these attributes are the education level, job role and job level as well as the monthly income or distance from work to residence. The given attributes are used to determine whether an employee is going to leave the company in the near future or not.

[3] Cf. Statista, 2020.

This prediction goal is called “attrition”.[4] As the figure in appendix 1 shows, the share of fictional employees at IBM who will leave the company in the next year is about 16 %.

Taiwan credit card

The Taiwan credit card dataset was published on January 26, 2016 on the machine learning platform UCI by I-Cheng Yeh. It consists of several features and attributes about credit card clients in Taiwan such as age, education, marital status or payment history. The goal is to classify whether a client will make a default payment in the next month or not. As can be seen in appendix 2 the default share of credit card clients is about 22 %.[5]

Polish companies’ bankruptcies

This dataset contains information about companies` bankruptcies in Poland in the time from 2000 – 2013. It contains a multitude of business performance figures. It was published in November 2016 on the UCI platform. The prediction goal is to classify whether a company goes bankrupt in the prediction timeframe or not. The dataset is divided in five parts. The first part contains financial key figures from the first year of the forecasting period and the goal is to predict the probability of a bankruptcy in the next five years. The second part contains data from the second year and the goal is to predict the classification in the next four years and so forth. The graphic in appendix 3 depicts that 217 companies out of 7027 companies went bankrupt within the first year.[6]

Bank marketing

The bank marketing dataset is made of data from a Portuguese banking institution. The bank initiated a phone call marketing initiative with the goal to sell a term deposit contract. It was published on the UCI platform by Moro et al. in December 2012. The database also contains phone calls to the same client which are sometimes necessary for the conclusion of a business deal.

[4] Cf. IBM, 2017.
[5] Cf. Yeh, 2016.
[6] Cf. Tomczak, 2016.

The goal is on the one hand to predict whether the client will subscribe a term deposit or not, and on the other hand the duration of the calls should be predicted. To carry out the predictions, twenty different attributes about the customers and phone calls such as age, job, education or number of calls to the customer are given.[7] The graphic in appendix 4 shows that nearly 44.000 deals were concluded and the average duration of successful calls, was 537 seconds.

Health insurance claims

The dataset contains attributes of health insurance clients in the United States of America. With several characteristics like body mass index, gender and number of children, the goal is to classify whether an insurance customer will file a claim. Furthermore, the charges and costs per claim should also be predicted. Charges are the individual costs billed by the health insurance company. In the graphic in appendix 5 it is shown that over 58 % of the clients filed a claim in the timeframe. The average charges per claim were 16.423,93 USD.[8]

Fair Isaac Company (FICO) Heloc

A Heloc is „a line of credit secured by your home that gives you a revolving credit line to use for large expenses or to consolidate higher-interest rate debt on other loans such as credit cards".[9] These kinds of loans are popular in the U.S. and played a major role in the Subprime crisis.[10] The dataset consists of anonymized information about houseowners with a Heloc. It was published in 2019 by the FICO Company in the US, which provides solutions for credit scoring. The target variable is the "Risk Performance". It shall be predicted whether a customer will repay his loan. "Bad" means they are minimum 90 days overdue with their payments at some point in the 24 months loan period, "Good" means the opposite. The graphic in appendix 6 shows that over 50 % of the clients were overdue with their payments at some point in the loan period.[11]

[7] Cf. Moro et al., 2012.
[8] Cf. Kaggle, 2018.
[9] Bank of America, 2020.
[10] Cf. Khandani/Lo/Merton, 2009.
[11] Cf. FICO, 2019.

Allstate insurance claims

The Allstate dataset was published on Kaggle in October 2016. The insurance company Allstate published data about the nature and severity of insurance claims in the past. The goal is to create or use an algorithm which is able to predict the claims severity in the future. As the figure in appendix 7 shows, for nearly 188.318 clients, the average severity of a claim is about 2.500 USD. The maximum amount is 106.863 USD, whereas the minimum is 6,00 USD.[12]

Russian housing prices

This dataset is provided by the Russian financial institution Sberbank and was published on Kaggle in 2016. The goal is to generate accurate price forecasts of houses in Russia that are sold in the time from July 2015 until May 2016. The training data from 2011 until 2015 also contains information about the macroeconomic situation in Russia at the time of sale as exogenous variables. The average price of the target variable is 157.000 USD.[13]

2.2 Time series data

The second category is filled with time series datasets, which are defined by the following characteristics: they emerge from an observation of the attributes of one object over a defined time-period at different specific time points. In simplified terms: One unit is observed at different contiguous time points.[14] This category consists of the following datasets: bike, car park, superstore and clicks.

Bike

The bike dataset evolved from a data collection of the bike sharing provider Capital bikeshare in Washington D.C. in the years 2011 and 2012. It contains information about the rental itself, the location of arrival, but also the weather conditions and seasonal information. It was published in December 2013 on the machine learning repository UCI by Hadi Fanaee and his Co-Authors.

[12] Cf. Allstate, 2016.
[13] Cf. Sberbank, 2016.
[14] Cf. Rottmann, 2020a.

The prediction goals in this case are as follows: Number of rentals per hour, temperature in Celsius, humidity, windspeed, number of casual users, and number of registered users.[15] As the visualizations in appendix 8 and 9 show, the number of registered users is twice as high as the number of casual users. It can also be observed that the number of rentals decreases with an increasing windspeed, humidity or temperature.

Car park

The car park dataset was created by collecting data from car parks in Birmingham in the time from October to December 2016. The data is provided by the Birmingham City Council. It consists of only four attributes: The car park identification number, its capacity and occupancy rates in the time from 8:00 until 4:30 pm as well as the date and time of the measurement. Thirty different car parks in Birmingham are included in the dataset. There are only two endogenous variables to predict: the number of cars in the car parks and the occupancy rate for the individual car parks at the given time-points.[16] As the graphic in appendix 10 reveals, the occupancy rate increases from the morning and reaches the peak after lunch time. Afterwards, the rate decreases and falls to the average occupancy rate of 49 %. The opening hours of the car parks are from 8:00 am until 4:30 pm. This means, cars can only enter until then but are still allowed to leave after 4:30 pm or the drivers can decide to leave the car overnight. This could explain the gap between the rate in the evening and morning and why it is not zero.

Superstore

The Superstore dataset is provided by the software company Tableau as a training sample for visualization and data analytics. It contains information about orders at the Canadian retail chain Superstore. The attributes given include information about the quantity of orders, sales, regional information as well as data about price discounts and profit rates. It is based on information of 2018. The prediction goals are as follows: quantity per order, discount, time to ship, profit rate. The discount is the price reduction of these goods, the time to ship is measured in days from order to shipping date and the profit rate represents the profit divided by the number of sales.

[15] Cf. Fanaee/Gama, 2013.
[16] Cf. Stolfi, 2016.

The table in appendix 11 demonstrates that the average order quantity was nearly four piece, for all segments and categories. Additionally, it can be seen that furniture was the category with the highest discounts but also lowest profit rates while the opposite is true for technology products. The time to ship is nearly four days for all categories and segments.[17]

Clicks

According to a confidentiality agreement, the dataset and source could not be made public. However, the most important key facts are presented as follows: The record contains 26,549 entries. It is provided by an online-shop and consists of information about the clicks that users made on the webpage. It has 6 target variables that should be predicted with the provided exogenous variables like time, product details or information about special commercial campaigns at the time. The prediction goals are labeled as follows: impressions, conversions per click, clicks per impression, profit, profit per click, and cost per click.

2.3 Time series panel data

The last category are time series panel datasets, which are defined by the following characteristics: They contain a dimension of cross-sectional data and a dimension of time series data from an observation of the attributes of different objects to different specific time points. To put it simply: Several units are observed at different contiguous time points.[18] The two datasets included in this category are the macro history dataset and a dataset about the performance of all companies that were ever listed in the Deutsche Aktien-Index (DAX).

[17] Cf. Tableau, 2018.
[18] Cf. Rottmann, 2020b.

Macro history

The data record contains data about annual returns of asset classes and various macroeconomic indicators in 17 different countries. There is a multitude of endogenous variables to predict: from macroeconomic variables such as the money supply or the interest rate level, to the total returns of individual asset classes. In addition, a large number of macroeconomic factors and key figures are included in the record. By using various economic parameters, unavailable data is recalculated and thus constructed.[19] In total, 52 different target variables had to be predicted. The dataset is provided by the cross functional research team Jordà, Schularick and Taylor and is hosted on the homepage of the "MACROFINANCE LAB" located in Bonn.

DAX panel data

This dataset includes information about the stock exchange prices of all companies that were ever listed in the DAX. This index consists of the thirty companies with the biggest free-floating market capitalization in Germany. As a result of changes in the stock market values, insolvencies and mergers and acquisitions, 114 companies were ever listed in the DAX. The specific dataset that is used is provided on the homepage of the international news organization Reuters and includes the period ranges from 1975 until March 2019.[20] Only seven example companies' stock prices had to be predicted to reduce the computing time. They are as follows: Allianz, BASF, Bayer, BMW, Commerzbank, Continental and Daimler-Benz.

2.4 Data record processing

It is to mention, that each approach has different preferences regarding the data preprocessing and feature engineering. To account for that fact, the underlying datasets are hardly processed. Thus, each approach has the same prerequisites. For the machine learning tools contained in the Scikit-Library, the following changes are made:

[19] Cf. Jordá/Schularick/Taylor, 2017.
[20] Cf. Reuters, 2020.

Exogenous/ explanatory variables

Exogenous variables are also called explanatory variables or independent variables. Based on them, the predictions for the endogenous variables shall be made.[21] Missing data in columns is replaced with the expanding mean in the respective column of the data record. In case the data could not be replaced, the whole row was removed. All column names contained in the dataset are transformed into strings for better representation. In addition, the categorical data in these columns is converted into numerical data using One-Hot-Encoding. This technique for variable transformation is commonly used because many ML-Algorithms do not work with categorical features.[22]

Endogenous variable/ prediction goals

Endogenous variables are the dependent components, often also called target variable or prediction goals. Their values are dependent on the exogenous variables and are predicted based on information given about the exogenous variables.[23] Columns with missing data found in the columns of the prediction goals, respectively the endogenous variables, are removed.

Special changes for extreme and light gradient boosting

The machine learning tools extreme gradient boosting (XG-Boost) and light gradient boosting (LightGBM), which are used during this thesis to analyze the described datasets, do not work with datasets containing special characters like colons, square brackets or comparison symbols. Therefore, these characters are replaced with a simple underscore.

ELBS-Tool

The changes mentioned above are not necessary for working with the ELBS-Tool provided by the center for emergent law-based statistics, at the DHBW Villingen-Schwenningen.

[21] Cf. Kenton, 2019.
[22] Cf. Burkov, 2019, p. 68.
[23] Cf. Kenton, 2019.

3 Applied comparison criteria

To define uniform criteria for the performance comparison of the different approaches, four different metrics were selected. These metrics are well-known performance measures in the field of machine learning. Their popularity and comprehensibility were the reasons for choosing these metrics.[24] Although they are widely spread metrics, a short description of these measures will be given for the completeness of this thesis.

- Mean absolute prediction error (MAPE)
- Mean squared prediction error (MSPE)
- Correlation (CORR)
- Area under curve (AUC)

3.1 Mean absolute prediction error (MAPE)

The MAPE is the average absolute difference between predicted values and actual values. It is often used with regression models.[25] One of its advantages is that it does not give too much importance to outliers. MAPE is a good method to determine how far the predictions differ from the actual values. In contrast to that, it does not give information, whether the values are over- or underestimated.[26] Mathematically it can be defined as follows:[27]

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

N is the number of predictions, $\hat{y}_i$ is the predicted value while y_i is the actual observed value.[28]

[24] Cf. Minaee, Shervin, 2019.
[25] Cf. Bhattacharjee, 2020, p. 57 – 59.
[26] Cf. Mishra, 2018.
[27] Cf. Bhattacharjee, 2020, p. 57 – 59.
[28] Cf. Bhattacharjee, 2020, p. 57 – 59.

3.2 Mean squared prediction error (MSPE)

The MSPE is the average of the squared differences between actual and predicted values. By using the MSPE, it is easier to compute the gradient than with the MAPE. Additionally, the effect of larger errors becomes bigger. This can be helpful if it is intended to focus on those but can be a problem with very high errors and like this it can distort the result.[29] The MSPE's are also called residuals and the mathematical notation can be seen below.[30]

$$MSPE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

3.3 Area under curve (AUC)

The area under the receiver operating characteristic curve (ROC) is a widely used metric in machine learning. The ROC is plotted with the false positive rate (FPR) against the true positive rate (TPR) for different cut-off points. The TPR, also called Recall or Sensitivity, can be found on the y-axis and the FPR, also called Fallout, on the x-axis. The ideal point is at the top left with an FPR of zero and a TPR of one. However, this scenario is not very realistic.[31] It is mainly used for binary classification problems. The AUC has a range of [0, 1]. The larger the value for the AUC, the better. A perfect model will achieve a score of 1.0, meaning that all examples are predicted correctly. A score for the AUC of 0.5 means the model is as good as random guessing. A score lower than 0.5 indicates a broken model.[32] The AUC gives information about the capability of a model to separate two classes correctly. To understand how the AUC works, four important concepts have to be defined:[33]

True Positives (Sensitivity)	YES is predicted and the actual output is YES
True Negatives (Specificity)	NO is predicted and the actual output is NO.
False Positives (Fallout)	YES is predicted and the actual output is NO.
False Negatives	NO is predicted and the actual output is YES.

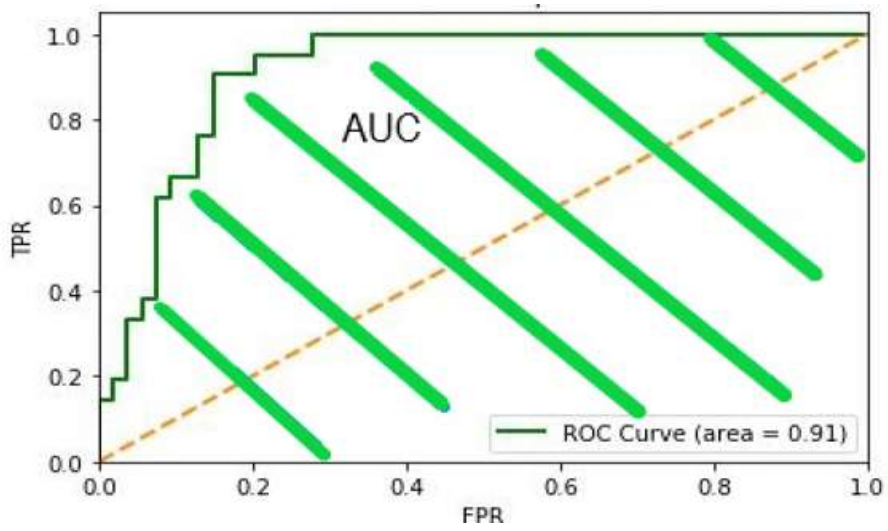
[29] Cf. Mishra, 2018.
[30] Cf. Bhattacharjee, 2020, p. 58.
[31] Cf. Vermeulen, 2020, p. 34 – 35.
[32] Cf. Burkov, 2019, p. 80 – 85.
[33] Cf. GoogleDevelopers, 2020b.

The TPR and the FPR are calculated as follows:[34]

$$TPR = \frac{TP}{TP + FN} \qquad FPR = \frac{FP}{TN + FP}$$

For the decision of the classification, a threshold can be defined. If the score exceeds the threshold the positive class are predicted and vice versa.[35] The AUC is easy to understand, can be visualized and includes more than one aspect of the classification.[36] An example for a ROC with the AUC is shown below. The ROC curve is colored green while the ROC curve with 0.5 is plotted as a yellow dotted line.

Figure 1 AUC example



Source: Own depiction with Scikit Learn in Python based on the Iris dataset.

3.4 Correlation (CORR)

Correlation in this case is the linear relation between the actual values of a dataset and the predicted values by the machine learning approach. In statistics, the correlation coefficient is used as a measure of the strength of a relationship between two numerical variables, the independent (x) and dependent variable (y). The correlation is indicated by the correlation coefficient. A perfect prediction made by the ML-Tool would achieve a correlation coefficient of 1.

[34] Cf. Narkhede, 2018.
[35] Cf. Burkov, 2019, p. 80 – 85.
[36] Cf. Burkov, 2019, p. 80 – 85.

This value is always between -1 and +1. A value close to 1 indicates a positive correlation, a value close to minus 1 a negative correlation and a value near 0 means hardly any correlation.[37] The correlation coefficient (r) is calculated as follows.[38]

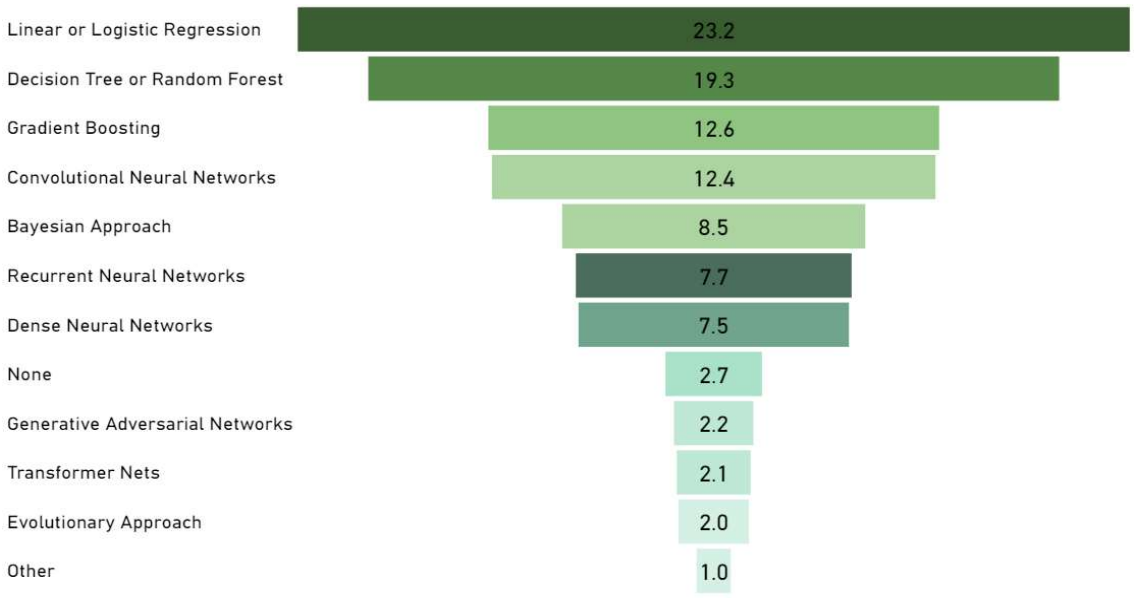
$$r = \frac{Cov(x,y)}{s(x)s(y)} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})/(N-1)}{s(x)s(y)}$$

The expression “Cov” is the covariance between the variables x and y. The term “s” describes the standard deviation of each variable, x_i is the actual value and $\bar{x}$ the mean of the sample for x. The variable y_i is the predicted value and $\bar{y}$ the mean of the sample for the predicted values. N is the size of the sample.[39]

4 Selected machine learning approaches

The selection of the used machine learning approaches is based on the most popular answers in a survey among members of the Kaggle-Community in 2019. The popularity distribution of used approaches in percentage can be seen in the graphic below.

Figure 2 Kaggle survey ML-Tool usage 2019 in %



Source: Own depiction based on Kaggle, 2019.

[37] Cf. Kurt, 2020, p. 77 – 78.
[38] Cf. Herzog/Francis/Clarke, 2019, p. 95 - 96.
[39] Cf. Herzog/Francis/Clarke, 2019, p. 95 - 96.

Although many of these approaches are well-known methods in the data science community, a short description of each approach will be given for the completeness of this bachelor thesis. The most popular approaches gained through the survey are selected for the comparison in this bachelor thesis: Linear and logistic regression approaches, decision trees and random forests as well as various decision tree-based ensemble methods like the gradient boosting approaches. Additionally, the commonly used methods of the support vector machine and the k-nearest neighbor approach are also applied.

To complement the comparison of well-known approaches, a new, powerful approach, which combines several advantages is also presented. The approach is based on emergent law-based statistics and is called ELBS-Tool. To analyze whether the implementation of all the mentioned approaches is really useful, a baseline estimation is also included. The baseline estimation is created with the expanding mean. Some of the approaches have subcategories that are also applied. These subcategories differ in some points from the main approach and are explained in further detail in chapter four.

4.1 Introduction to machine learning

In current opinion, machine learning is understood as a branch of f intelligence (AI).[40] There are many definitions that try to summarize this term. However, there is no consensus. One example is as follows: “The term machine learning encompasses methods that use learning processes to recognize connections in existing datasets in order to make predictions based on them”.[41] In other words: knowledge shall be generated artificially based on experience.[42] An important part is that the machine learns from the underlying data and does not need to be explicitly programmed for the task.[43] All approaches that are applied in this bachelor thesis are supervised learning algorithms. The only exception is the ELBS-Tool which is able to handle unlabeled data as well.

[40] Cf. Kreutzer/Sirrenberg, 2019, p. 4.
[41] Buxmann/Schmidt, 2019, p. 8.
[42] Cf. Kreutzer/Sirrenberg, 2019, p. 4.
[43] Cf. Buxmann/Schmidt, 2019, p. 8.

Supervised learning

By training with a part of a labelled dataset (training data) a model shall be developed which is able to make decisions (predictions) independently. A label is a feature of a data entry. An example task could be to build a model from labelled pictures of cats and dogs and to predict the animal species of new, unknown pictures (test data). There are other types of algorithms like unsupervised and reinforcement learning. Since no method in this thesis can be counted to these types they are not further explained.[44]

Machine learning basics

All different types of machine learning are carried out automatically. Nevertheless, it often takes a lot of human interacting to do the preprocessing, feature engineering, hyperparameter tuning or to interpret the results and understand the way of how the predictions emerge. This applies especially when it comes to complex models. The performance of a models' predictions is tested by using different criteria (metrics) on the testing data. These predictions can be based on regression or classification problems. Regression is the problem of predicting a continuous value for unlabeled examples. Classification is the problem of assigning a label to unlabeled examples, such as categorical values.

Categorical values are defined by a finite number of categories that must not follow a logical order. Continuous variables instead have an infinite number of values between any two values.[45] The dataset is usually divided into two parts: training data and testing data. This procedure is carried out to make sure the model is not only able to predict the renowned data from the training data, but also unknown values from the test data frames. If needed, an evaluation test data frame can be used as well. The training data is used to create the prediction model. The evaluation data is needed to choose the best learning algorithm and to evaluate the best values for the hyperparameter of the algorithm. With the testing data a final performance measurement is carried out.[46]

[44] Cf. Buxmann/Schmidt, 2019, p. 8- 10.
[45] Cf. Burkov, 2019, p. 36 – 37.
[46] Cf. Burkov, 2019, p. 74 – 75.

Bias and variance

Bias and variance are two common terms, which are essential for machine learning applications. The bias measures the prediction performance of a model for the training data. A low bias means the model is good at predicting the training data, a high bias reveals a weak performance in predicting the training data. A low variance indicates good predictions for testing data and vice versa.[47] If the model has very low bias it most likely has a high variance and is usually “over fitted”.

The model is too much adapted to the training data. It has practically learned the training data by memory and predicts them perfectly. However, it is very weak at predicting the unknown test data. Reasons for overfitting can be a too complex model for the data or too many features and too less observations. A possible solution is using an easier model to regularize the model.[48]

On the other hand, a model can also be underfitted, with a high bias but low variance. Reasons for that can be a too simple model for the data or the fact, that the underlying features are not informative enough to create a suitable. This problem can be solved through a more complex model or more suitable features.[49]

The bias and variance are considered to be in a trade-off relationship over model complexity. The intention is to build a model with low bias, low variance and a model complexity as low as possible. Predictions made by such a model are also called “sweet spot”. Highly different results for training and test data indicate over- or underfitting.[50]

Noisy Data

Noise in a dataset is supplementary information that is redundant. This expression also includes data corruption and data that cannot be interpreted or understood correctly by the model.[51]

[47] Cf. Hackernoon, 2019.
[48] Cf. Burkov, 2019, p. 75 – 78.
[49] Cf. Burkov, 2019, p. 75 – 78.
[50] Cf. Hackernoon, 2019.
[51] Cf. Burkov, 2019, p. 52.

Machine learning algorithms- Cost, loss and objective function

A machine learning algorithm consists of three components: an objective function, a cost function and a loss function. An objective function is the most general term for any function that is optimized during the training phase. This function is often called loss function. Commonly known examples are MAPE or MSPE. For linear regression, it is as follows: $Loss\ function = (f(x_i) - y_i)^2$. Hence, the goal is to minimize the squared error loss or, in simple terms, the squared distance between each observation and prediction for each individual sample.[52]

The second building block is an optimization criterion, often called cost function, which is based on the loss function.[53] The cost function is more general and by minimizing the cost function for the whole dataset the minimization for each single sample is also solved. Both are types of an objective function.[54] The loss function measures how wrong a model is in predicting each sample and the cost function measures the ability of predicting the whole dataset. The measure is usually the difference between actual value and prediction. The objective of an ML-tool is to find parameters, weights or a structure that minimizes the error.[55]

Finding the minimizing parameters is carried out by the third building block, an optimization routine, that uses training data to find a solution that meets the optimization criterion like the gradient descent or cross- validation for example.[56]

Regularization

The expression regularization illustrates methods that are used to prevent overfitting in machine learning tools in order to improve their ability to generalize predictions and reduce the loss.[57] The aim is to force the learning algorithm to build less complex models. This usually leads to a slightly increasing bias while the variance significantly drops. The complicated part is to set the penalized term right in order to reduce overfitting but avoid setting it too high and cause underfitting.

[52] Cf. Burkov, 2019, p. 57 – 74.
[53] Cf. Burkov, 2019, p. 57 – 74.
[54] Cf. McDonald, 2017.
[55] Cf. Rakhecha, 2019.
[56] Cf. Burkov, 2019, p. 57 – 74.
[57] Cf. Akerkar, 2019, p. 21.

The parameter used to control this is called Lambda (λ).[58] A regularized model is created by adding a penalty term to the objective function.[59] Two famous regularization approaches are L1 (Lasso)- and L2 (Ridge)-regularization.[60] Both are further explained in chapter 4.2.[61]

Cross validation testing

This method is commonly used to estimate the skills of models or parameters in machine learning problems. The method is often called k-fold cross-validation, since k is the only parameter and it decides how often a given dataset is split.[62] The dataset is randomly shuffled and then equally split into k groups. The first group is used as test dataset while the remaining groups are used for training. Afterwards, the second group is used as test data and the other groups are used for training. This is repeated until each partition is used for training and testing. The average results can be used as the estimate for Out-of-Sample error. K is usually chosen as a size of five, ten or statistically representative. However, there is no fixed rule.[63]

"Black Box" and "White Box" algorithms

The selected machine learning approaches can be divided into two classes: "Black Box" and "White Box" approaches.[64] The "Black Box" approaches are the following: Support vector machine, random forest, extra trees, bagged trees, adaptive boosting, stochastic gradient boosting, extreme gradient boosting, and Light gradient boosting. In conclusion these are mainly ensemble methods that use bagging and boosting.

The "White Box" approaches can be listed as follows: linear, ridge, lasso and elastic net regression, logistic regression, classification and regression trees, the k-nearest neighbors approach and the simple estimation with the expanding mean (baseline-estimator).

[58] Cf. Burkov, 2019, p. 78 – 79.
[59] Cf. Burkov, 2019. P. 78 – 79.
[60] Cf. Nagpal, 2017.
[61] Cf. Burkov, 2019, p. 78 – 79.
[62] Cf. Vermeulen, 2020, p. 39 – 40.
[63] Cf. Brownlee, 2018a.
[64] Cf. Burkov, 2019, S. 73.

Generally, the “Black Box” approaches achieve better performance results than the “White Box” approaches”. However, their results are considered less explainable and interpretable, and the way of how they emerge is vague, thus the expression “Black Box”.[65]

Interpretability is the ability to observe the relations between a cause and an effect within a system. In simple terms: It represents the ability to predict the outcome of changes in input or algorithmic parameters while the rest remains unchanged.[66] Explainability, on the other hand, refers to the ability to understand the relation between the feature values of an instance and the prediction or outcome of a model in the way of: “why is the prediction done in the way it is done?”.[67] It depicts how well “internal mechanics of a machine or deep learning system can be explained in human terms”.[68]

The better the accuracy of a machine learning algorithm becomes, the less interpretable and explainable is the model. This is also called a trade-off relation between the accuracy of the model and the ability to explain and interpret it.[69]

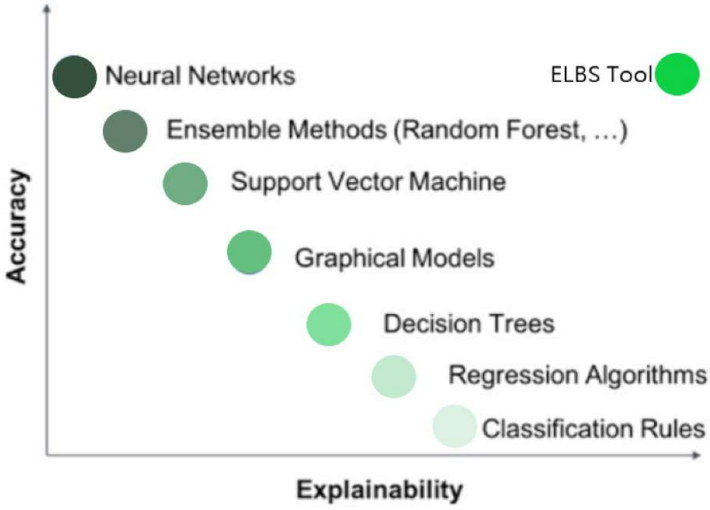
Especially a causal analysis is almost impossible in “Black Boxes”. Moreover, even if the model is good in predicting future values, it is still not clear what factors lead to these predictions and which are important influential factors.

Most of these approaches are prebuilt in machine learning libraries and enable good predictions. Nevertheless, most of the users are not able to comprehend their functionality. Since data privacy, ethics and equal rights require companies to explain their automated applicant pre-sorting or banks to explain their automatic loan decisions, this becomes a topic of high importance. Symptomatically, even if the same parameters are set for the same methods, the prediction results are usually not exactly the same.[70]

[65] Cf. Buxmann/Schmidt, 2019, p. 17.
[66] Cf. Choudhury, 2019.
[67] Cf. Gandhi, 2019.
[68] Gall, 2018.
[69] Cf. Kalayci, 2018.
[70] Cf. Ghoneim, 2020.

With the rise of more complex models like the “Black Box” approaches, the desire of the users for more comprehensible methods rises as the google trends graphic from January 2004 until May 2019 in appendix 12 reveals below. The search queries for these terms have more than quadrupled in the timeframe. The graphic below visualizes the tradeoff between the explainability and the accuracy of popular machine learning models. The ELBS-Tool instead combines the advantage of a comprehensible way of predicting with a high accuracy.

Figure 3 Tradeoff explainability and prediction accuracy



Source: Own depiction based on Gandhi, 2020.

According to the Kaggle survey about popular ML-Algorithms, a contradictory observation can be made: The most commonly used approaches are linear and logistic regression, followed by the other “White Box” approaches.[71]

The reasons why the “White Box” approaches are the most frequently used ML tools are as follows: they lead to faster results, do not occupy as much memory and have a “more understandable” way of leading to the decisions. This applies, even though prediction performance is sacrificed.[72]

The ELBS-Tool takes a special role in the comparison. While its prediction results are comparable or even better than the “Black Box” predictions, it combines significant advantages, which will be explained further in chapter 4.8.

[71] Cf. Carvalho/Pereira/Cardoso, 2019.
[72] Cf. Burkov, 2019, p. 73 – 75.

Regarding their nature, every mentioned approach has its strengths and weaknesses in specific application fields. The goal in the bachelor thesis is to find an approach that can be described as the best choice overall different datasets and for each application field. In general, all the approaches can be applied on all datasets. However, the performance differs depending on the application field. The only exception is the logistic regression. With endogenous variables of more than 15, this approach is overstrained and not applied on the dataset anymore.

The best way to determine the suitable algorithm for a specific problem is to apply a multitude of approaches on various datasets and compare the test data results, which is exactly what is done in this bachelor thesis.[73]

4.2 Regression models

Regression models use different regression techniques to solve machine learning problems.

Linear regression

In general, a linear regression is a mathematical description, which gives the ability to analyze relationships between the two numerical variables x and y. It can be used for multivariate problems and classification problems as well.[74] A linear regression algorithm creates a model, where a mapping function with input variables is used to predict a numerical output variable.[75] Linear regression in Scikit-Learn is also called “ordinary least squares” regression (OLS).[76] To create a linear regression model, a collection of labelled examples with target values as numerical features is needed. The goal is to describe the different data points by the most suitable line.[77] A model as linear combination of the characteristics of x is defined by:[78] $f(x) = w \cdot x + b$

The variable x is defined as the independent, explanatory variable, also known as value of a feature f(x), is the dependent variable, or outcome. w is the slope of the line and b represents the intercept.[79]

[73] Cf. Burkov, 201, p. 73 – 75.
[74] Cf. Vermeulen, 2020, p. 76.
[75] Cf. Paper, 2020, p.105.
[76] Cf. Akerkar, 2019, p. 22 -23.
[77] Cf. Burkov, 2019, p. 39 – 43.
[78] Cf. Burkov, 2019, p. 39 – 43.
[79] Cf. Burkov, 2019, p. 39 – 43.

The distance of each datapoint to the function is the error of the regression line and is measured with the MSPE. By minimizing this error, through finding the best parameters for w and b, the best describing line can be found. They can be found in following ways: Experiment, gradient descent, or for simple linear problems, differential equation. The squaring is performed to level out the fact whether a deviation is in positive or negative direction.

When the search for the correct parameters is successful, one can insert the input values in the equation and get a prediction.[80] To minimize the squared prediction error, a cost function that shall be minimized is defined:[81]

$$Cost\ function = \sum_{i=1....N} (f(x_i) - y_i)^2$$

The expression $(f(x_i)$ is the predicted value while y_i is the actual value, N is the size of the collection and i the number of the example.[82]

Ridge regression

Ridge regression is an algorithm used for solving regression problems in machine learning. It can be used for multivariate problems and classifications as well.[83] Using ridge regression shall solve the problem of overfitting and is an extension of the linear regression.[84] The goal is to keep the number of features constant, but to reduce the magnitude of coefficients. This is often used, if only a few features highly impact the predictions.[85]

Linear regression tends to overfit for two reasons: First of all, the procedure of least ordinary squares finds the best parameters for the given training dataset, not the whole dataset. Secondly, the approach does not weigh the parameters, which means it takes the unbiased coefficients. If the optimal parameters for w and b are found, the low bias of the model is most likely accompanied by a high variance in the test data.[86]

[80] Cf. Paper, 2020, p.105.
[81] Cf. Burkov, 2019, p. 39 – 43.
[82] Cf. Singh/Manure, 2020, p. 28 - 29.
[83] Cf. Maklin, 2018.
[84] Cf. Singhal, 2018.
[85] Cf. Jain, 2017.
[86] Cf. Kim, 2019.

To find the lowest total error, linear regression is extended by the ridge regression penalty term to slightly increase the bias while getting a significant drop in variance and finally move the model to the sweet spot.[87] The cost function is as follows:[88]

$$Cost\ function = \sum_{i=1....N} (f(x_i) - y_i)^2 + \lambda * \sum_{i=1....N} w_i^2$$

(Sum of squared error term) + Penalty term ($\lambda *$ slope^2)

The penalty is calculated through multiplying lambda with the squared weight of each individual feature.[89] The new term is called “penalty” because it increases the residual sum of squares. The optimum lambda can be found and tuned by cross-validation to find the model’s best fit. It should be chosen where the mean squared error is the lowest.[90]

The sum of the square error shall be minimized while satisfying the constraint of the penalty term. If Lambda is set > 0, the constraint is added to the coefficients. If Lambda is set to zero, the values are the same as for the linear regression.[91] The squaring of the parameters results in an increased punishment of very influential parameters. For the minimization of the cost function the parameters are forced to take smaller values and be more useful for predicting unknown data. The idea is that the parameters (coefficients) have to be set in the way that low influential features are more penalized and vice versa.[92] The slope is reduced by the addition of the penalty term. Thereby, the model becomes less sensitive to changes and variations in the independent variable. A bigger λ means a decrease in the slope and the regression line becomes more horizontal.[93] The penalty term shrinks the coefficients towards but never equal to zero, the larger lambda gets. The benefit is that the variance and error value is lowered, but it does not reduce the number of features. It reduces the model complexity and shrinks the effects of the coefficients.[94]

[87] Cf. Maklin, 2018.
[88] Cf. Kim, 2019.
[89] Cf. Maklin, 2018.
[90] Cf. Kim, 2019.
[91] Cf. Kim, 2019.
[92] Cf. Bhattacharya, 2018.
[93] Cf. Cf. Kim, 2019.
[94] Cf. Hackernoon, 2019.

Least absolute shrinkage selector operator (Lasso) regression

Lasso regression is short for “least absolute shrinkage selector operator” and it resembles the ridge regression. It is especially helpful in overfitted models with a high number of features because it automatically carries out feature selection.[95]

This means lasso regression not only reduces overfitting, but also reduces the number of features.[96] The lasso regression approach selects some features and leaves their values almost unchanged but reduces the other coefficients to absolute zero, even for a small lambda.[97]

The lasso regression also contains a penalty term. The difference is, instead of adding the squared values of the coefficients, it adds the absolute values of the coefficients (slope). The squaring is the reason why in ridge regression the coefficient’s value can never reach zero, while this is possible with absolute values in lasso regression.[98] Because lasso regression is able to exclude useless features from the equation, if the slope is reduced to 0, it can achieve better results at reducing variance with models that consist of several useless features.[99] The cost function for lasso regression is as follows: [100,101]

$$Cost\ function = \sum_{i=1....N} (f(x_i) - y_i)^2 + \lambda * \sum_{i=1....N} |w_i|$$

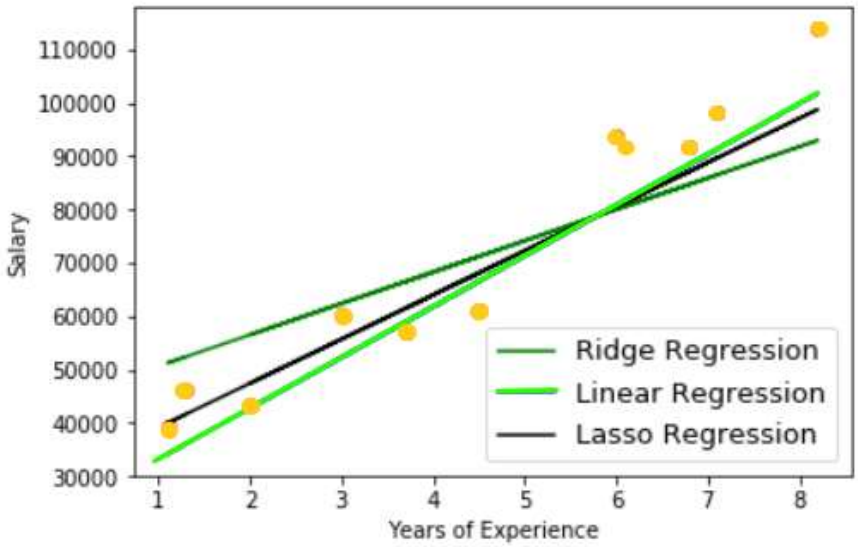
(Sum of squared error term) + Penalty term ($\lambda *$ slope)

If lambda is set 0, the penalty term is equal to the OLS equation. The larger lambda is chosen, the more features are reduced to zero. Some features are eliminated completely, and a subset of important predictors remains. This can help reducing model complexity. The remaining predictors are considered to be important. This procedure can be described as feature selection.[102]

[95] Cf. Jain, 2017.
[96] Cf. Bhattacharya, 2018.
[97] Cf. Jain, 2017.
[98] Cf. Jain, 2017.
[99] Cf. Maklin, 2018.
[100] Cf. Bhattacharya, 2018.
[101] Cf. Jain, 2016.
[102] Cf. Hackernoon, 2019.

The following graphic depicts an example of the effects of ridge and lasso regression on the slope of the regression line in the context of predicting an employee's salary with the years of work experience. It shows how the added penalty term reduces the slope and lowers the effects of overfitting.

Figure 4 Example- Effects of ridge and lasso regression



Source: Own depiction based on Acharya, 2019.

Elastic net regression

Elastic net regression is a combination of lasso and ridge regression. It is calculated by extending the linear regression cost function with the penalty terms of the ridge and lasso regression. At first, the parameters are grouped and then shrunk, associated with correlated variables. Based on this process, the parameters in each group either remain or are removed all at once.

In the process, the weight of the ridge regression penalty term and lasso regression penalty can be set differently.[103] The equation is as follows:[104,105]

$$Cost\ function = \frac{\sum_{i=1....N}(f(x_i) - y_i)^2}{2n} + \lambda * (\frac{1-\alpha}{2} \sum_{i=1....N} w_i^2 + \alpha * \sum_{i=1....N} |w_i|)$$

[103] Cf. Maklin, 2017.
[104] Cf. Bhattacharya, 2018.
[105] Cf. Jain, 2016.

The expression α is the mixing parameter between the ridge and lasso regression. If it is set to 0, only ridge regression comes into play and if it is set to 1 only lasso regression will play a role. Accordingly, an alpha value between 0 and 1 has to be set to optimize the elastic net. Alpha must be tuned by cross-validation.[106]

Elastic net regression is especially helpful in cases with correlations between parameters. If a dataset contains a bunch of correlated, independent variables, the algorithm will form groups of those correlated variables. If a group contains a strong predictor (strong relation to dependent variable), the whole group is included in the model. The reason for this is that the other variables in the group are needed for interpretability of the high influential variable and therefore cannot be removed. If there are no strong predictors, the whole group is removed. This approach combines the advantages of ridge and lasso regression. At first, useless features are removed through lasso regression and ridge regression will adjust the weights of the important features.[107]

Logistic regression

The approach is mainly used for binary classification problems when the target variable is categorical, for example classifying whether an e-mail is spam or not.[108] However, it can also be used for multinomial, ordinal or regression problems as well, but in a limited way.[109]

The linear and logistic regression approaches are the most popular machine learning approaches according to the Kaggle survey.[110] The logistic regression is, in contrast to the name, originally a classification algorithm. It also provides the probability of the allocation to one class. The probability can have values between 0 and 1, while the linear regression predictions can have an infinite or negative infinite value.[111] Based on the predicted probability the classification is carried out.[112]

[106] Cf. Oleszak, 2019.
[107] Cf. Jain, 2017.
[108] Cf. Swaminathan, 2018.
[109] Cf. Singh/Manure, 2020, p. 37- 47.
[110] Cf. Kaggle, 2019.
[111] Cf. Burkov, 2019, p. 44- 46.
[112] Cf. Schüler, 2019, p. 105 – 110.

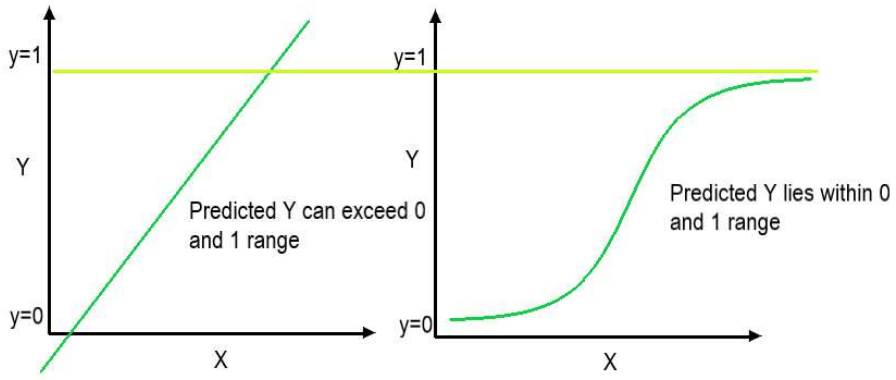
The name confusion comes from the fact that the predictions are based on the same method as the linear regression, instead no numerical values shall be predicted but the outcome as categorical values.[113] But instead of optimizing a linear function, a logistic function, the sigmoid function, is used.[114,115]

$$Cost\ Function\ P(x) = \frac{1}{1 + e^{-(w_n x_n * b)}}$$

The sigmoid function is mainly used to explain growth of populations with a high growth rate and limitation at the maximum capacity of the given environment. If plotted, the curve is s-shaped and takes any real-valued number between 0 and 1. Logistic regression models the probability of the default, e.g. the first class.

For any input in the function, the output is a probability for the default class between 0 and 1. To convert the predicted probability into the classification a decision threshold e.g. >= 0.5 can be set. Since this method uses a regression technique, the categorical target values must be converted into numerical values, using One-Hot Encoding.[116] The optimization of the function can be carried out by using the gradient descent.[117] The probability shall be maximized by minimizing the logistic loss function.[118] The difference between linear and logistic regression can be seen below.

Figure 5 Example- Linear regression vs. logistic regression



Source: Own depiction based on Prabhakaran, 2020.

[113] Cf. Singh/Manure, 2020, p. 37- 47.
[114] Cf. Singh/Manure, 2020, p. 37- 47.
[115] Cf. Burkov, 2020, p. 44 – 46-
[116] Cf. Swaminathan, 2018.
[117] Cf. Brownlee, 2019.
[118] Cf. Swaminathan, 2018.

4.3 K-nearest neighbors (kNN)

The k-nearest-neighbors algorithm was initially introduced as a classification algorithm. The idea is, that objects with similar features tend to have the same values. For data with discrete labels, a classification algorithm is carried out, and for datasets with continuous labels, a regression algorithm is used.[119]

The algorithm retains all training examples and searches for the k- training examples (nearest Neighbors) closest to a new sample x and returns the most frequently occurring label (in case of classification) or the average value (in case of regression). The closeness of the examples is determined by distance functions, often Euclidean distance.[120] In general, it can be any metric measure, like physical distance in km, or the number of features that are different to the first sample. This distance is called "degree of diversity".[121]

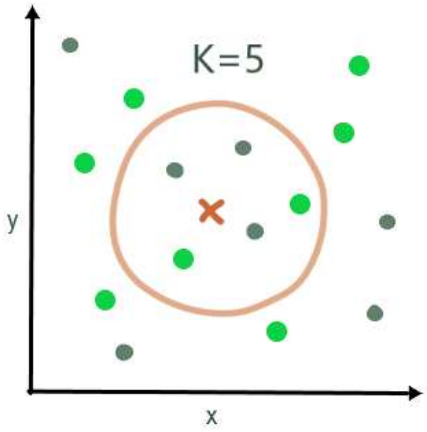
$$Euclidean\ Distance = \sqrt{(x_2 - x_{1)}{}^2 + (y_2 - y_{1)}{}^2}$$

The procedure of predicting is also called "vote of majority": If k is set to five for example, the new object gets the same class as the majority of the nearest five neighbors.

In a large dataset the k needs to be set high, thus outliers are not weighted too heavily.[122] The number of neighbors used for classification must be determined in advance, in the form of the selected k. The optimal k can be found by trial and error. If k is set as one, each new value is predicted similar to the closest data point.[123]

[119] Cf. Bhattacharjee, 2020, p. 79 -80.
[120] Cf. Burkov, 2019, p. 55 – 56.
[121] Cf. Scikit-Learn, 2020a.
[122] Cf. Neumann, 2019, p. 73-80.
[123] Cf. Akerkar, 2019, p. 23- 24.

Figure 6 Example- Binary classification with kNN



Source: Own depiction based on Neumann, 2019, p. 73 - 80.

4.4 Support vector machine (SVM)

The support vector machine (SVM) was intentionally built to solve binary classification problems. Deviating from this, the method can also be used for multiclass classifications and regression problems.[124] The goal of this approach is to find a separating hyperplane between classes by using support vectors. For two-dimensional data the hyperplane is a dividing line, defined by the support vectors.[125]

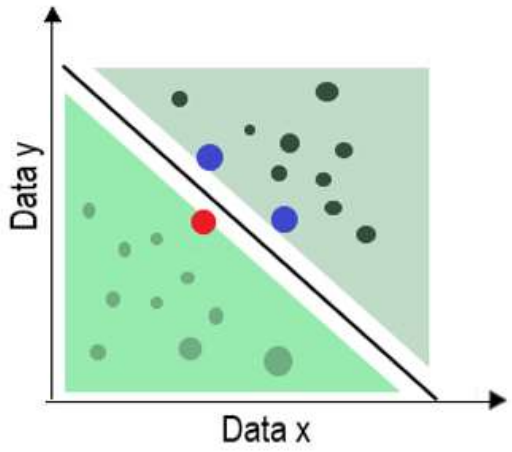
Within the SVM, every feature is a vector as a point in a high-dimensional space. All feature vectors are plotted on an imaginary n-dimensional plot. An n- dimensional line (hyperplane) separates the classes. The separating line is also called decision boundary and can be a complex function, straight or curved.[126]

The graphic below displays an example for a binary classification problem. The data points are brought into a diagram where they represent point-clouds. The separation line shall divide the classes and is located in the middle between them.[127] To solve a linear, binary classification problem, three “support vectors” are needed.

[124] Cf. Döbel/Leis/Vogelsang, 2018, p. 15 - 33.
[125] Cf. Bhattacharjee, 2020, p. 77 - 78.
[126] Cf. Burkov, 2019, p. 50 – 54.
[127] Cf. Bhattacharjee, 2020, p. 77 - 78.

Two support vectors out of one point-cloud will define the first class (blue points) and the direction of the separation line. The third point is needed to determine the second class (red point). The algorithm creates a linear equation for which the distance between the two point-clouds is maximized. In case the problem cannot be solved with a linear equation, the Kernel-trick can be used. In this case the data input is transferred into a new, high dimensional coordination system, called feature room where the separation with the hyperplane is possible and carried out.[128] Afterwards, the data is transferred back into the original coordination system and creates the separating function.[129]

Figure 7 Visualization of an SVM classification



Source: Own depiction based on Aberham/Kuruc, 2019, p. 95 – 105.

4.5 Decision trees

Decision Trees are an intuitive way of generating predictions. Nevertheless, trees can become very complex, especially when ensemble methods like bagging and boosting are used. These will be explained in more detail in chapter 4.6.

[128] Cf. Aberham/Kuruc, 2019, S. 95 – 103.
[129] Cf. Predictive DataMining Models S. 102 -105.

Decision trees- Classification and regression trees (CART)

The abbreviation CART which is used in Scikit-Learn, stands for the combination of "classification and regression trees". The main idea behind this combination is to build an algorithm that could universally be used for classification (categorical data) and regression problems (continuous data) as well.[130]

The models built by the decision tree method are founded on the "If-Then" format.[131] The main structure of a decision tree can be concluded as following: A tree is built starting from the root node. The dataset shall be split into different "branches" by the features that split the training data best with respect to the target variable.[132] The decision tree algorithm works from the root node up, all the way to the leaves. The branches contain decision nodes which split the data by the next-best splitting attribute.

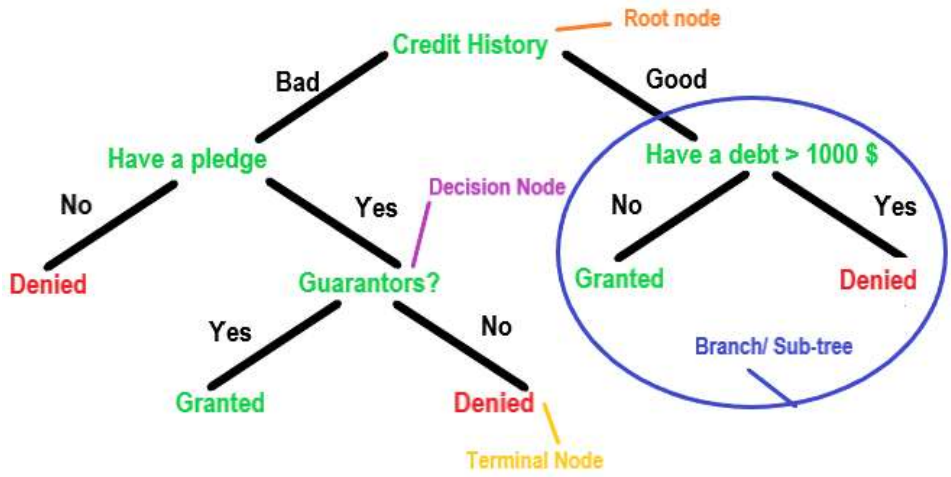
If the value of the feature is less than a certain threshold, the left way is followed, otherwise the right way is followed. When a leaf node is reached, the system decides to which class an example belongs.[133] The first split (branching off) is the one with the potential to split the data best, the following branching off uses the second-best attribute and so forth.[134] At each node, the system searches for a distribution that minimizes a predefined measure or aborts the procedure in the following cases: When all examples are correctly classified, splitting reduces the entropy by less than a specified value or the tree reaches its maximum predetermined depth.[135]

For new values, the sample is tested against the tree structure. A new object follows the way from the root node until the leaf node with the correct category or value for this sample.[136] For classification cases, the best splitting feature is usually measured with an entropy heuristic or Gini-Impurity.[137] Entropy is used as a measure of uncertainty about random variables.[138]

[130] Cf. Scikit-Learn, 2020d.
[131] Cf. Olson/Wu, 2020, p. 107 – 122.
[132] Cf. Bhattacharjee, 2020, p. 68 – 69.
[133] Cf. Burkov, 2019, p. 44 -46.
[134] Cf. Akerkar, 2019, p. 24 -27.
[135] Cf. Burkov, 2019, p. 44 -46.
[136] Cf. Akerkar, 2019, p. 24 -27.
[137] Cf. Olson/Wu, 2020, p. 107 – 122.
[138] Cf. Burkov, 2019, p. 46 – 50.

The Gini Impurity measures how often a randomly chosen element from the set would be incorrectly labeled, in case it is randomly labeled, with respect to the distribution of labels in the subset.[139] For regression problems, the criteria are the MAPE or MSPE, described in chapter 3.[140] The graphic below reveals an example of how banks could make loan decisions with decision trees.

Figure 8 Example- Loan decision with decision tree



Source: Own depiction based on Prajapati, 2019.

4.6 Ensemble methods

Ensemble methods can be described as a collection of predictors which use several models of one or more learning approach to achieve better results. The outputs are combined through different averaging methods and from them, the final prediction is derived. Famous methods are the bagging or boosting structure.[141]

Bagging

Bagging is a method that uses parallel training of independent models on random subsets of a dataset. It creates a class of algorithms which build several instances of “Black Box” estimators on random subsets (with replacement) of the original training set and aggregates their individual predictions to get a final prediction.[142]

[139] Cf. Zhou, 2019.
[140] Cf. Scikit-Learn, 2020d.
[141] Cf. Singh/Manure, 2020, p. 47 -52.
[142] Cf. Singh/Manure, 2020, p. 47 -52.

The abbreviation bagging stands for bootstrap aggregation. The averaging of the different models is carried out by different weighting, mean calculation or majority voting. Well-known bagging methods are the bagged tree, random forest method or extra trees. Since Bagging is a good way to reduce overfitting, it is best used with complex models. Through adding randomization in the structure, the variance as a result of overfitting shall be reduced.[143]

Boosting

Boosting is a technique of sequential model training on random samples (with replacement) from the dataset, but different weights of the samples. The weight of incorrect predictions is increased to give special attention to more difficult cases. Increasing the weights means that they are sampled more often to "boost" the chance of a correct prediction. Similar to the Bagging approach it uses a multitude of models, usually decision trees, to improve the prediction performance. The boosting method builds the different models in a feed forward, sequential manner. The output of the first model is fed to the next model and applied on the next sequential subset.[144]

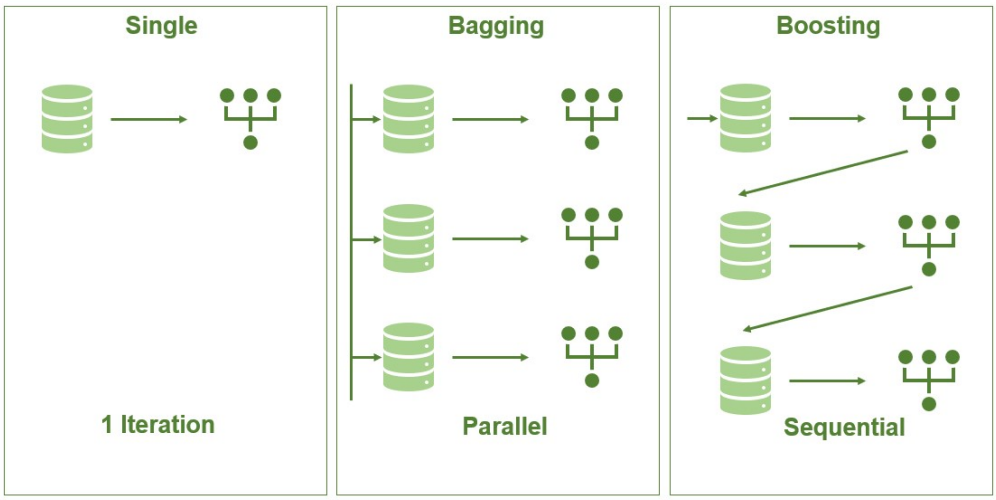
The main idea of the sequential training is that the approaches shall learn from the mistakes of the previous models. Weak learners shall be combined into a strong classifier. A weak learner is an ML algorithm that "provides accuracy marginally better than random guessing".[145] A well-known boosting method without gradient boosting is called AdaBoost. Famous ensemble methods which use gradient boosting are extreme gradient boosting, stochastic gradient boosting or light gradient boosting.[146]

In the bagging methods, trees are usually fully grown. In contrast, boosting uses simpler models. For decision trees as base estimator, the trees only have a few or one split, called decision stumps. The optimum parameters are usually found through cross-validation or, in case of gradient boosting models, by gradient descent techniques.[147]

[143] Cf. Singh/Manure, 2020, p. 47 -52.
[144] Cf. Singh/Manure, 2020, p. 47 -52.
[145] Vermeulen p. 138.
[146] Cf. Singh/Manure, 2020, p. 47 -52.
[147] Cf. Bhattacharjee, 2020, p. 72.

The visualization below reveals the different structures of a decision stump with only one iteration, a bagging model which builds the trees in parallel and a boosting model where they are built in a sequential manner.[148]

Figure 9 Single decision stump, Bagging and Boosting examples



Source: Own depiction based on Grover, 2017.

In the following, the methods that base on the bagging technique are introduced firstly.

Bagged trees

The bagged tree, random forest and extra trees algorithms are all Scikit-Learn tools which are based on randomized decision trees and can be seen as a development of one technique. The procedure is as follows: A multitude of classifiers or regressor models, based on the problem, is built and randomness is added to the structure. The final prediction is derived by the average or majority of the individual models.[149]

In bagged trees, same weighted random subsets are taken for model building. The final output of a multitude of parallel built, unweighted tree model prediction is averaged across the predictions of all sub-models. In simple terms it is just a consortium of simple decision trees. In this approach, all trees are fully grown.

[148] Cf. Bhattacharjee, 2020, p. 72.
[149] Cf. Scikit-Learn, 2020e.

Furthermore, at every node the algorithm searches in all features in the dataset the one that splits the data best at the specific node.[150] Hence, Bagged trees consider all features in the dataset for the decision, while the next method, random forest can only decide based on the features in the random subset.[151]

Random forest

The random forest algorithm is a further development of the decision tree approach and an extension of the bagged tree method. It belongs to the group of ensemble learning methods. It can be used for either classification- or regression problems.[152]

In the training phase, a multitude of decision trees (forest) are built in parallel. Every tree makes a prediction and the overall decision are built on the average (regression) or majority (classification) of all the trees in the forest. Each tree is trained with a different, random subset of the data.[153] Due to the training with the subsets, each of the single trees in the forest is weaker than a full decision tree model, but the combination of the individual trees in the forest leads to a better performance. The randomness from the bagged tree approach is extended because not only the subsamples but also the features for splitting are chosen randomly.[154] The random sampling in the random forest approach shall reduce variance and the correlation between the individual classifiers or regressors.[155]

Extra trees

The extra trees algorithm is also called extremely randomized trees approach.[156] The model building process in the training phase is based on random subsets. The decisions at each node are based on the features that best split the data, chosen only out of the specific random subset. Thresholds are randomly drawn for each candidate feature and out of these, the best threshold is picked as the splitting rule. Since the splits (thresholds) are chosen randomly for each feature, the memory occupancy is reduced in comparison to the random forest approach.[157]

[150] Cf. Ravindran, 2020.
[151] Cf. Ravindran, 2020.
[152] Cf. Paper, 2020, p. 5 -7.
[153] Cf. Akerkar, 2019, p. 26.
[154] Cf. Akerkar, 2019, p. 26.
[155] Cf. Ravindran, 2020.
[156] Cf. Scikit-Learn, 2020e.
[157] Cf. Scikit-Learn, 2020e.

The goal of the evolving introduction of randomness in the algorithms is to decrease the variance, by putting up with a slight increase in bias. Decision trees have a high variance, random forests have a medium variance, and extra trees have a low variance. The decreasing variance is accompanied by a slightly increasing bias.[158]

The different structures of the three approaches can be concluded in the following. For bagged trees, the model is built from random subsets, and decisions at each node are based on the features that best split the data, chosen out of the whole dataset. For each of these selected features, the approach searches for the best cutting point in the whole dataset to determine the split for the given feature, also called threshold.[159]

In random forest structures, the model is built from random subsets, decisions at each node are based on the features and thresholds that split the data best, chosen only out of the specific random subset.[160] The methods that are described in the following are founded on the decision tree-based boosting technique.

Adaptive boosting technique (AdaBoost)

A famous example of the supervised boosting technique is the adaptive boosting approach, called AdaBoost. The main idea of this method is to improve the predictions through weighted, sequential model building.[161] The algorithm can be used for regression as well as for classification problems. It can be summarized as the weighted combination of m- weak classifiers.[162]

For AdaBoost, as for any other boosting algorithm, the goal is to combine several weak learners to one strong learner. The procedure of AdaBoost predicting can be described as follows: A random training sample is selected. At first, all the samples are equally weighted. The weight is an indicator for the algorithm to recognize how important it is to classify this observation correctly. The algorithm continues to iteratively train the sample and tries to erase the error from the previous model. This is done by assigning a higher weight to misclassifications.

[158] Cf. Scikit-Learn, 2020e.
[159] Cf. Scikit-Learn, 2020e.
[160] Cf. Scikit-Learn, 2020e.
[161] Cf. Vermeulen, 2020, p.138 – 140.
[162] Cf. Freund/Schapire, 1997, p. 119 – 139.

Since the next tree is fitted on the new weighted dataset, the cost of misclassifying those observations are higher, and the algorithm is careful not to misclassify them again, because this would imply a higher cost.[163]

In the Scikit-Learn library, the AdaBoost algorithm uses decision trees with a depth of one as weak learners (decision stumps), which means they split just once in each model.[164] For a binary problem, the examples are split into two subsets based on the value of one feature. A threshold is defined which indicates the classification in one of the two groups. Each of the decision stumps uses a different feature. The best fitting decision stump is found by trying every feature and every possible threshold. It seems that there is an infinite number of possible thresholds. But a new threshold is only tried if it significantly changes the distribution of classifications.[165] This process is continued until no errors are left or the predefined maximum number of estimators is reached.[166]

4.7 Gradient boosting models (GBM)

The gradient boosting models are ensemble methods. They are all based on the gradient descent technique and usually provide much better predictions than the previously explained ensemble techniques. This is the reason why the approaches are divided into ensemble methods without and ensemble methods that use gradient descent technique with boosting.

Introduction gradient descent technique

The gradient is a vector whose entries are the first partial derivatives of the loss function, or in simple terms: the slope of it. It describes the steepness of the error function of a loss function. A higher gradient means a higher slope. The gradient is used to find the direction in which the parameters must be optimized to minimize the error. This is called “descending the gradient”.[167]

[163] Cf. Avlani, 2018.
[164] Cf. Scikit-Learn, 2020c.
[165] Cf. Desarda, 2019.
[166] Cf. Navlani, 2018.
[167] Cf. Burkov, 2019, p. 57 – 60.

The method tries to find the cost-function minimizing parameters through iterations. First of all, the gradient is calculated.

Secondly, a starting point (random values for the parameters) is defined from where the gradient descent is carried out. From the starting point, the algorithm travels down the slope in small steps until the minimum of the function is reached.

It “travels down” by nudging the parameters in the opposite (negative) direction of the gradient, because the goal is to minimize the gradient. During each iteration the updated parameters are used, and the algorithm stops when the gradient is almost zero. This technique is especially helpful in cases in which the optimal parameters cannot be found through equating the function to 0.[168]

The steps in which the iterations are carried out are called “learning rate”. A large learning rate leads the algorithm to take big steps down the slope, which makes the computation much faster, but it is also likely to miss the minimal point. It carries the risk of reaching a local minimum instead of the global minimum. Hence, it is a good idea to keep the learning rate low, e.g. 0,01. The best way to avoid local minima is the stochastic gradient descent.[169]

The gradient descent technique is carried out for “batches”. A batch is the number of samples that is taken from a dataset to calculate each iterations’ gradient. Usually a batch is the whole dataset. Nevertheless, especially for large datasets, it can get computationally extremely expensive, since all samples have to be used for gradient descent in each iteration. Therefore, the stochastic gradient descent technique is introduced.[170]

[168] Cf. V Srinivasan, 2019.
[169] Cf. GeeksforGeeks, 2020.
[170] Cf. V Srinivasan, 2019.

Stochastic gradient descent (SGD)

With the stochastic gradient descent technique, the way of finding the best parameters shall be made more reliable and efficient. The stochastic gradient descent introduces randomness into the algorithm to make the computation much faster. The SGD uses only a batch size of one, which is one randomly chosen data point from the dataset for each iteration. This means that only the gradient of the cost function for one sample at each iteration is found, instead for all examples.

This makes the way of leading to the final result noisier, but the computations much faster. A solution to the noisy data could be mini-batch SGD which takes more examples per iteration but is still faster than a full batch.[171]

Gradient boosting

The gradient boosting technique is a famous technique in the data science community with some subtypes explained in the next section. It currently wins the most Kaggle competitions. It is also known under the name gradient tree boosting or gradient boosting machines.[172] The gradient boosting approach can be used for regression as well as for classification problems.[173]

It represents a numerical optimization problem where the goal is to minimize the loss through sequentially adding new weak learners which correlate with the negative gradient of the loss function of the whole model.[174] Through the negativity, the gradient is minimized, and a minimized gradient is equal to the minimum of the loss function.[175]

Instead of training a very complex single model, as in stochastic gradient descent, the gradient boosting trains an ensemble of simple models. At first, a simple model like a decision tree is trained. The following model shall focus on the gradient of the error of the previous model, built in the way that it moves the gradient of the previous model into a negative direction towards 0.

[171] Cf. GoogleDevelopers, 2020a.
[172] Cf. Brownlee, 2020.
[173] Cf. Vermeulen, 2020, p. 143.
[174] Cf. Brownlee, 2016.
[175] Cf. ODSC, 2018.

By doing so it corrects the previous models' mistakes and finds the minimum of the loss function by moving the gradient towards zero. This is the case because the gradient describes the steepness of the loss function´s error function and a gradient near zero indicates the minimum of the loss function. Models are added stage-wise.

The specification of the old model does not matter to the new model, because it tries to minimize the gradient of the previous model and therefore can take any form of a function that fulfils this purpose.[176] In summary, the error of the previous weak learner takes the place of the target variable and the tree is built using the error function as the target variable. The goal with this is to minimize the prior error and by doing so the prediction accuracy is improved.[177]

Stochastic gradient boosting

The stochastic gradient boosting approach belongs to the ensemble methods and in Scikit-Learn, it is used with decision trees by default. It is the gradient boosting approach with artificially added randomness.[178]

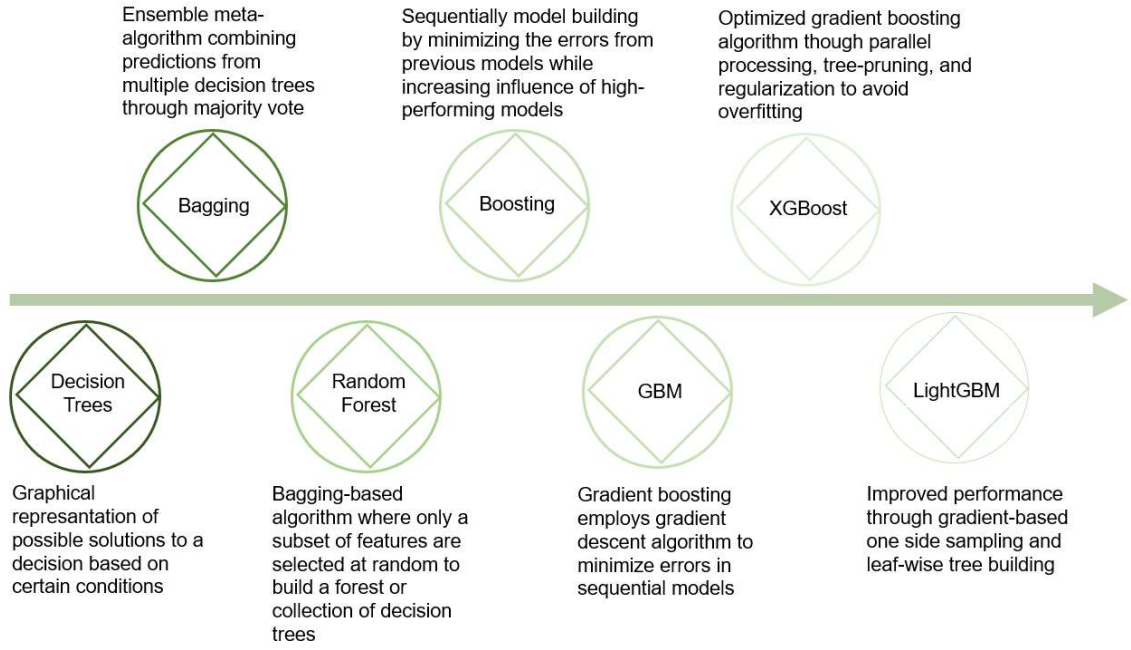
The method is implemented by introducing randomness in the model. Instead of taking the whole dataset for each iteration, a subsample of the training data is randomly drawn without replacement. The new model in each iteration is then used to fit the base learner and calculate the model instead of using the whole dataset. This shall reduce the variance and increase the speed of the approach. A base learner is a machine learning approach that is used in the ensemble and then combined together.[179] Experience shows that aggressive subsampling (e.g. 50 % of the data) enables good results.[180]

[176] Cf. Tseng, 2018.
[177] Cf. ODSC, 2018.
[178] Cf. Brownlee, 2020.
[179] Cf. Friedman, 2002.
[180] Cf. Brownlee, 2019.

Extreme gradient boosting

XG-Boost is an advanced version of the gradient boosting approach. The abbreviation is short for “extreme gradient boosting”. It is one of the most recent developments in the evolution of gradient boosting.[181] Since the introduction in 2014 at the University of Washington, its popularity rapidly increased by being the winning approach in many competitions on data science platforms. It is also a decision-tree based approach.[182] The evolution of tree-based approaches can be seen below:

Figure 10 Evolution of decision tree-based ensemble methods



Source: Own depiction based on Morde, 2019.

Stochastic gradient boosting and extreme gradient boosting are both based on the same idea of using boosting and the gradient descent to minimize the cost function and enable good predictions. However, there are key differences that make XG-Boost faster, more efficient and even better in predicting. XG-Boost uses the second order partial derivatives as a proxy for minimizing the loss function of the base model (decision-tree in this case) instead of the first order derivative. This is intended to provide more information about directions of gradients on the one hand and how to minimize the loss function on the other hand.

[181] Cf. Akerkar, 2019, p. 29 - 30.
[182] Cf. Morde, 2019.

Additionally, XG-Boost uses L1 & L2 regularization to reduce overfitting and to improve the results. In XG-Boost, the building of nodes within each tree can be parallelized to increase efficiency.[183] Another advantage is the usage of multicores which reduces the training time significantly.[184]

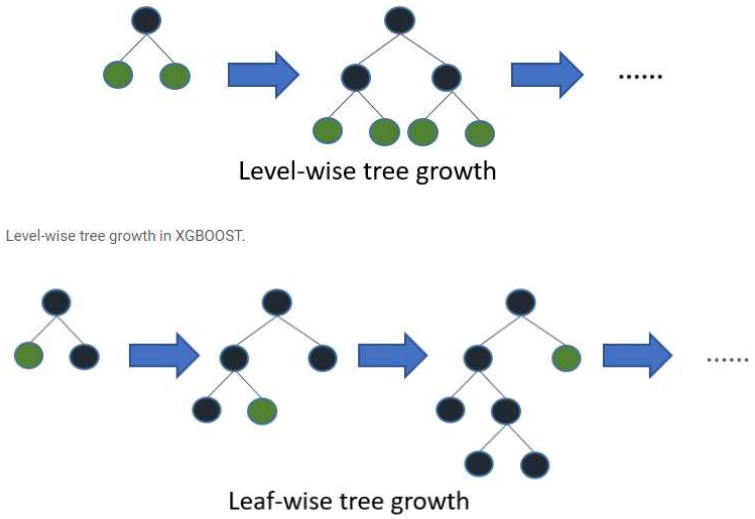
Light gradient boosting model (LightGBM)

Since the world of data science is fast moving, there is already another evolution of the gradient boosting technique, called LightGBM. In fact, it is a further development of the XG-Boost and is considered faster in computing while the results are even better, or at least comparable. It is also a supervised ensemble method based on decision trees and can be used for classification and regression problems. The extension “Light” shall imply the great computing speed of the approach. It was published by Microsoft in 2017.[185]

A big difference between XG-Boost and LightGBM is the leaf-wise (vertical) tree building in LightGBM instead of level-wise tree building (horizontally) in XG-Boost. This technique shall increase computing speed and enable the user to reduce the loss even more.[186] LightGBM splits the leaf nodes that maximize the information gain in a leaf-wise manner, although this can lead to unbalanced trees. In contrast to that, the XG-Boost builds all nodes on each level before adding more levels.[187]

[183] Cf. Brownlee, 2020.
[184] Cf. Tseng, 2018.
[185] Cf. Khandelwal, 2017.
[186] Cf. Khandelwal, 2017.
[187] Cf. Jansen, 2020.

Figure 11 Leaf-wise tree building



Source: Khandelwal, 2017.

The efficiency is also increased through a different method of computing the best split. The XG-Boost algorithm still considers all possible splits on the available features and uses pre-sorting with the histogram-based algorithm. This means that all the data points for a feature are sorted by their value and then split into discrete bins. These bins are then used to find the split solution along all features.[188]

The LightGBM instead uses the "gradient based one side sampling" (GOSS) method to find the best splits. This is carried out by taking all samples with large gradients (bigger training error) into account and carry out random sampling on instances with small gradients (smaller training error).[189] Points with large gradients are important to find the best split since they have a higher error. This increases efficiency and accuracy. It is assumed that points with smaller values are already well-trained. XG-Boost and LightGBM are constantly updated, so some features that were only used in LightGBM were also implemented in XG-Boost over time.[190]

[188] Cf. Kasturi, 2019.
[189] Cf. Kasturi, 2019.
[190] Cf. Swalin, 2018.

4.8 Emergent law-based statistics (ELBS)

Despite the undisputed good results of the “Black Box” approaches, they carry two main problems. At first, the renowned lack of interpretability and explainability.[191,192] Secondly, the fact that the methods generate different results for the same dataset and target variable, even if they are based on the same assumption of probabilistic statistics. Moreover, the results vary, if carried out with different hyperparameter values.[193]

A solution to the mentioned problems is presented in form of the ELBS-Tool, which is based on emergent law-based statistics, domiciled at the center for emergent law-based statistics at the DHBW in Villingen-Schwenningen, Germany.

It combines a clearly understandable and empirically verifiable way of computing predictions with a performance that is at least comparable, most often, even better than the predictions of the “Black Boxes”.[194,195]

The central idea behind emergent statistics is to find patterns that are always true in the past (emergent laws) in order to predict the future. The laws are found through autonomous search processes and algorithmic testing of hypotheses.[196] The found emergent laws are considered the best estimators for the future since patterns that could be found in the past tend to be repeated in the future. This applies, even though these laws can, explicitly, be falsified in the future.[197]

The main difference between the well-known ML tools and the ELBS approach is the different understanding of statistics. The standard ML-tools are based on the general understanding of probability while the ELBS-Tool uses emergent law-based statistics to make predictions.[198]

[191] Cf. Kuck/Frischhut, 2017, p. 16.
[192] Cf. Kuck/Kratz/Frischhut, 2018, p. 25.
[193] Cf. Kuck, 2019b, Chapter 1.
[194] Cf. UDPL, 2020, Start Page.
[195] Cf. Kuck/Frischhut, 2017, p. 16.
[196] Cf. Kuck/Frischhut, 2017, Chapter 1.
[197] Cf. Kuck/Kuck/Harries, 2015, p. 27.
[198] Cf. UDPL, 2020, Starting Page.

An example is given as follows: in probability-based statistics a statement like “if it has rained today, the probability of rain for tomorrow is 60 % “, is not objectively verifiable nor falsifiable and cannot empirically be proven from past observations.[199]

The same applies to probability-based ML tools which make different predictions, based on the variation of parameters and assumptions. It is impossible to prove one of the claims or theories definitively true or false and one theory often contradicts another.[200] The ELBS methodology on the other hand uses emergent laws to make predictions as follows: “In every sequence of 10 days (each following a rainy day) the relative frequency of rain is at least 60%.” [201]

Per definition, an emergent law is “a pattern as a relation between functions of sequences of measurements. Any pattern which has never been falsified in the past (and thus always has been true) is called an “emergent law”.

Rather than searching for predictions of single observations, the ELBS-Tool searches for patterns in sequences of observations.[202]

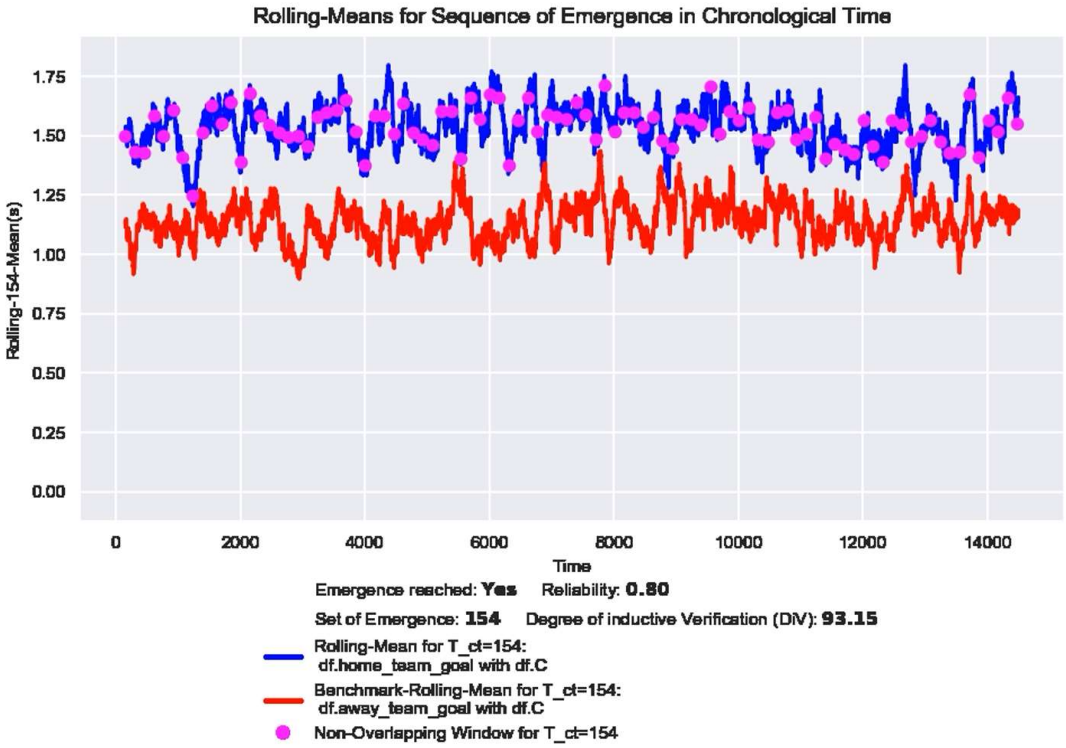
However, considering that only what has always been true in the past can possibly be true in general, emergent laws create the chance to make falsifiable predictions: In case such a prediction is falsified once, the falsified universal statement will remain false forever.[203]

The prediction rule “predict that a pattern that always is true in the past will also become true the next time” cannot result in conflicting predictions. Thus, it becomes possible to collect consistent empirical knowledge of verified and falsified patterns in form of emergent laws in databases which can be made available for future use cases.[204] An example of an emergent law can be observed while examining the concept of home advantage in the top European football leagues in the years between 2004 and 2016.

[199] Kuck/Kuck/Harries, 2015, p. 9.
[200] Cf. Kuck/Kratz/Frischhut, 2018, Chapter 1.1.
[201] UDPL, 2020, Starting Page.
[202] UDPL, 2020, Starting Page.
[203] Cf. Kuck/Frischhut, 2015, p. 7
[204] Cf. Kuck, 2019a.

The graphic below shows that in sequences of 154 games (Set of emergence, T) the rolling mean of goals scored by the Home team (blue) is always higher than the rolling mean of the goals scored by the Away team (red). This pattern is true in 93 cases, meaning it is verified 93 times (Degree of inductive Verification, DiV). However, it must be pointed out that in single games and sequences of other lengths, the Away team of course can score more goals than the Home team. In addition, this pattern can be falsified in the future and will never become true again. Any falsified law never needs to be examined again in the future.[205]

Figure 12 Example law about mean scored goals



Source: UDPL, 2020, Section Emergent Laws.

Meta-laws, on the other hand, are laws about the prediction quality (Reliability) of prediction strategies, usually emergent laws. The laws are found at time point t, for a certain sequence of observations, and confirmed x- times in the past.

[205] Cf. UDPL, 2020, Section Emergent Laws.

Based on these laws found in the past, the future prediction is made that, at a time point t+ T this pattern will hold. [206] For model building in emergent statistics, a minimum prediction quality, i.e. called minimum reliability, can be defined in which only those prediction strategies are consulted that have had a minimum prediction quality.[207] In simple terms: You can search for laws in partial sequences of data and then count how many laws you have found and how often the prediction that the pattern will repeat is correct.[208]

The reliability depends on the DiV. The more often a law has been confirmed in the past, the more likely it is to be confirmed again in the future.[209] The dataset is screened for emergent laws with the same DiV. If it is predicted that the found patterns with the certain DiV will hold in the future, one can evaluate how often this prediction is correct. It is calculated as follows:[210]

Prediction quality/Reliability = True Predictions / Total predictions

It is then possible to make statements about the minimum prediction quality (Reliability) of laws with the same DiV in a dataset.[211]

Figure 13 Minimum Reliability and Degree of Inductive Verification

	DiV_Laws	AirQuality	Lending Club	Crime	Soccer
0	DiV=1	Rel > 0.5	Rel > 0.5	Rel > 0.5	Rel > 0.5
1	1<DiV<=2	Rel > 0.6	Rel > 0.6	Rel > 0.6	Rel > 0.5
2	2<DiV<=4	Rel > 0.7	Rel > 0.7	Rel > 0.7	Rel > 0.6
3	4<DiV<=8	Rel > 0.8	Rel > 0.8	Rel > 0.8	Rel > 0.7
4	8<DiV<=16	Rel > 0.8	Rel > 0.9	Rel > 0.8	Rel > 0.8

Source: Kuck, 2019a.

From the graphic above one can see that the predictions, with patterns that are confirmed only two times in the Lending Club dataset, enable a minimum prediction quality of 60 %, whereas patterns in the same dataset that are confirmed sixteen times make predictions with an accuracy of at least 90 % possible.[212]

[206] Cf. Kuck, 2017, S. 8.
[207] Cf. Kuck/Frischhut, 2017, p. 2.
[208] Cf. Kuck, 2019a, Chapter 2.
[209] Cf. Kuck/Kuck/Harries, 2015, p. 55
[210] Cf. Kuck/Frischhut, 2017, p. 2.
[211] Cf. Kuck, 2019a.
[212] Cf. Kuck, 2019a.

The meta-laws also allow statements about the relative performance or advantage of selection or prediction strategies, which is the basis for the T-Dominance principle. If a decision rule A always led to better results compared to decision rule B after T decisions, rule A is T-Dominant against rule B.[213]

If one focusses again on the example of the European Football Leagues, the pattern that it would have been better to bet on a victory of the Home team (Strategy A) than on the Away team (Strategy B) in sequences of 154 games can be found and it reveals that Strategy B is dominated by Strategy A in sequences of T= 154 games.[214]

The search processes based on the emergent law-based statistics construct and evaluate hypotheses automatically and autonomously.[215] The model building can be described as follows: Not the target variable itself is predicted, but the model searches recursively for estimation heuristics, which would have always improved the prediction accuracy with a baseline Estimator based on simple prediction rules like, for example the rolling or the expanding mean. A better accuracy can be measured through the standard error metrics as explained in chapter 3. It is not searched for laws about the target variable itself but for laws about the prediction error.[216] The search is based on a certain quality level for the laws (Minimum-Reliability), e.g. 80 %. This is controlled by the previous explained Meta-Laws to ensure a good prediction quality.[217]

As mentioned before, the first step is the simple prediction with a baseline estimator that always improved the selected error metric in comparison to the initial benchmark, which is usually set as Zero-Estimation.[218]

The first heuristic that is found is set as the new benchmark and the algorithm continues searching for heuristics to reduce the prediction error of the previous heuristic with simple estimators like expanding or rolling mean as well as the least squares estimator. Afterwards the next found heuristic is the new benchmark and so forth.[219]

[213] Cf. Kuck/Kratz/Frischhut, 2018, Chapter 1.3.
[214] Cf. UDPL, 2020, Section Emergent Laws.
[215] Cf. Kuck/Kuck/Harries, 2015, p. 23
[216] Cf. Kuck/Frischhut, 2017, p. 1.
[217] Cf. Kuck/Frischhut, 2017, p. 5.
[218] Cf. Kuck/Frischhut, 2017, p. 5.
[219] Cf. Kuck/Frischhut, 2017, p. 7.

After the implementation of the first benchmark models, the Knowledge Nets are created. In several iterations, the underlying dataset is examined for selections of observations (Objects), based on a specific combination of features, that always had different mean values with respect to the selected error metric compared to the other created objects. Through this process the algorithm collects information about connections in the database and enables the further development of heuristics to increase the prediction performance.[220]

After the previous explained steps, the prediction model consists of objects with associated heuristics in a certain minimum prediction quality, which always increased the prediction performance.[221] This is carried out until the maximum number of iterations is reached, or the algorithm cannot find any more heuristics which would improve the predictions from the baseline estimator.[222] The complete model consists of the benchmark model, also called model zero, and the model for the selected error metric.[223] The final prediction is based on the combination of the individual heuristics which are found.[224,225] This way of analyzing the data also allows a causal analysis, which is not possible in the “Black Box” approaches.[226]

In conclusion: An emergence-based model is a sequence of always (according to a performance metric) prognosis-improving prognosis heuristics, in a sequence of T estimates. However, this is no guarantee that every individual prediction is correct.[227]

[220] Cf. Kuck/Frischhut, 2017, p. 8.
[221] Cf. Kuck/Frischhut, 2017, p. 9 – 11.
[222] Cf. Kuck/Frischhut, 2017, p. 9 – 11.
[223] Cf. Kuck/Frischhut, 2017, p. 12.
[224] Cf. Kuck/Frischhut, 2017, Chapter 1 - 4.
[225] Cf. Kuck/Frischhut, 2017.
[226] Cf. Kuck/Kratz/Frischhut, 2018.
[227] Cf. Kuck/Kratz/Frischhut, 2018, Chapter 2.2

Unique features

The resulting models from the ELBS-Tool are easier to understand and empirically verifiable. Furthermore, they can be stored, transferred and used in other problems. In addition, a significant difference is that the ELBS-Tool works without any assumptions but with empirical knowledge.[228] In addition, emergent laws are never contradictory.[229] Even more, the ELBS-Tool enables a causal analysis of given information about how to interpret the results while also explaining how they emerge. A unique feature is that each forecast can be explained and viewed individually and in detail as well as the approach as a whole in a comprehensible way.[230]

For the ELBS-Tool, no pre-processing is needed, and it is usable for regression and classification problems as well. Moreover, it is especially useful for more complex problems like time series data and time series panel data. Since the majority of real-world machine learning problems consists of this kind of application fields, it proves, the usefulness of the ELBS-Tool.[231] For the search of meta laws in this bachelor thesis, a minimum reliability of 0.75 is determined. Thus, the laws found are laws for which the prediction of the future confirmation of the pattern is correct in at least 75% of the cases.[232] For more detailed information as well as current projects and papers, see the website https://udpl.info/.

4.9 Summary

In the following, a brief, summarizing description of the individual approaches and their different advantages and disadvantages will be given.

[228] Cf. Kuck/Kratz/Frischhut, 2018, p. 19.
[229] Cf. Kuck, 2019a, Chapter 2.
[230] Cf. Kuck/Kuck/Harries, 2015, Chapter 3.2.
[231] Cf. Kuck/Kratz/Frischhut, 2018, Chapter 2.3.
[232] Cf. Kuck/Kratz/Frischhut, 2018, Chapter 1.3.

Figure 14 Linear-, ridge-, lasso-, and elastic net-regression

The parameters (the distance from the zero point and the slope) are learned by minimizing the deviations from the known data of the examples. In response to new examples the function value is returned. By adding a penalty term and varying its values, the slope of the line can be changed in order to reduce the variance with a small increase in the bias.

	Advantages	Disadvantages
Concept	• Comprehensible structure • Easy to implement and train	• Tendency to overfit
Results	• Good performance for simple, linear separable data	• Worse results for complex problems, highly correlated data, and a high number of features
Efficiency	• Fast model building and predictions	
Other	• Overfitting can be reduced through regularization techniques (Ridge, lasso, elastic net)	

Sources: Own depiction based on Cf. Akerkar, 2019, p. 22 -23, Cf. Kumar, 2019b, p. 22 -23, Cf. Jain, 2017, Cf. Oleszak, 2019.

Figure 15 Logistic regression

The goal is to create a dividing line, which separates e.g. two classes from each other. The parameters of a linear transformation are learned so that a subsequent logistic function can assign a class to each input value. This creates a decision line that can be used to separate unseen data. Nonlinear decision lines can also be learned by appropriate transformation.

	Advantages	Disadvantages
Concept	• Especially useful in classification problems and applicable in non-linear problems	• Not useful for classifications with more than 15 classes
Results	• Good performance for classification problems	• Dependent variable is restricted between 0 and 1, hence not the best choice for continuous data
Efficiency	• Easy implementation and fast training	
Other		• Overfits if the number of features is higher than the number of observations

Source: Own depiction based on Cf. Swaminathan, 2018.

Figure 16 K-nearest neighbor approach

The example data should be divided into clusters so that the examples in one cluster are as similar as possible and as dissimilar as possible to data in other clusters. The similarity is measured using a previously defined distance function. Classification is based on a majority decision within the cluster, continuous data is determined using a mean value based on the cluster.

	Advantages	Disadvantages
Concept	• Uses only two parameters: k and distance function, hence easy to understand and implement	
Results	• Helpful for classifications with irregular decision boundaries	• Usually feature scaling needed for good results
Efficiency	• Useful for noisy data and no training needed due to real time predictions	• Calculation of distance function is computationally expensive in large datasets
Other	• New data will not affect the functionality of the algorithm	

Sources: Own depiction based on Cf. Akerkar, 2019, p. 23- 24, Cf. Scikit-Learn, 2020a, Cf. Kumar, 2019a.

Figure 17 Support vector machine

A support vector machine learns a decision plane in the space of the input data, exactly the one which has the maximum distance to the nearest data points. Non-linear data can be learned in a computationally efficient way using kernel tricks.

	Advantages	Disadvantages
Concept	• Comprehensible structure	• Less comprehensible for high-dimensional data
Results	• Usable and efficient for high-dimensional problems thanks to the kernel trick	• Hyperparameter tuning is needed for good results
Efficiency	• Very memory efficient	• Generalization becomes more complex with a high number of features
Other	• Highly customizable and versatile	• Probabilities are not provided directly but calculated with cross validation

Sources: Own depiction based on Cf. Scikit-Learn, 2020b.

Figure 18 Decision trees (CART)

The model is a decision tree that assigns a numerical value to each end node. This value is to be chosen in the best possible way for exactly those examples that end up in this leaf according to the decisions in the tree. For example, the average of the labels from the training examples of this leaf can be chosen. New examples are assigned the values of their corresponding leaf.

	Advantages	Disadvantages
Concept	• Structure is easy to understand and comprehend	• Trees can become very complex for large datasets
Results	• Good results with automatic feature selection	• Small changes in the dataset can cause large changes in the tree structure and performance
Efficiency	• Fast in model building and predicting • Able to handle numerical and categorical data	• Processing of large datasets occupies a lot of time
Other	• Do not need a lot of preprocessing nor scaling or normalization	• "Greedy nature" of the algorithm • Overfitting is possible, but can be resolved by using pruning, bagging or boosting

Sources: Own depiction based on Cf. Bhattacharjee, 2020, p. 68 – 69, Cf. Akerkar, 2019, p. 24 -27, Cf. Kossen/Müller/Ruckriegel, 2019, p. 111- 118.

Figure 19 Ensemble tools: Bagging with bagged,- extra trees, random forest

Bagging represents parallel model building of independent, more complex models on random subsets of a dataset. By default, decision tree-based models are used as a base learner. By adding randomization in the structure (random forest or extra trees), the variance as a result of overfitting shall be reduced.

	Advantages	Disadvantages
Concept	• Better comprehensibility compared to boosting techniques	• Ensemble methods are inherently less interpretable than single decision trees
Results	• Bagging leads to a better accuracy compared to single decision tree models	
Efficiency	• Robust to outliers and no scaling is needed	• Training of large datasets can occupy a lot of memory. Hence, slower in predicting and training
Other	• Usable even with a huge number of predictive variables and sample sizes	

Sources: Own depiction based on Cf. Akerkar, 2020, p. 26, Cf. Kumar, 2019c.

Figure 20 Ensemble tools: Boosting with AdaBoost, stochastic gradient boosting, XG-Boost and LightGBM

Boosting uses sequential model training with simple models on random samples but different weighting of samples. The errors of previous models, decision trees by default, are used for training and thus reduced. For optimizing the parameters, the gradient boosting technique is used (except for AdaBoost). The addition of randomness shall result in a drop in the variance with a slight increase in the bias.

	Advantages	Disadvantages
Concept		• Less interpretable and explainable compared to "White Boxes"
Results	• Superior results to most ML-approaches, even if only few preprocessing is carried out	• Deep knowledge for correct parameter tuning necessary • Leaf wise model building can lead to overfitting (only LightGBM)
Efficiency	• Flexible, fast and many options for parameter tuning • Possibility of parallel processing • No tendency to overfit (except LightGBM)	• Training of large datasets is usually computationally expensive (except LightGBM) • Carries the risk of local minima • Sensitive and dependent to outliers
Other	• Can handle missing values	• No causal analysis possible

Sources: Own depiction based on Cf. Navlani, 2018, Cf. Github, 2020, Cf. CorporateFinanceInstitute, 2020, Cf. Kumar, 2019d, Cf. Tseng, 2018, Cf. Khandelwal, 2017, Cf. Mandot, 2017.

Figure 21 ELBS-Tool

The algorithm finds patterns that were always true and empirically verifiable in the past in order to predict the future and makes predictions based on them.

	Advantages	Disadvantages
Concept	• Comprehensible, interpretable and empirically verifiable • Opportunity of causal analysis	• Not yet known to the vast majority thus few standard literature or code documentation available
Results	• Superior results to most approaches, at least comparable with leading ML tools	• Observation of patterns in the past is no guarantee for correctness in the future
Efficiency	• No pre-processing is needed, and it is usable for regression and classification problems	
Other	• Each forecast can be explained and viewed individually	

Sources: Own depiction based on Cf. Kuck/Kratz/Frischhut, 2018, p. 19, Cf. Kuck, 2019a, Chapter 2, Cf. Kuck/Kuck/Harries, 2015, Chapter 3.2, Cf. Kuck/Kratz/Frischhut, 2018, Chapter 2, Cf. Kuck/Kratz/Frischhut, 2018, Chapter 1.3.

5 Results for each application field

Procedure

Chapter five and six are dedicated to present the application results of the different machine learning methods on the selected application fields, cross-sectional data, time series data and time series panel data. Each application field consists of different datasets as described in chapter two. The machine learning approaches are used to predict the values of the target variables in each dataset. They contain regression and/ or classification problems. Some datasets have more than one prediction target and can include both kinds of problems.

The prediction performance for the test data is measured with the metrics defined in chapter three. In chapter five, the prediction performance of each approach is compared for each individual application field. Chapter six demonstrates the performance of each approach overall application fields. Through this procedure the central question of the thesis can be answered individually for each application field and combined overall application fields.

Two different rankings

For the performance comparison, two different rankings were established: a standard performance ranking and a T-Dominance ranking. The standard performance ranking is a usual ranking one would expect for a competition. The T-Dominance ranking analyzes the performance for a multitude of different calculation scenarios. Both types are further explained below.

Standard performance ranking

The performance of each approach in predicting the testing data of the different target variables is measured with the renowned metrics: MAPE, MSPE, AUC, CORR. For each target variable in each metric a ranking in descending order is established, sorted by the performance in predicting the target variables correctly. The first place is taken by the best performance, the last place by the weakest performance. Afterwards, the rankings for each target variable (separate for each metric) are summed up and divided by the number of target variables which creates an average ranking of the approaches for each metric.

For the results in chapter five only the performance for the respective target variables for each individual application field were considered. The results in chapter six combine the performance for all 88 target variables in the three application fields.

T-Dominance ranking

In practical use of machine learning, a data scientist tries to find an approach that achieves the best results for a specific problem. In reverse, that means he tries to identify the tool that was least dominated by others. This idea is expressed in the performance results with the T-Dominance ranking, where it is counted how often an approach is dominated by another. This criterion is more suitable to answer the initial question, since the calculations are based on a multitude of different calculation scenarios.[233]

The T-Dominance criterion is a term from emergent law-based statistics to explain dominance relations between objects or decision rules. In this bachelor thesis, the objects are the predictor models.[234] A decision rule A that has always been better than a decision rule B in T-sequences, with respect to an evaluation metric, is called T-Dominant. The forecasting quality or advantageousness is measured using the previously defined criteria. T is the number of sequences.[235]

The difference to the standard performance ranking is that the performance of the approaches is measured in rolling means and in a multitude of different calculation lags. A lag is the size of a calculation window.

For the comparison, the rolling means for each metric by each approach for every prediction goal have been calculated for a multitude of different calculation lags. Afterwards, the results for each lag are compared. It is counted how often any other approach dominated the selected one (Number of being dominated, “NumBeDom”). Dominated in this case means that the results of another approach have been, with respect to the specific prediction goal and the specific metric, better in all calculated lags.

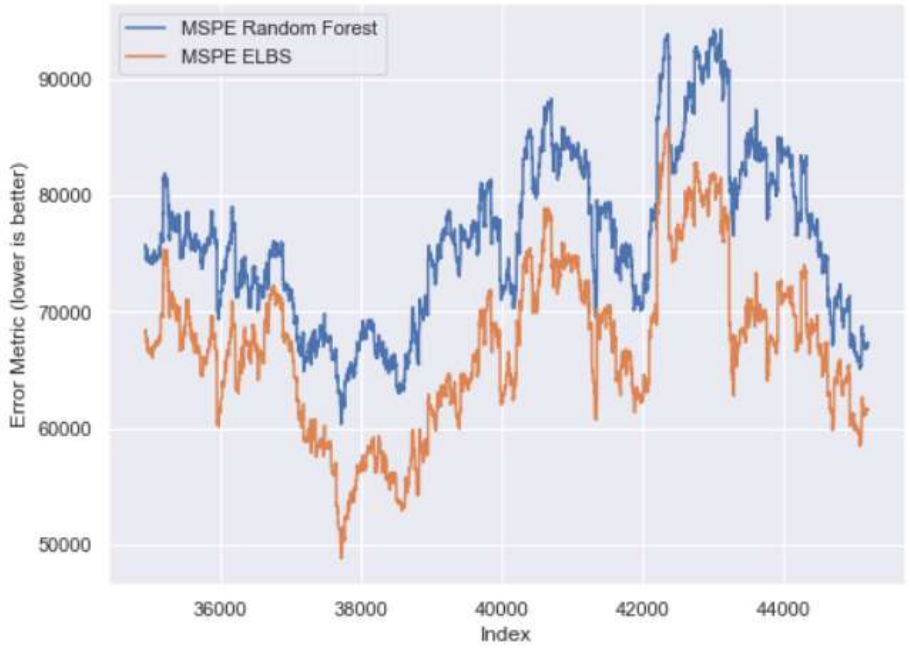
[233] Cf. Kuck/Kratz/Frischhut, 2018, Chapter 1.2.
[234] Cf. Kuck/Frischhut, 2015, p. 11 – 12.
[235] Cf. Kuck/Frischhut, 2015, p. 11 – 12.

Subsequently, the number of how often an approach is dominated is counted overall datasets, summed up and divided by the number of calculation lags that are carried out. The result is an average number, of how often an approach is dominated in each metric.

As for the standard performance ranking, in chapter five the T-Dominance rankings for each individual application field are analyzed, whereas in chapter six the T-Dominance rankings combined for all 88 target variables overall application fields are displayed. Since an analysis of rolling windows is required for the calculation of the T-Dominance and the AUC just views the overall distribution, it could not be considered for the T-Dominance ranking.

In the figure below an example for the T-Dominance criterion is displayed. It shows the comparison of the rolling means for the MSPE criterion overall calculation lags between the ELBS-Tool and random forest approach in the bank marketing dataset. It reveals that the rolling mean of the MSPE is always higher for the random forest method. Since the goal is to minimize this metric, the random forest method is dominated by the ELBS-Tool in predicting the call duration in the bank marketing dataset.

Figure 22 Example MAPE- Bank marketing call duration ELBS vs RF



Source: Own depiction based on Appendix 13.

5.1 Cross-sectional data

In this chapter the performance of the approaches in the cross-sectional datasets will be analyzed using the standard performance ranking first and second the T-Dominance ranking.

5.1.2 Standard performance ranking- Cross-sectional data

The tables below give an overview of the performance of the different approaches in the cross-sectional data in each metric for the standard performance ranking.

Figure 23 Standard performance ranking per metric- Cross-sectional data

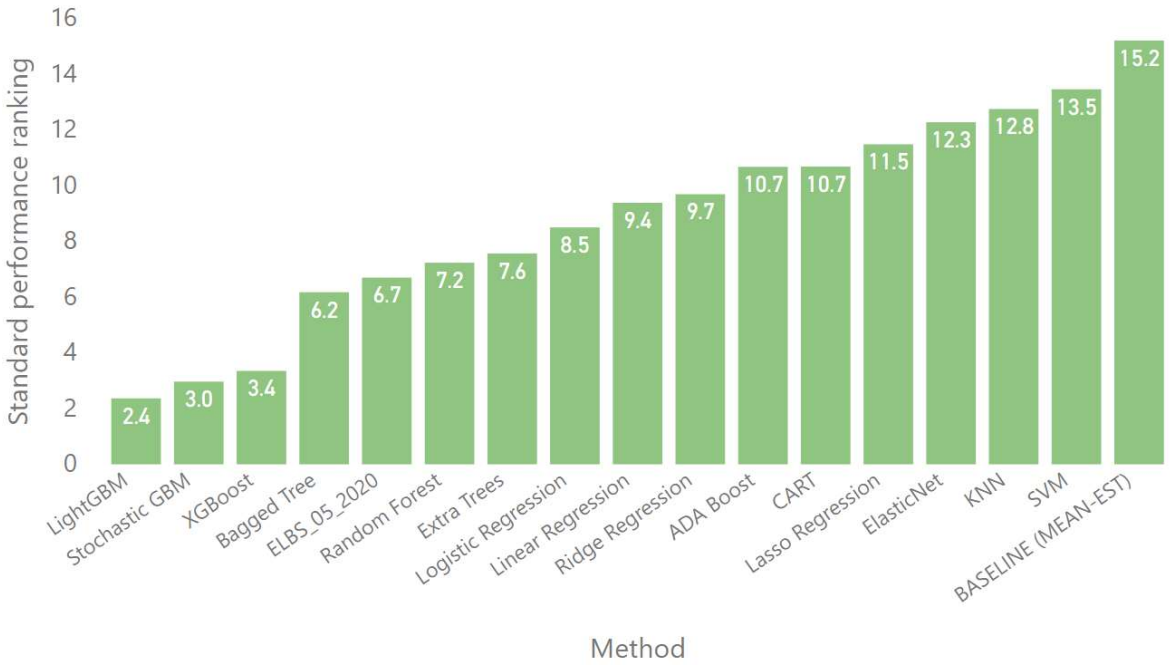
Method	Rank_MAPE	Method	Rank_MSPE	Method	Rank_CORR	Method	Rank_AUC
LightGBM	2.70	LightGBM	2.40	LightGBM	2.20	LightGBM	2.17
Stochastic Gradient Boosting	4.70	Stochastic Gradient Boosting	2.40	Stochastic Gradient Boosting	2.60	Stochastic Gradient Boosting	2.17
Ridge Regression	5.20	XGBoost	2.90	XGBoost	2.80	XGBoost	2.50
XGBoost	5.20	Bagged Tree	5.90	Bagged Tree	5.60	ADA Boost	5.00
ELBS_05_2020	6.60	ELBS_05_2020	6.80	ELBS_05_2020	6.90	Bagged Tree	6.50
Bagged Tree	6.70	Random Forest	6.90	Random Forest	7.20	ELBS_05_2020	6.50
Random Forest	7.00	Extra Trees	7.20	Logistic Regression	7.33	Random Forest	7.83
Extra Trees	7.20	Logistic Regression	7.33	Extra Trees	7.70	Extra Trees	8.17
CART	7.50	Linear Regression	8.90	Linear Regression	8.90	Linear Regression	8.33
SVM	8.10	Lasso Regression	9.60	Ridge Regression	9.10	Logistic Regression	8.33
Logistic Regression	11.00	Ridge Regression	10.80	ADA Boost	9.40	CART	11.83
Linear Regression	11.40	ElasticNet	10.90	CART	10.60	ElasticNet	12.00
KNN	12.20	CART	12.80	Lasso Regression	11.70	Lasso Regression	12.33
Lasso Regression	12.30	BASELINE (MEAN-EST)	13.00	KNN	12.80	KNN	13.00
ElasticNet	13.30	KNN	13.00	ElasticNet	12.90	Ridge Regression	13.67
BASELINE (MEAN-EST)	14.70	ADA Boost	13.50	SVM	15.10	SVM	15.83
ADA Boost	14.80	SVM	14.80	BASELINE (MEAN-EST)	16.30	BASELINE (MEAN-EST)	16.83

Source: Own depiction based on Appendix 13.

As figure 23 displays, the LightGBM approach achieves the first place with an average ranking better than position three in all of the metrics. The LightGBM approach is followed by the stochastic gradient boosting approach in all metrics. The third place is taken by the XG-Boost approach, with the exception of the MAPE criterion where the ridge regression is on third position. As the two middle tables show, the bagged tree approach was in fourth place for the MSPE and CORR criterion. The ELBS-Tool is next in all metrics, with the exception of the AUC criterion as the right table reveals, where the AdaBoost and bagged tree approaches are ranked on fourth and fifth position.

The AUC metric can only be used for classification problems. As can be seen, not only for the combination of regression and classification problems combined but also for the classification problems alone the gradient boosting models rank best in the standard performance ranking for the cross-sectional data. The individual rankings for each metric can be combined by summing up the ranking of each approach for each metric and dividing the result by the number of metrics (4). The combined ranking is shown below.

Figure 24 Standard performance ranking overall metrics- Cross-sectional data



Source: Own depiction based on Appendix 13.

Figure 24 reveals that the top group is formed by ensemble models which use boosting. The gradient boosting models LightGBM with an average ranking of 2.4, stochastic gradient boosting with 3.0, and XG-Boost with 3.4 are placed on 1 to 3, followed by the bagged tree approach and the ELBS-Tool. Ranked after the ELBS-Tool are the other decision tree-based bagging methods random forest and extra tree. After them the “White Box” approaches are ranked and the performance decreases as can be seen in figure 24. However, the “Black Box” approaches SVM and AdaBoost are ranked among the “White Box” methods.

5.1.3 T-Dominance ranking- Cross-sectional data

The tables that are displayed below demonstrate the T-Dominance ranking of the approaches in the cross-sectional data for each metric. Since the AUC criterion cannot be used for the T-Dominance criterion, only the MAPE, MSPE and CORR criterion are considered.

Figure 25 T-Dominance ranking per metric- Cross-sectional data

Method	NoBeDom_MAPE
LightGBM	1.50
Stochastic Gradient Boosting	3.20
Ridge Regression	3.60
XGBoost	3.90
Bagged Tree	5.20
CART	5.20
ELBS_05_2020	5.50
Random Forest	5.50
Extra Trees	5.90
SVM	6.70
Logistic Regression	9.67
Linear Regression	10.10
Lasso Regression	10.90
KNN	11.00
ElasticNet	12.20
ADA Boost	13.70
BASELINE (MEAN-EST)	13.70

Method	NoBeDom_MSPE
LightGBM	0.80
Stochastic Gradient Boosting	1.20
XGBoost	1.30
Bagged Tree	4.10
Random Forest	5.10
ELBS_05_2020	5.40
Extra Trees	5.70
Logistic Regression	6.17
Linear Regression	6.90
Lasso Regression	8.50
Ridge Regression	9.30
ElasticNet	9.90
CART	11.40
KNN	11.70
BASELINE (MEAN-EST)	11.80
ADA Boost	12.40
SVM	13.50

Method	NoBeDom_CORR
LightGBM	0.70
XGBoost	1.10
Stochastic Gradient Boosting	1.30
Bagged Tree	4.20
ELBS_05_2020	5.30
Random Forest	5.30
Extra Trees	5.90
Logistic Regression	6.00
Linear Regression	7.50
Ridge Regression	7.70
ADA Boost	7.80
CART	9.20
Lasso Regression	10.00
SVM	10.40
ElasticNet	11.50
KNN	11.60
BASELINE (MEAN-EST)	15.20

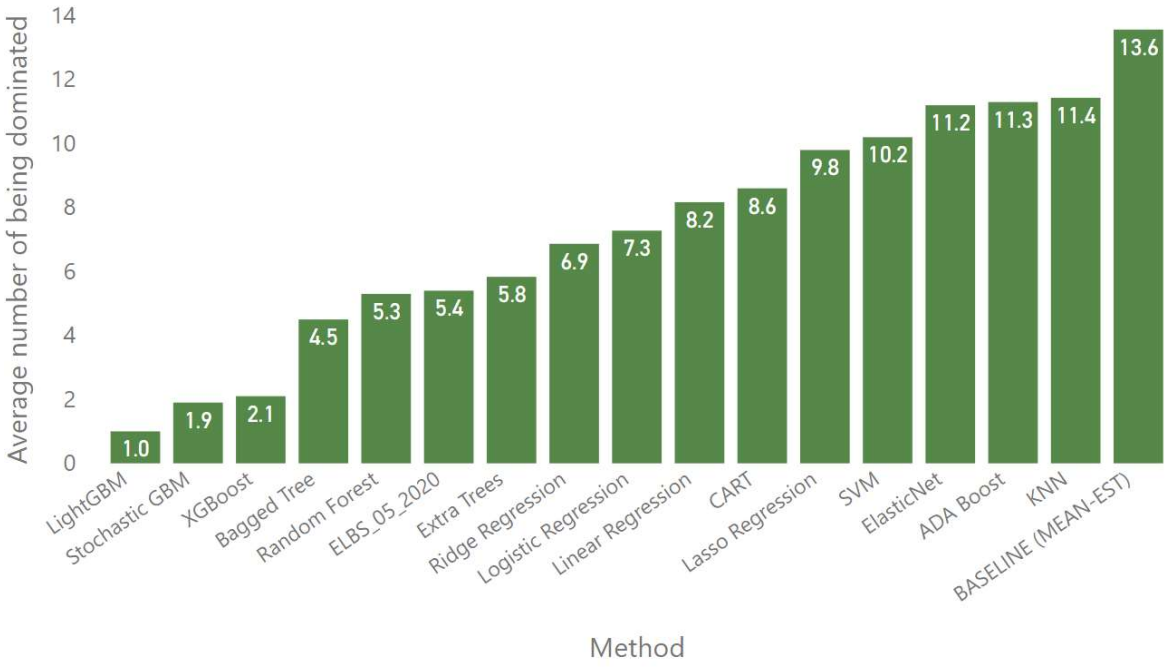
Source: Own depiction based on Appendix 13.

As figure 25 demonstrates, the results for the standard performance ranking resemble the results for the T-Dominance ranking

Regarding all performance criteria, the LightGBM approach achieves again the best results and was least dominated, followed by the other gradient boosting models, except for the MAPE criterion in the left table. In this case the ridge regression approach is placed before the XG-Boost. The bagged tree approach is placed in fourth position for the MSPE and CORR criteria and fifth in the MAPE criterion. The ELBS-Tool only takes the fifth place in the CORR criterion, the sixth position in the MSPE and seventh position in the MAPE criterion. since the random forest in the MSPE criterion and CART in the MAPE criterion are less dominated.

Similar to the standard performance ranking, the individual rankings for each metric can be combined by summing up the rankings of each approach for each metric and dividing the result by the number of metrics (3). The combined T-Dominance ranking is shown below in figure 26.

Figure 26 T-Dominance ranking overall metrics- Cross-sectional data



Source: Own depiction based on Appendix 13.

If the central question of this bachelor thesis is focused, the following results for the best approaches in cross-sectional data emerge from figure 24 and figure 26: The best approaches in the cross-sectional data can be grouped into the ensemble methods using boosting: LightGBM, stochastic gradient boosting, XG-Boost and the bagged tree approach. They are followed by the random forest method and the ELBS-Tool. As the results show, the decision tree-based boosting technique is useful in combination with less complex, cross-sectional data. The “White Box” approaches and the SVM and AdaBoost approaches are not a good choice for estimating cross-sectional data as the figures 24 and 26 reveal. Nevertheless, the performance of the latter ones could be improved by further hyperparameter tuning.[236]

[236] Cf. Bakharia, 2016.

Conclusion for cross-sectional data

Both rankings, the standard performance ranking and T-Dominance ranking are led by the GBM models, which lead the standard performance ranking and the ELBS-Tool, which leads the T-Dominance ranking. On the next positions are the decision tree-based approaches, random forest, extra tree and bagged tree ranked among the ELBS-Tool. They are followed by the “White Box” approaches and two “Black Box” methods, AdaBoost and SVM.

5.2 Time series data

Chapter 5.2 shows the performance of the approaches in the time series datasets using the standard performance ranking first and second the T-Dominance ranking. The time series datasets only contain regression problems. For this reason, the AUC criterion could not be included since it is only applicable in classification problems.

5.2.1 Standard performance ranking- Time series data

The graphic displayed below demonstrates the performance of the approaches in the time series data in each metric for the standard performance ranking.

Figure 27 Standard performance ranking per metric- Time series data

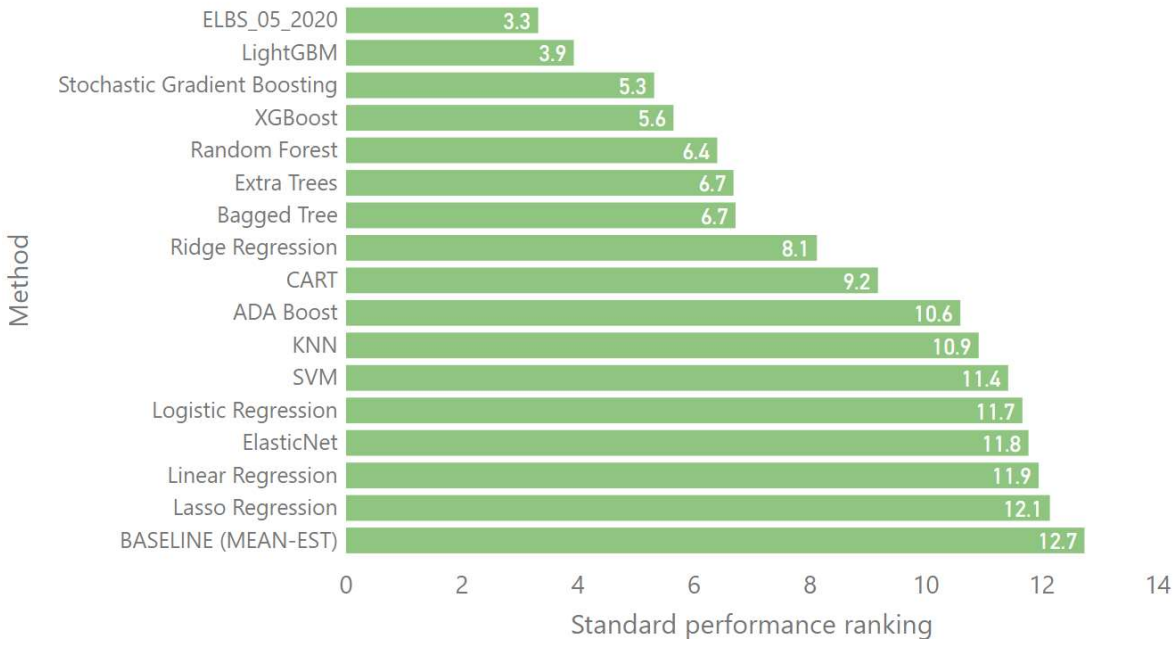
Method	Rank_MAPE	Method	Rank_MSPE	Method	Rank_CORR	Method	Rank_AUC
ELBS_05_2020	2.16	ELBS_05_2020	3.53	LightGBM	2.89	ADA Boost	NaN
LightGBM	4.74	LightGBM	4.16	ELBS_05_2020	4.26	Bagged Tree	NaN
Extra Trees	5.84	Stochastic Gradient Boosting	5.32	Stochastic Gradient Boosting	4.68	BASELINE (MEAN-EST)	NaN
Random Forest	5.89	XGBoost	5.32	XGBoost	5.42	CART	NaN
Stochastic Gradient Boosting	5.95	Random Forest	7.05	Random Forest	6.26	ElasticNet	NaN
Bagged Tree	6.00	Bagged Tree	7.53	Extra Trees	6.32	ELBS_05_2020	NaN
XGBoost	6.21	Extra Trees	7.89	Bagged Tree	6.63	Extra Trees	NaN
CART	9.00	Ridge Regression	8.32	Ridge Regression	6.79	KNN	NaN
Ridge Regression	9.26	CART	10.00	CART	8.53	Lasso Regression	NaN
KNN	10.37	ADA Boost	10.42	ADA Boost	9.68	LightGBM	NaN
SVM	10.84	ElasticNet	10.47	Logistic Regression	10.33	Linear Regression	NaN
Logistic Regression	11.67	Lasso Regression	10.79	Linear Regression	11.11	Logistic Regression	NaN
ADA Boost	11.68	BASELINE (MEAN-EST)	10.84	KNN	11.32	Random Forest	NaN
ElasticNet	12.00	KNN	11.05	SVM	11.68	Ridge Regression	NaN
Lasso Regression	12.05	SVM	11.74	ElasticNet	12.84	Stochastic Gradient Boosting	NaN
BASELINE (MEAN-EST)	12.32	Linear Regression	12.21	Lasso Regression	13.58	SVM	NaN
Linear Regression	12.53	Logistic Regression	13.00	BASELINE (MEAN-EST)	15.05	XGBoost	NaN

Source: Own depiction based on Appendix 13.

As can be seen in figure 27, for the MAPE and MSPE criteria the ELBS-Tool leads the standard performance ranking. The only exception is the CORR criterion which is led by the LightGBM approach and the ELBS-Tool is ranked second.

In the MSPE and CORR metrics, the LightGBM approach and ELBS-Tool are followed by the remaining gradient boosting models stochastic GBM and XG-Boost on third and fourth place as well as the random forest method on fifth. In contrast to that, the extra tree and random forest approaches are ranked on third and fourth position in the MAPE criterion. The individual rankings for each metric can be brought together in one combined overview by summing up the rankings of each approach for each metric and dividing the result by the number of metrics (3). The combined standard performance ranking for the time series data is shown below in figure 28.

Figure 28 Standard performance ranking overall metrics- Time series data



Source: Own depiction based on Appendix 13.

The combined overview in figure 28 confirms the results for the individual metrics in figure 27: The ELBS-Tool is ranked on first place with an average ranking in the metrics of 3.3, followed by the three gradient boosting models and the random forest method on fourth position with an average ranking of 6.4. This top group of is followed by the other decision-tree based methods, extra trees and bagged trees.

5.2.2 T-Dominance ranking- Time series data

The tables in the graphic below present the T-Dominance ranking of the approaches in the time series data for each metric.

Figure 29 T-Dominance ranking per metric- Time series data

Method	NoBeDom_MAPE
ELBS_05_2020	0.95
LightGBM	3.35
Random Forest	4.15
Extra Trees	4.30
Stochastic Gradient Boosting	4.35
XGBoost	4.40
Bagged Tree	4.50
CART	6.85
Ridge Regression	7.70
KNN	8.30
SVM	9.25
Lasso Regression	9.55
BASELINE (MEAN-EST)	9.90
ADA Boost	9.95
ElasticNet	10.10
Logistic Regression	10.67
Linear Regression	10.85

Method	NoBeDom_MSPE
ELBS_05_2020	2.21
LightGBM	2.60
Stochastic Gradient Boosting	3.85
XGBoost	3.95
Random Forest	5.40
Bagged Tree	5.80
Extra Trees	6.35
Ridge Regression	6.85
CART	8.30
ADA Boost	8.65
ElasticNet	8.80
Lasso Regression	8.85
BASELINE (MEAN-EST)	9.20
KNN	9.45
SVM	9.75
Linear Regression	10.65
Logistic Regression	10.67

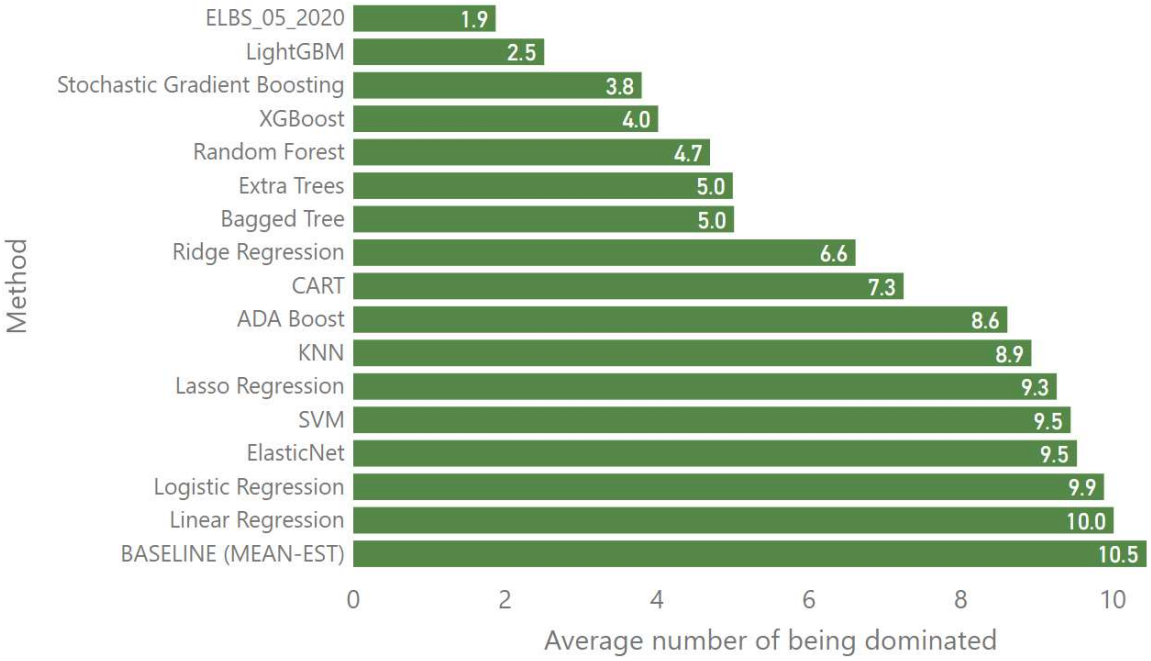
Method	NoBeDom_CORR
LightGBM	1.60
ELBS_05_2020	2.47
Stochastic Gradient Boosting	3.20
XGBoost	3.70
Extra Trees	4.35
Random Forest	4.55
Bagged Tree	4.75
Ridge Regression	5.30
CART	6.60
ADA Boost	7.25
Logistic Regression	8.33
Linear Regression	8.55
KNN	9.05
SVM	9.35
Lasso Regression	9.40
ElasticNet	9.70
BASELINE (MEAN-EST)	12.25

Source: Own depiction based on Appendix 13.

The results for the T-Dominance ranking in the time series data are similar to the standard performance ranking, as figure 29 reveals. In the MAPE and MSPE criteria the ELBS-Tool leads the field and was least dominated with values of 0.95 and 2.21. The LightGBM method is ranked second in these criteria, except for the CORR criterion in the right table where they switch positions and LightGBM is ranked best. They are followed by the stochastic gradient boosting method on third position, in the MSPE and CORR criteria (middle and right table). Only for the MAPE metric in the left table, the random forest and extra tree methods achieve superior results to the stochastic GBM. After these approaches, the XG-Boost method is sixth least dominated in the MAPE criterion, whereas it could achieve fourth position in the MSPE and CORR tables.

Each metric's individual ranking can be combined in one overview by summing up the rankings of each approach for each metric and dividing the result by the number of metrics (3). The combined T-Dominance ranking is displayed below in figure 30.

Figure 30 T-Dominance ranking overall metrics- Time series data



Source: Own depiction based on Appendix 13.

Figure 30 confirms the previous results of the standard performance ranking and the T-Dominance ranking. The results show the superior performance of the ELBS-Tool in the application field of time series data as it takes the first place and was 1.9 times dominated on average overall three metrics MAPE, MSPE and CORR. As the results show it is outstanding useful for more complex problems like times-series data. Like the results for the cross-sectional data, the gradient boosting models LightGBM, XG-Boost and stochastic gradient boosting achieve good rankings as well. The random forest, bagged tree and extra tree methods are also good estimators for time series predictions, since the results are close to the gradient boosting models. Moreover, they represent the boundary of better results for the “Black Boxes” and worse results with a better comprehensibility in the “White Box” approaches. The only exceptions out of the “Black Box” approaches are again, the SVM and AdaBoost estimators with weak results, even worse than many “White Boxes”.

The ELBS-Tool achieves the highest ranking, is the approach that is least often dominated and achieves the best standard performance ranking. Therefore, it can be seen as the best choice for time series data.

Conclusion for time series data

The top rankings are achieved by the GBM models and the ELBS-Tool. The ELBS-Tool leads the standard performance ranking as well as the T-Dominance ranking. The following positions in both rankings are taken by the other decision tree-based approaches, random forest, extra tree and bagged tree. On the other positions the "White Box" approaches are ranked among the SVM and AdaBoost methods.

5.3 Time series panel data

In this chapter the performance results of the approaches in the time series panel datasets using the standard performance ranking first and second the T-Dominance ranking are presented.

5.3.1 Standard performance ranking- Time series panel data

In figure 31 the performance of the approaches in the time series panel data in each metric for the standard performance ranking is displayed.

Figure 31 Standard performance ranking per metric- Time series panel data

Method	Rank_MAPE	Method	Rank_MSPE	Method	Rank_CORR	Method	Rank_AUC
ELBS_05_2020	5.05	ELBS_05_2020	4.72	LightGBM	4.26	ADA Boost	2.33
XGBoost	5.12	BASELINE (MEAN-EST)	4.88	XGBoost	4.91	ELBS_05_2020	5.67
LightGBM	5.28	XGBoost	5.54	Stochastic Gradient Boosting	5.30	LightGBM	5.67
BASELINE (MEAN-EST)	5.67	LightGBM	5.61	Bagged Tree	6.28	Random Forest	6.00
Stochastic Gradient Boosting	6.19	Stochastic Gradient Boosting	6.12	ELBS_05_2020	6.74	XGBoost	6.00
Extra Trees	7.54	ElasticNet	7.39	Random Forest	6.93	Linear Regression	7.67
ElasticNet	8.05	Lasso Regression	7.40	Extra Trees	7.00	Ridge Regression	7.67
Lasso Regression	8.14	ADA Boost	7.77	ADA Boost	8.33	Bagged Tree	8.00
Bagged Tree	8.19	SVM	8.39	SVM	8.68	Stochastic Gradient Boosting	8.33
Random Forest	8.44	Extra Trees	8.51	CART	9.12	Extra Trees	10.67
ADA Boost	8.67	Bagged Tree	8.74	KNN	9.49	Logistic Regression	10.67
SVM	8.79	Random Forest	8.93	Ridge Regression	10.53	SVM	10.67
KNN	10.02	KNN	9.82	Logistic Regression	11.00	CART	11.00
Ridge Regression	12.42	Logistic Regression	12.67	Linear Regression	11.09	Lasso Regression	11.33
CART	13.26	Ridge Regression	13.42	BASELINE (MEAN-EST)	11.79	ElasticNet	12.00
Logistic Regression	15.00	CART	13.61	ElasticNet	12.72	KNN	14.00
Linear Regression	15.26	Linear Regression	15.37	Lasso Regression	13.14	BASELINE (MEAN-EST)	15.33

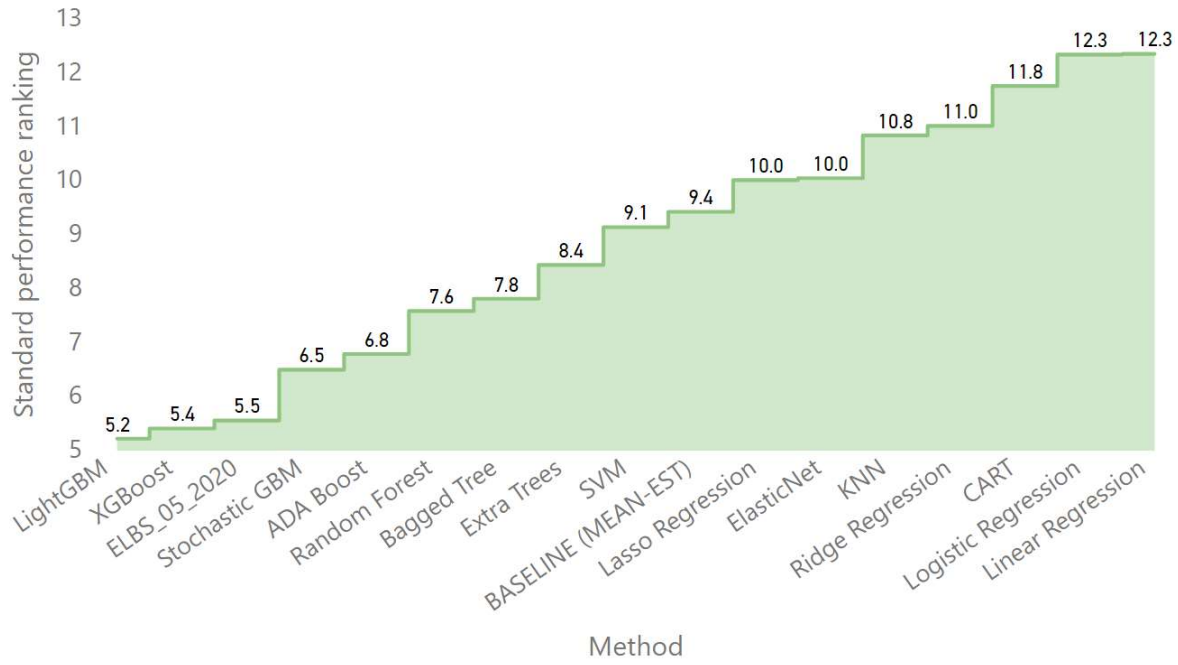
Source: Own depiction based on Appendix 13.

From figure 31 it can be observed that the ELBS-Tool reached the best standard performance ranking in the MAPE and MSPE criteria with an average ranking of 5.05 and 4.72. However, it only ranks fifth in the CORR metric but took the second position in the AUC criteria as can be seen in the right table.

The ELBS-Tool is followed by the LightGBM and XG-Boost approaches in the MAPE and MSPE criteria, whereas they achieve the first two position in the CORR metric. The remaining GBM model, stochastic gradient boosting, achieves good results as well but is only placed in the midfield for the AUC criterion. Since the AUC metric only considers classification problems, the results imply that stochastic gradient boosting is not a good choice for time series panel data. The baseline estimator achieves ranks in the top positions for the MAPE and MSPE metrics, where it takes the fourth and second position. Since the ADABoost approach leads the ranking in the AUC criterion the results imply that it is a good choice for classification problems in time series panel data.

Each metric's individual ranking can be brought together in a comprehensive overview by summing up the rankings of each approach for each metric and dividing the result by the number of metrics (4). This comprehensive standard performance ranking is shown below in figure 32.

Figure 32 Standard performance ranking overall metrics- Time series panel data



Source: Own depiction based on Appendix 13.

The combined standard performance ranking for all metrics in figure 32 enhances the impression from the results for the individual metrics: the top positions are occupied by the GBM models as well as the ELBS-Tool as the positions one to four show. Although the ADABoost approach and baseline estimator achieved top positions in some single metrics they are not ranked in the top group when all metrics are considered together.

5.3.2 T-Dominance ranking- Time series panel data

The figure 33 below displays the T-Dominance ranking of the approaches in the time series panel data for each metric.

Figure 33 T-Dominance ranking per metric- Time series panel data

Method	NoBeDom_MAPE
ELBS_05_2020	2.67
XGBoost	3.33
LightGBM	3.82
BASELINE (MEAN-EST)	4.30
Stochastic Gradient Boosting	4.30
Lasso Regression	5.16
ElasticNet	5.33
Extra Trees	5.86
ADA Boost	6.47
Bagged Tree	6.58
SVM	6.61
Random Forest	6.72
KNN	7.82
Ridge Regression	10.93
CART	11.89
Logistic Regression	14.00
Linear Regression	14.25

Method	NoBeDom_MSPE
ELBS_05_2020	2.60
XGBoost	3.70
BASELINE (MEAN-EST)	3.72
LightGBM	4.09
Stochastic Gradient Boosting	4.30
Lasso Regression	4.40
ElasticNet	4.60
ADA Boost	5.39
SVM	6.21
Bagged Tree	7.00
Extra Trees	7.00
Random Forest	7.44
KNN	7.93
Logistic Regression	11.67
Ridge Regression	11.96
CART	12.46
Linear Regression	14.28

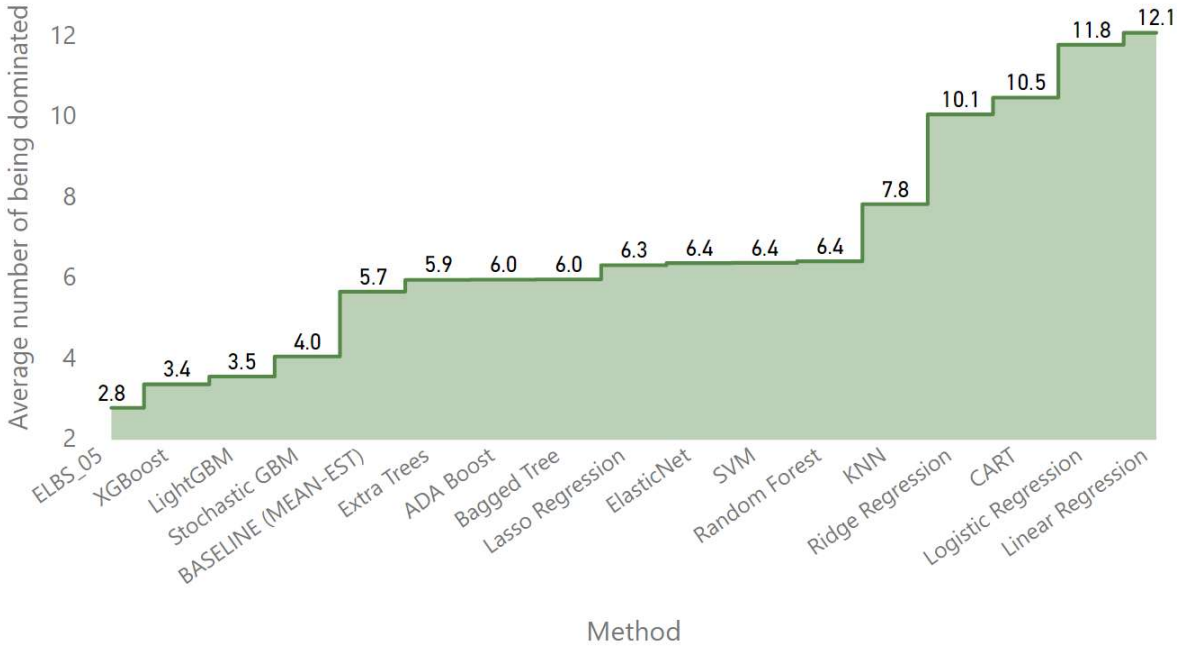
Method	NoBeDom_CORR
LightGBM	2.74
XGBoost	3.04
ELBS_05_2020	3.05
Stochastic Gradient Boosting	3.54
Bagged Tree	4.30
Extra Trees	4.98
Random Forest	5.07
ADA Boost	6.00
SVM	6.28
CART	7.05
Ridge Regression	7.26
Linear Regression	7.70
KNN	7.72
BASELINE (MEAN-EST)	8.95
ElasticNet	9.16
Lasso Regression	9.37
Logistic Regression	9.67

Source: Own depiction based on Appendix 13.

As figure 33 reveals, the ELBS-Tool was least dominated overall calculation scenarios in the MAPE and MSPE criteria and leads the T-Dominance ranking in these metrics. In these metrics it is followed by the GBM-models and the baseline estimator as the middle and left tables show. However, the GBM models LightGBM and XG-Boost are able to achieve even better scores in the CORR criterion and are ranked on the first and second position. In contrast, the baseline estimator is not a good choice when the CORR metric is considered as the right table shows. For the CORR criterion the ELBS-Tool was only third least dominated overall calculation scenarios.

As well as for the other application fields, the individual ranking of each metric can be combined in a comprehensive overview by summing up the rankings of each approach for each metric and dividing the result by the number of metrics (3). This overview is shown below in figure 34.

Figure 34 T-Dominance ranking overall metrics- Time series panel data



Source: Own depiction based on Appendix 13.

The impression that emerges for the time series panel data overall metrics in figure 34 and 32 coincide with the previous results. The ELBS-Tool and gradient boosting approaches form the top group. In contrast to the results for the cross-sectional and time series data, not only the remaining decision-tree based ensemble methods are the next best group, but also the AdaBoost and baseline estimator. The results imply that the AdaBoost estimator is a better choice for more complex data than for easier datasets like cross-sectional data. Although the baseline estimator is located in the midfield in the standard performance ranking, it is the approach that is least dominated after the top group. Since the T-Dominance ranking is the more important criterion, because it shows the universal usability and constancy in many calculation scenarios, it indicates the difficulty of making predictions for complex data and sometimes the simple estimation with the expanding mean can save a lot of work and time.

A well-known phenomenon in data science is the fact that a good baseline estimator is sometimes able to achieve comparable results to the far more complex models in time series and panel application fields. This observation can also be made for the used baseline estimator (expanding mean) in the time series panel data. Since this structure of data belongs to the most complex predictable application fields, the difficulty of building models to predict such datasets becomes clear. The problem is most likely, that the complex models tend to overfit in such complex and large datasets.[237] However, it has to be stated that this application field, with the DAX and macro dataset, only contains macro-economic target variables.

Conclusion for time series panel data

The top positions are taken by the GBM models, which lead the standard performance ranking and the ELBS-Tool, which leads the T-Dominance ranking. They are followed by the remaining decision tree-based approaches, random forest, extra tree and bagged tree. The AdaBoost is also able to achieve good results in this field, whereas the baseline estimator with the expanding mean shows its strength through simplicity in this field.

Due to the missing standard architecture no typical deep learning approaches for time series and panel data forecasting, like convolutional neural networks (CNN) or multilayer perceptrons (MLP) are applied. It can be assumed that they would have achieved significant good results in this field.[238]

6 Results combined overall application fields

This chapter is dedicated to present the performance of each approach not for single application fields but for all three application fields combined. At first the results will be presented using the standard performance ranking and second the T-Dominance ranking is presented.

6.1 Standard performance ranking- Overall application fields

The figure 35 displays the performance of the approaches overall application fields in each metric for the standard performance ranking.

[237] Cf. Brownlee, 2017.
[238] Cf. Brownlee, 2018b.

Figure 35 Standard performance ranking per metric- Overall application fields

Method	Rank_MAPE
ELBS_05_2020	4.59
LightGBM	4.86
XGBoost	5.37
Stochastic Gradient Boosting	5.97
Extra Trees	7.13
Bagged Tree	7.53
Random Forest	7.71
BASELINE (MEAN-EST)	8.19
SVM	9.16
Lasso Regression	9.49
ElasticNet	9.53
ADA Boost	10.05
KNN	10.35
Ridge Regression	10.88
CART	11.65
Logistic Regression	12.17
Linear Regression	14.21

Method	Rank_MSPE
ELBS_05_2020	4.70
LightGBM	4.92
XGBoost	5.19
Stochastic Gradient Boosting	5.51
BASELINE (MEAN-EST)	7.14
Bagged Tree	8.14
Extra Trees	8.22
Random Forest	8.28
Lasso Regression	8.41
ElasticNet	8.48
ADA Boost	9.02
SVM	9.87
Logistic Regression	10.08
KNN	10.47
Ridge Regression	11.99
CART	12.72
Linear Regression	13.92

Method	Rank_CORR
LightGBM	3.72
XGBoost	4.78
Stochastic Gradient Boosting	4.85
ELBS_05_2020	6.21
Bagged Tree	6.28
Random Forest	6.81
Extra Trees	6.93
ADA Boost	8.76
Logistic Regression	9.00
CART	9.16
Ridge Regression	9.53
SVM	10.09
KNN	10.28
Linear Regression	10.84
ElasticNet	12.77
BASELINE (MEAN-EST)	13.03
Lasso Regression	13.07

Method	Rank_AUC
LightGBM	3.33
XGBoost	3.67
ADA Boost	4.11
Stochastic Gradient Boosting	4.22
ELBS_05_2020	6.22
Bagged Tree	7.00
Random Forest	7.22
Linear Regression	8.11
Extra Trees	9.00
Logistic Regression	9.11
CART	11.56
Ridge Regression	11.67
ElasticNet	12.00
Lasso Regression	12.00
KNN	13.33
SVM	14.11
BASELINE (MEAN-EST)	16.33

Source: Own depiction based on Appendix 13.

The figure 35 above shows that the MAPE and MSPE rankings are led by the ELBS-Tool, followed by the GBM models in the order: LightGBM, XG-Boost, stochastic gradient boosting. This reveals again the success of the evolution of the gradient boosting technique. The top positions in the CORR metric in the half right table are occupied by the GBM models whereas the ELBS-Tool is ranked at fourth position. The LightGBM and XG-Boost also lead the ranking for the AUC criterion, and therefore the classification problems. On third place the ADABoost approach is ranked and the stochastic gradient boosting approach and the ELBS-Tool only reach the fourth and fifth place. This indicates that the boosting approaches are good advice in classification problems.

The graphic below depicts the combination of all metrics overall application fields. This comprehensive overview is depicted by summing up the results of each approach for each metric and dividing the result by the number of metrics (4). This overview is shown below in figure 36 and represents the performance in the standard performance ranking overall 88 different target variables, not only one specific application fields.

Figure 36 Standard performance ranking overall metrics- Overall application fields

Source: Own depiction based on Appendix 13.

The visualization in figure 36 enhances the results for each individual metric in figure 35. The top positions are taken by the LightGBM, XG-Boost and stochastic gradient boosting approaches. They are followed by the ELBS-Tool on the fourth position. The results in the top group do not differ much, whereas the performance after the ELBS-Tool significantly decreases. After the ELBS-Tool, the tree-based bagging approaches bagged tree, random forest and extra tree are placed from position five to seven, followed by the AdaBoost method, the “White Box” approaches and the support vector machine.

6.2 T-Dominance ranking- Overall application fields

The figure 37 below gives an overview of the T-Dominance ranking of the different approaches overall application fields for each metric.

Figure 37 T-Dominance ranking per metric- Overall application fields

Method	NoBeDom_MAPE	Method	NoBeDom_MSPE	Method	NoBeDom_CORR
ELBS_05_2020	2.62	ELBS_05_2020	2.84	LightGBM	2.24
LightGBM	3.45	LightGBM	3.37	XGBoost	2.97
XGBoost	3.64	XGBoost	3.48	ELBS_05_2020	3.19
Stochastic Gradient Boosting	4.18	Stochastic Gradient Boosting	3.84	Stochastic Gradient Boosting	3.21
Extra Trees	5.51	Lasso Regression	5.90	Bagged Tree	4.39
Bagged Tree	5.94	BASELINE (MEAN-EST)	5.91	Extra Trees	4.94
Random Forest	5.99	ElasticNet	6.17	Random Forest	4.98
BASELINE (MEAN-EST)	6.67	Bagged Tree	6.39	ADA Boost	6.49
Lasso Regression	6.83	Extra Trees	6.70	Ridge Regression	6.86
ElasticNet	7.22	Random Forest	6.70	CART	7.20
SVM	7.23	ADA Boost	6.94	SVM	7.46
ADA Boost	8.10	SVM	7.86	Logistic Regression	7.50
KNN	8.30	Logistic Regression	8.67	Linear Regression	7.87
Ridge Regression	9.34	KNN	8.71	KNN	8.47
CART	9.97	Ridge Regression	10.48	Lasso Regression	9.45
Logistic Regression	11.00	CART	11.38	ElasticNet	9.55
Linear Regression	12.99	Linear Regression	12.60	BASELINE (MEAN-EST)	10.43

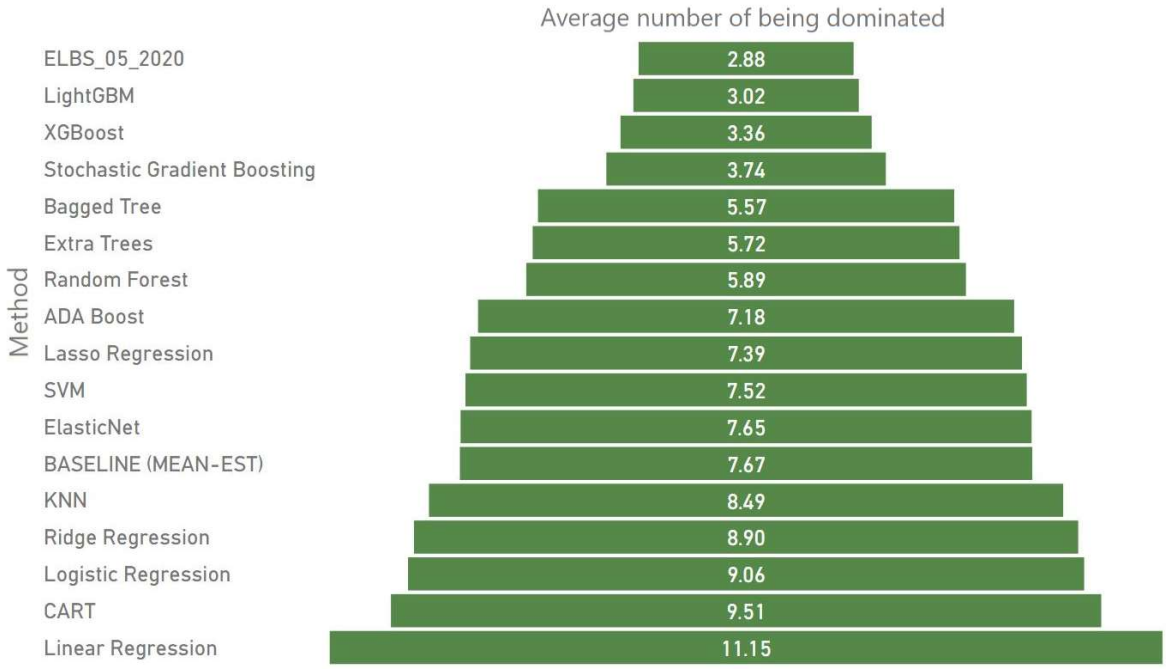
Source: Own depiction based on Appendix 13.

In the following the scores of each approach for each metric are displayed and further analyzed. As shown in the tables of figure 35 and 37, the best approaches to minimize the mean absolute and mean squared prediction error are the following: ELBS-Tool, LightGBM, XG-Boost, and stochastic gradient boosting. In these metrics they are usually followed by the decision tree-based approaches which use bagging: bagged tree, random forest, extra tree. Only for the MSPE criterion, the baseline estimator, lasso regression and elastic net approach rank better. However, the LightGBM approach was least dominated in the CORR metric, followed by XG-Boost, the ELBS-Tool and stochastic gradient boosting on positions two to four. In the following positions, the decision tree-based bagging approaches are placed again. The results confirm the assumption that the best approaches are grouped into the ones using the ELBS technique and the “Black Box” approaches using boosting with the gradient descent. It can also be observed that the evolution of the gradient boosting reached its goal with the LightGBM as best approach out of this class.

These approaches are usually followed by the decision tree-based bagging approaches, bagged tree, random forest and extra tree. However, in some particular application fields like time series panel data, simple estimators like the expanding mean (baseline-estimator) can achieve good results as well.

This illustrates that performance is also a matter of the given criterion and reveals that further developed “White Box” approaches can also achieve acceptable results. The “Black Boxes” would probably score even better with hyperparameter tuning.

Figure 38 T-Dominance ranking overall metrics- Overall application fields



Source: Own depiction based on Appendix 13.

As shown in figure 38, the methods that are least dominated, overall application fields and metrics, are the ELBS-Tool and the gradient boosting approaches LightGBM, XG-Boost, stochastic gradient boosting. The results do not differ much, and it can be concluded that all of them deliver good results overall different application fields. Following in the midfield are the remaining decision tree-based ensemble methods. While they still are “Black Box”-approaches, their average rankings are significantly worse than the ones of the top group. However, the results are still better than the rankings of the “White Boxes” that are following. Most likely due to missing hyperparameter tuning, the “Black Box” approaches SVM and AdaBoost cannot achieve good results and are ranked among the “White Box” approaches.

If one looks at the T-Dominance ranking overall application field in figure 38 it reveals that the “White Box” approaches cannot keep up with the results of the “Black Box” approaches.

As the comprehensive visualizations in figure 36 and 38 reveal, the consortium of “Black Box”-approaches achieves significantly better scores than the “White Box” approaches. Thus, for a better comprehensibility with the “White Box” approaches, one must put up with a drop in the performance results and vice versa. The AdaBoost approach marks the boundary between the better results of the “Black Boxes” and the worse results of the “White Boxes”.

The rankings also confirm the hypothesis that the most common used approaches, linear and logistic regression deliver the worst prediction performance. The ELBS-Tool is present in the top group of the average rankings. More important the ELBS-Tool is the approach that is least dominated overall application fields. The T-Dominance ranking answers the initial question best since it displays the superiority in a multitude of different calculation lags.

In addition to the good performance results, it has the mentioned advantages of a better understanding, empirical verifiability and many more as explained in figure 21 and in chapter 4.8. Using the ELBS-Tool can be described as a solution to the tradeoff between comprehensibility and performance. It achieves at least comparable results, often even better, to the gradient boosting models. The overall results for clearly reveal the difference between the “Black Box” and “White Box” approaches. Nevertheless, the “Black Box” approaches own the disadvantages of a bad comprehensibility while the ELBS-Tool combines a good performance with clear comprehensibility and verifiability. The results can differ for other underlying data. However, the performance of these approaches can be improved through hyperparameter tuning. The weak performance of the “White Box” approaches, especially the most popular linear and logistic regression approaches, for complex tasks like the panel time series data can be clearly observed. It is presented very clearly that the “White Box” approaches like the linear and logistic regression, kNN, CART and the baseline estimator occupy the last places.

7 Conclusion and critical appraisal

The predefined performance metrics MSPE, MAPE, CORR and AUC are used to evaluate the test data performance of all approaches over the three different application fields: cross-sectional data, time series data and time series panel data. The application fields consist of a multitude of various datasets and 88 different target variables. The results were analyzed for each individual application field and in a comprehensive overview overall application fields.

However, the time series data only contained regression problems and the panel time series data is only built from macro-economic datasets, but with a multitude of different target variables.

Even though the results sometimes vary for the different metrics, a coherent overall picture emerges across all metrics and all datasets, as well as for the individual fields of application.

The results of the research in chapters 5 and 6 allow a clear answer to the central question in this bachelor thesis: “Is there a machine learning approach that always leads to better results?”. No, there is not an approach that always leads to superior results with respect to the defined metrics and overall application fields. This bachelor thesis seems to support the widespread thesis that no approach is the best for all fields of application.[239]

This is mainly for three reasons: first of all, the renowned and presented methods have their different strengths and weaknesses that make them sometimes a good predictor for one application field, whereas they deliver weak results for another. Secondly, no hyperparameter tuning was carried out. If this is applied on all methods, for some approaches like AdaBoost and the SVM the results could significantly increase while the performance of others may remain almost unchanged. And in the end, the choice of the defined performance metrics is a significant factor in a performance comparison. Although all of them are commonly used metrics in the data science community, there is a multitude of possible choices. Since the nature of how they are calculated is different, not every metric can be applied on every problem. Therefore, the results can differ.

[239] Cf. Kaggle, 2019.

However, the previous discrepancy of the performance between the “Black Box” and “White Box” approaches can be confirmed. In nearly every case, the “Black Box” approaches enable better scores and are less often dominated than the “White Box” approaches. An exception is made by the AdaBoost and the SVM approach. They tend to deliver weak results, most likely due to missing hyperparameter tuning although they belong to the class of “Black Box” approaches. New approaches are constantly being developed or old ones are updated as can be seen for the GBM-approaches. The results confirm the success of the constant evolution since the LightGBM achieves the best results overall application fields out of the GBM-approaches.

In general, the gradient boosting models and the ELBS-Tool outperform the other approaches with respect to the defined metrics. In some cases, other methods are able to achieve good results as well. Examples are the other decision tree-based ensemble methods like random forest, extra tree and bagged tree in the time series data. They are also the group with the best rankings after the top group of GBM-models and the ELBS-Tool.

A special observation could be made for the time series panel data. The baseline estimator is able to achieve good results, especially for the T-Dominance ranking. This shows the difficulty of predicting complex data and that sometimes, predicting with simple rules can be helpful as well. A data scientist could save a lot of time with implementing complex models and instead carry out predictions with the baseline estimator. Nevertheless, it must be pointed out, that for this category only macroeconomic datasets are used, but with a multitude of target variables which could affect the predictions.

However, the ELBS-Tool delivers overall the best results. Although for some criteria and especially the simpler cross-sectional data, other approaches may achieve better results. For the cross-sectional data, the gradient boosting approaches showed their strengths and the LightGBM approach is the one with the best ranking and it was least dominated. While it is not easy to reconcile how the results of the gradient boosting models emerge, it is well comprehensible and verifiable how the predictions in the ELBS-Tool are created.

Furthermore, the ELBS-Tool is the approach that is least dominated in the more complex application fields of time series and panel time series data. Since the T-Dominance ranking considers a multitude of different calculation scenarios, it is the more constant criterion to compare superiority. It can clearly be stated that the ELBS- Tool's predictions are mostly superior, but at least comparable with the GBM group overall three application fields. Another reason why it can be considered as the best approach are the many advantages mentioned in chapter 4.8. Moreover, it shows a way of combining comprehensible methods with empirical verifiability and highly accurate predictions.

Since a multitude of different datasets were used and the approaches were applied on the most commonly used fields in data science the comparison gives a good performance overview between the different approaches. However, more datasets, extended preprocessing or an extensive hyperparameter tuning could affect the results. Furthermore, the popular and successful methods of deep learning and neural networks are not included due to the lack of standard hyperparameter. However, this way of predicting is considered even more complex and less interpretable and explainable as well. These limitations offer the possibility of further research in the future based on the results in this thesis.

Due to the stochastic nature of data and algorithms, any prediction model makes errors and cannot predict the data perfectly. Therefore, in the machine learning community a kind of natural limit where the accuracy cannot be increased is assumed.[240] Nevertheless, the good results of the ensemble GBM methods and the ELBS tool show that this limit does not seem to be far away anymore.

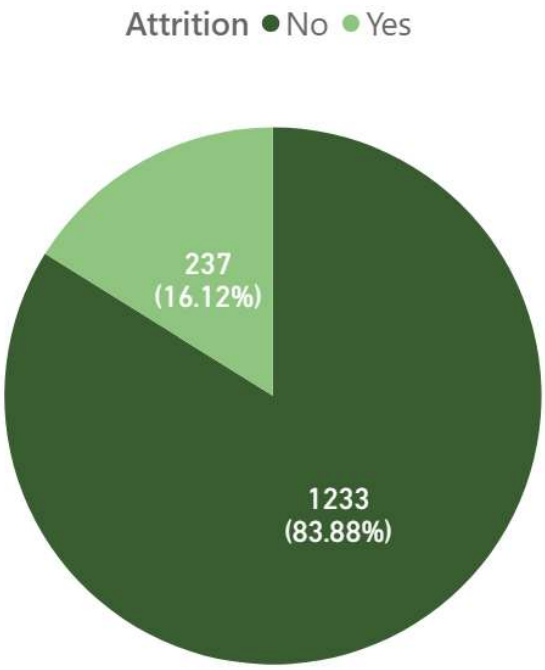
[240] Cf. Brownlee, 2018c.

List of appendices

The results of the comparison in the bachelor thesis are, unless otherwise explained, taken from the python notebooks mentioned above. These notebooks are included in the digital appendix on the enclosed USB-stick. I was supported in the programming by the team of the center for emergent law-based statistics at the cooperative state university Baden-Württemberg, Villingen-Schwenningen.

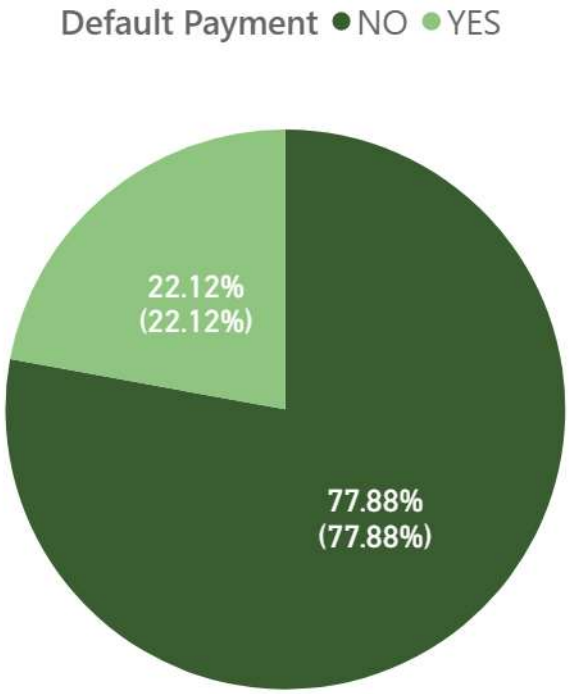
Appendices

Appendix 1 IBM dataset



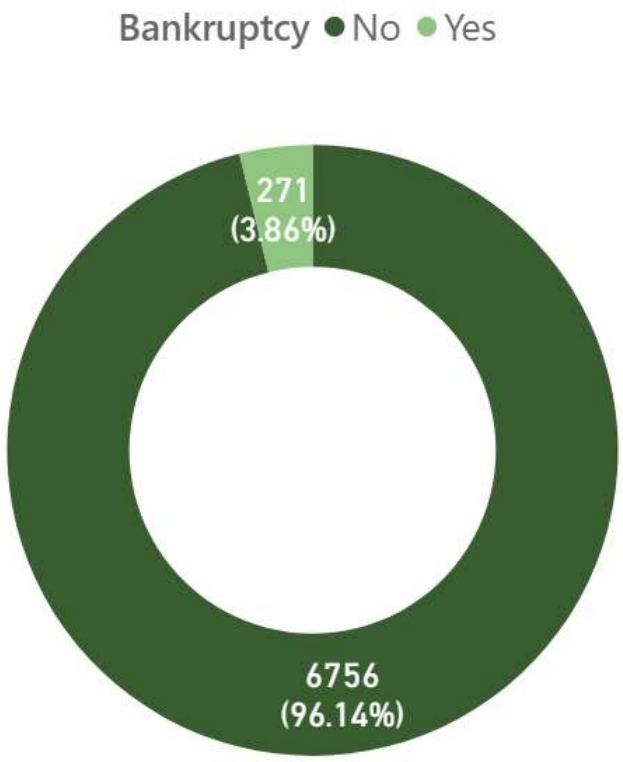
Source: Own depiction based on IBM, 2017.

Appendix 2 Taiwan credit card default



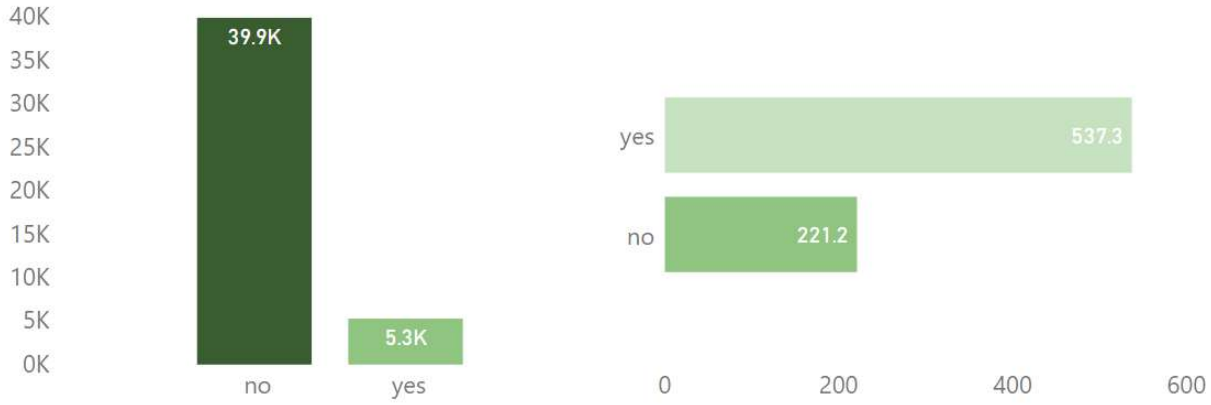
Source: Own depiction based on Yeh, 2016.

Appendix 3 Polish companies' bankruptcies



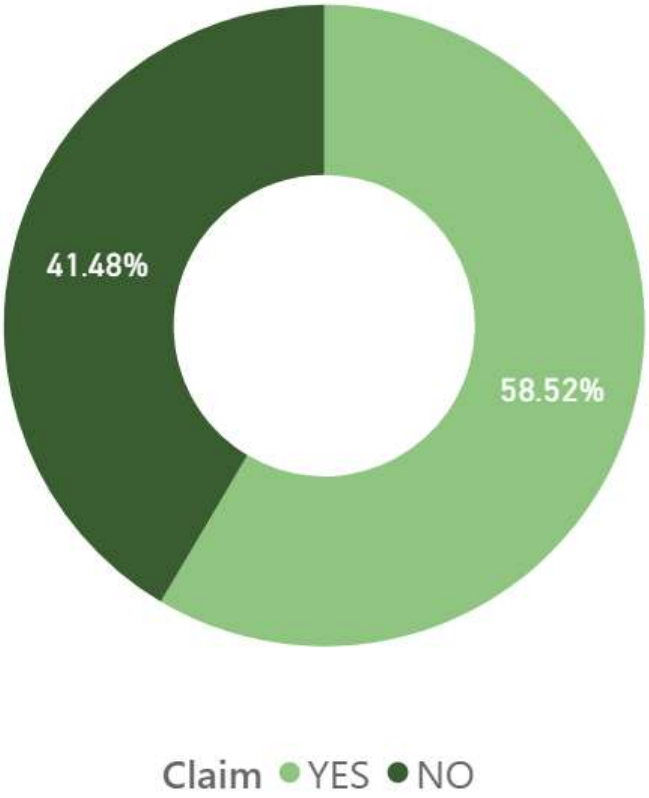
Source: Own depiction based on Tomczak, 2016.

Appendix 4 Bank marketing- Subscription quantity and average call duration



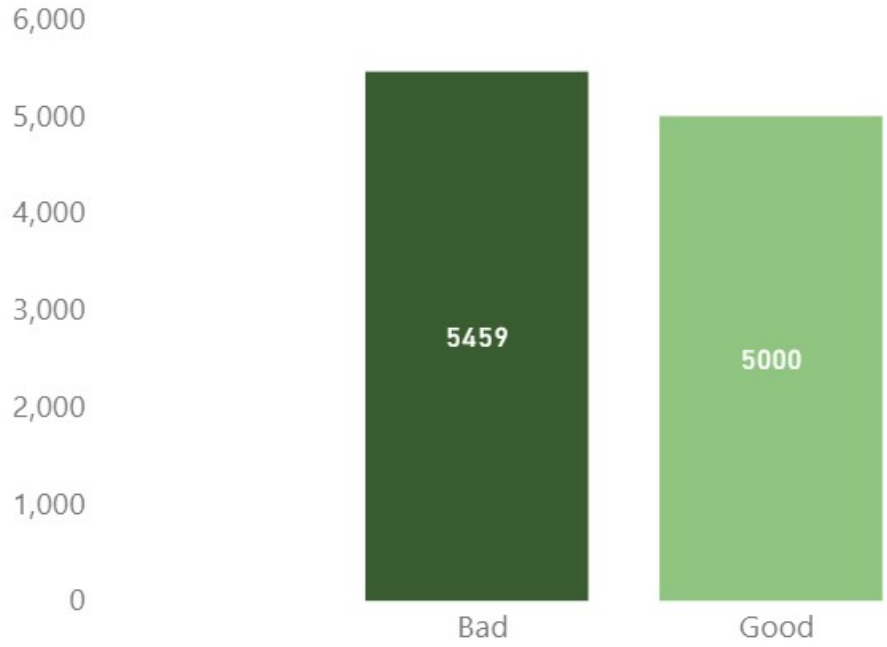
Source: Own depiction based on Moro et al., 2012.

Appendix 5 Health insurance claims



Source: Own depiction based on Kaggle, 2018.

Appendix 6 FICO- Risk performance



Source: Own depiction based on FICO, 2019.

Appendix 7 Allstate insurance claims severity

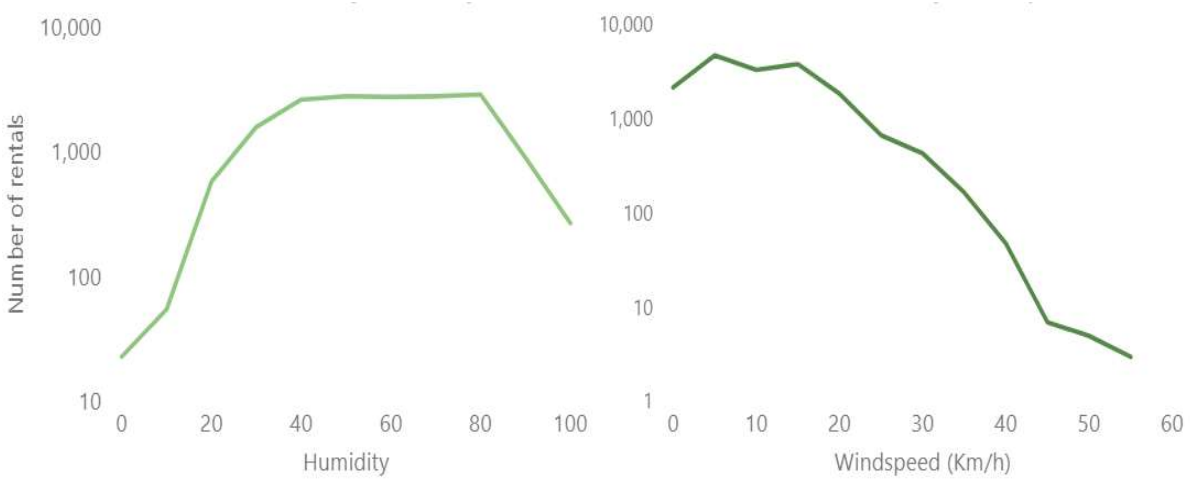
Allstate Claims			
Count	Mean	Maximum	Minimum
188.318	$2.465,33	$106.863,00	$6,00

Source: Own depiction based on Allstate, 2016.

Appendix 8 Bike dataset- Casual vs regular users and rentals by temperature

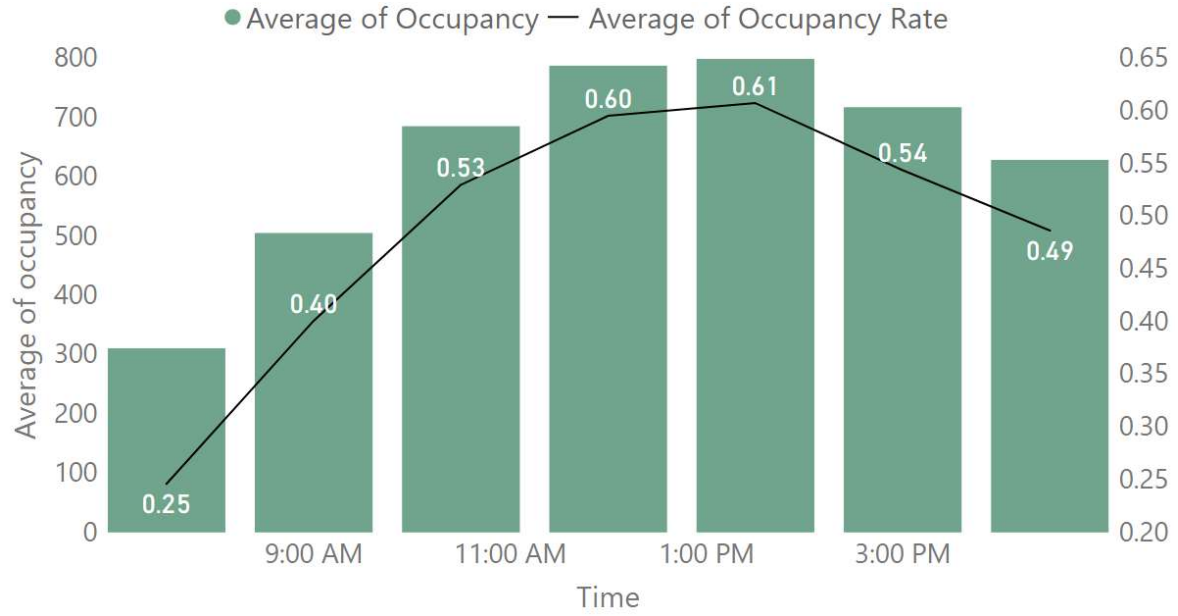
Source: Own depiction based on Fanaee/Gama, 2013.

Appendix 9 Bike dataset- Number of rentals by humidity and windspeed



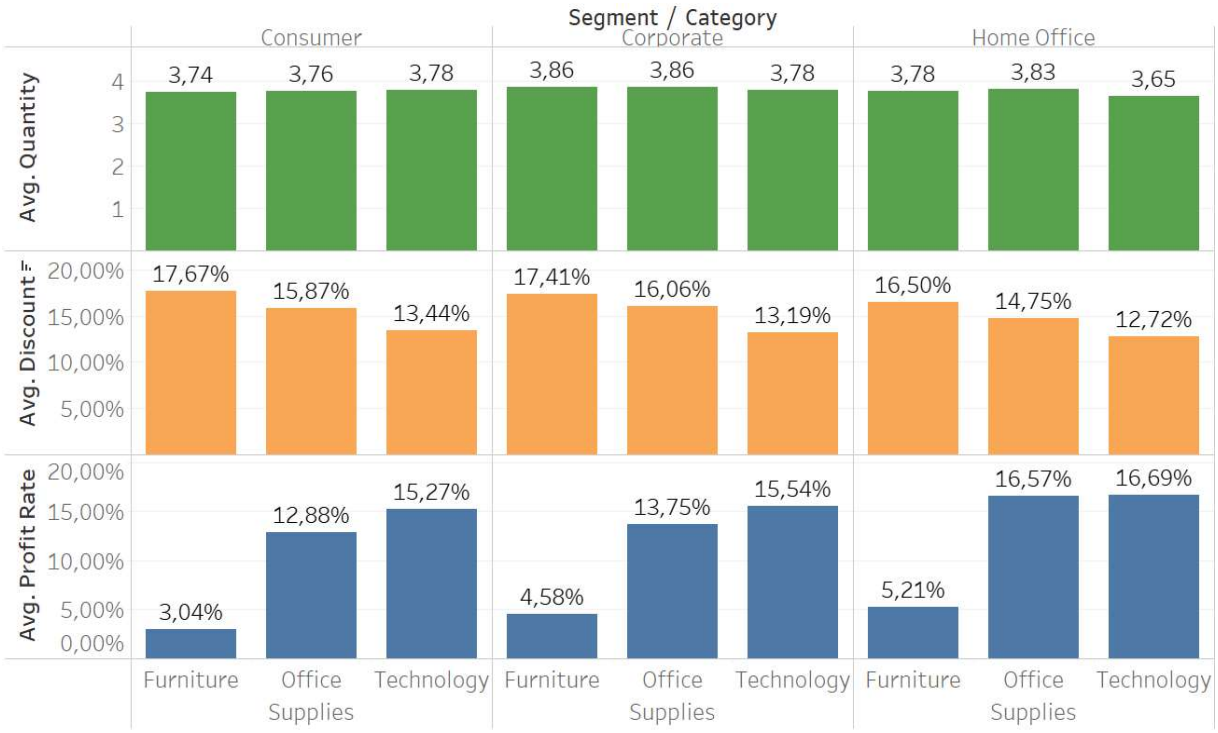
Source: Own depiction based on Fanaee/Gama, 2013.

Appendix 10 Car park- Hourly average of occupancy and occupancy rate



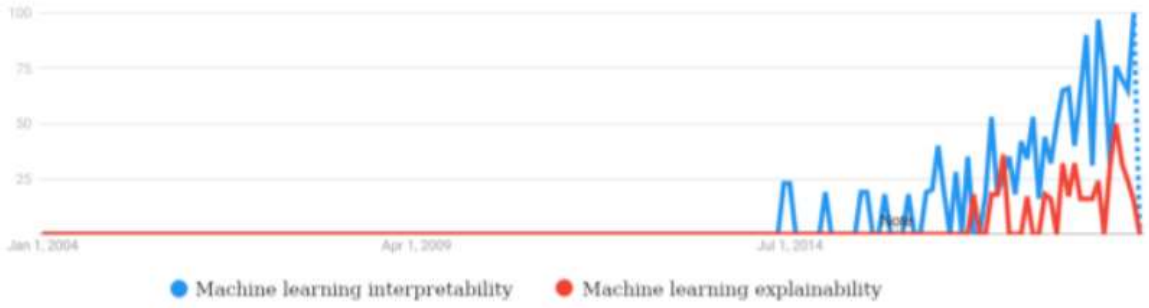
Source: Own depiction based on Stolfi, 2016.

Appendix 11 Superstore- Quantity, discount and profit rate



Source: Own depiction based on Tableau, 2018.

Appendix 12 Google trends for interpretability and explainability



Source: Carvalho/Pereira/Cardoso, 2019, p. 7.

Bibliography

Aberham, Jana Kuruc, Fabrizio	(2019) Support Vector machine, in: Kersting, Kristian, Lampert, Christoph, Rothkopf, Constantin, Publisher, Wie Maschinen lernen, Wiesbaden 2019, p. 95 – 105.
Acharya, Tarun	(2019) Regression with Regularization Technique, in: https://towardsdatascience.com/regression-with-regularization-techniques-7bbc1a26d9ba, June 6, 2019, Access May 9, 2020.
Akerkar, Rajendra	(2019) Artificial Intelligence for Business, Cham 2019.
Allstate	(2016) Allstate Claims Severity, in: Kaggle-.com/c/allstate-claims-severity/data, October 10, 2016, Access February 27, 2020.
Bank of America	(2020) What is a home equity line of credit, in: https://www.bankofamerica.com/mortgage/learn/what-is-a-home-equity-line-of-credit/, 2020, Access February 29, 2020.
Bhattacharjee, Joydeep	(2020) Practical machine learning with Rust, New York 2020.
Bhattacharyya, Saptashwa	(2018) Ridge and Lasso Regression: L1 and L2 Regularization, Complete Guide Using Scikit-Learn, in: https://towardsdatascienc-e.com-/ridge-and-lasso-regression-a-com-pleteguide-with-python-scikit-learn-e20e3-4bcbf0b, September 26, 2018, Access April 6, 2020.
Bakharia, Aneesha	(2016) SVM Parameter Tuning in Scikit Learn using GridSearchCV, in: https://medium.com-/@aneesha/svm-parameter-tuning-in-scikit-learn-using-gridsearchcv-2413c02125a0, January 18, 2016, Access March 24, 2020.

Burkov, Andriy	(2019) The Hundred Page Machine Learning book, Bonn 2019.
Buxmann,Peter Schmidt, Holger	(2019) Künstliche Intelligenz- Mit Algorithmen zum wirtschaftlichen Erfolg, Wiesbaden 2019.
Brownlee, Jason	(2020) Gradient Boosting with Scikit-Learn, XG-Boost, LightGBM, and CatBoost, in: https-://machinelearningmastery.com/gradient-boosting-with-scikit-learn-XG-Boost-lightgbm-and-catboost/, April 1, 2020, Access May 9, 2020.
Brownlee, Jason	(2019) Logistic Regression for machine learning, in: https://machinelearningmastery.-com/logistic-regression-for-Machine learning/, August 12, 2019, Access April 5, 2020.
Brownlee, Jason	(2018a) A gentle introduction to k-fold cross-validation, in: https://machinelearningmastery.com/k-fold-cross-validation/, May 23, 2018, Access April 22, 2020.
Brownlee, Jason	(2018b) Deep Learning For Time Series Forecasting, Vermont 2018.
Brownlee, Jason	(2018c) How To Know if Your Machine Learning Model Has Good Performance, in: https-://machinelearningmastery.com/how-to-know-if-your-machine-learning-model-has-good-performance/, April 20, 2018, Access June 23, 2020.
Brownlee, Jason	(2017) Introduction To Time Series Forecasting with Python: How to Prepare Data and Develop Models to Predict the Future, Vermont 2017.
Brownlee, Jason	(2016) A Gentle Introduction to the Gradient Boosting Algorithm for machine learning, in: https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-Machine learning/, September 9, 2016, Access April 13, 2020.

Carvalho, Diogo Pereira, Eduardo Cardoso, Jaime	(2019) Machine learning Interpretability: A Survey on Methods and Metrics, in: https-://www.mdpi.com/2079-9292/8/8/832, July 26, 2019, Access May 8, 2020.
Choudhury, Ambika	(2019) Explainability vs Interpretability in Artificial Intelligence and machine learning, in: https://analyticsindiamag.com/explainability-vs-interpretability-in-artificial-intelligence-and-Machine learning/, January 14, 2019, Access May 8, 2020.
Clarke, Aaron Francis, Gregory Herzog, Michael	(2019) Understanding Statistics and Experimental Design- How not to lie with statistics, Cham 2019.
Desarda, Akash	(2019) Understanding AdaBoost, in: https-://towardsdatascience.com/understanding-adaboost-2f94f22d5bfe, January 17, 2019, Access April 10, 2020.
Döbel, Inga Leis, Miriam Vogelsang, Manuel et al.	(2018) Maschinelles Lernen- Kompetenzen, Anwendungen und Forschungsbedarf, in: https://www.bigdata.fraunhofer.de/content/dam/bigdata/de/documents/Publikationen/BMBF_Fraunhofer_ML-Ergebnisbericht_Gesamt.pdf, March 29, 2018, Access January 15, 2020.
Fanaee, Hadi Gama, Joao	(2013) Bike Sharing Dataset, in: https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset, December 20, 2013, Access March 6, 2020.
FICO	(2019) Explainable machine learning Challenge, in: https://community.fico.com/s/explainable-Machine learning-challenge, January 2019, Access February 29, 2020.
Freund, Yoav Schapire, Robert	(1995) A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting, in: Journal of Computer and System Sciences, August 1, 1997, p. 119 – 139.

Friedman, Jerome H.	(2002) Stochastic Gradient Boosting, in: https-://www.sciencedirect.com/science/article-/pii/S0167947301000652, 2002, Access April 14, 2020.
Gall, Richard	(2018) Machine learning Explainability vs Interpretability: Two concepts that could help restore trust in AI, in: https://www.kdnuggets.com/2018/12/Machine learning-explainability-interpretability-ai.html, 2018, Access May 8, 2020.
Gandhi, Preet	(2019) Explainable Artificial intelligence, in: https://www.kdnuggets.com/2019/01/explainable-ai.html, January 2019, Access May 8, 2020.
GeeksforGeeks	(2020) Stochastic Gradient Descent, in: https-://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/, 2020, Access April 13, 2020.
Ghoneim, Salma	(2020) 5 Types of bias & how to eliminate them in your machine learning project, in: https://towardsdatascience.com/5-types-of-bias-how-to-eliminate-them-in-your-Machine learning-project-75959af9d3a0, April 16, 2020, Access March 13, 2020.
Github	(2020) Gradient Boosting machines, in: http://uc-r.github.io/gbm_regression, 2020, April 15, 2020.
GoogleDevelopers	(2020a) Reducing Loss: Stochastic Gradient Descent, in: https://developers.google.com-/Machine learning/crash-course/reducing-loss/stochastic-gradient-descent, 2020, April 13, 2020.
GoogleDevelopers	(2020b) Classification ROC Curve and AUC, in: https://developers.google.com/Machine learning/crash-course/classification/roc-and-auc, February 10, 2020, Access March 19, 2020.

Grover, Prince	(2017) Gradient Boosting from Scratch, in: https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d, December 9, 2017, Access April 17, 2020.
Hackernoon	(2019) An Introduction to Ridge, Lasso, and Elastic Net Regression, in: https://hackernoon.com/an-introduction-to-ridge-lasso-and-elastic-net-regression-cca60b4b934f, February 20, 2019, Access April 6, 2020.
IBM	(2017) IBM HR Analytics Employee Attrition & Performance, in: https://www.kaggle.com/pavansubhasht/ibm-hr-analytics-attrition-dataset/metadata, March 31, 2017, Access February 27, 2020.
Jain, Shubham	(2017) A comprehensive beginners guide for Linear, Ridge and Lasso Regression in Python and R, in: https://www.analyicsvidhya-.com/blog/2017/06/a-comprehensive-guide-for-linear-ridge-and-lasso-regression/, June 22, 2017, Access April 6, 2020.
Jain, Aarshay	(2016) A complete Tutorial on Ridge and Lasso Regression in Python, in: https-://www.analyticsvidhya.com-/blog-/2016/01/ridge-lasso-regression-python-complete-tutorial/, January 28, 2016, Access May 9, 2020.
Jansen, Stefan	(2020) Depth-wise versus leaf-wise growth, in: https://www.oreilly.com/library/view/hands-on-Machine learning/9781789346411/52ffbff1-9242-4de6-9beb-e053a5dfb6f3.xhtml, 2020, Access April 21, 2020.
Jordá, Oscár Schularick, Moritz Taylor, Alan	(2017) The Rate of Return on Everything, 1870–2015, in: http://www.macrohistory.net/?-sermons=the-rate-of-return-on-everything-1870-2015, December 14, 2017, Access March 20, 2020.

Kaggle	(2019) Kaggle ML & DS Survey, in: https://www.kaggle.com/c/kaggle-survey-2019, November 2019, Access January 15, 2020.
Kaggle	(2018) Sample insurance claim Prediction Dataset, in: https://www.kaggle.com/easonlai/sample-insurance-claim-prediction-dataset, 2018, Access February 29, 2020.
Kalayci, Erdem	(2018) The trade-off in machine learning: Accuracy vs Explainability, in: https://medium.com/@erdemkalayci/the-tradeoff-in-Machine learning-accuracy-vs-explainability-fbb13914fde2, December 25, 2018, Access May 8, 2020.
Kenton, Will	(2019) Endogenous Variable, in: https://www.investopedia.com/terms/e/endogenous-variable.asp, August 22, 2019, Access May 17, 2020.
Khandani, Amir Lo, Andrew Merton, Robert	(2009) Systemic Risk and Refinancing Ratchet Effect, in: https://www.nber.org/papers/w15362, September 2009, Access February 29, 2020.
Khandelwal, Pranjal	(2017) Which algorithm takes the crown: Light GBM vs XG-Boost, in: https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-XG-Boost/, June 12, 2017, Access April 15, 2020.
Kim, Kyoosik	(2019) Ridge Regression for better usage, in: https://towardsdatascience.com/ridge-regression-for-better-usage-2f19b3a202db, January 3, 2019, Access April 6, 2020.
Kreutzer, Ralf Sirrenberg, Marie	(2019) Künstliche Intelligenz verstehen, Grundlagen, Use Cases, unternehmenseigene KI-Journey, Wiesbaden 2019.

Kuck, Andre — (2019a) Digitale Intelligenz, in: https-://www.udpl.info/wp-content/uploads/2019/11/-Vortrag_EUFH_Handout.html, October 21, 2019, Access May 12, 2020.

Kuck, André — (2019b) Anwendungen emergenzbasierter KI in Produktion und Betriebswirtschaft, in: https://www.udpl.info/wp-content/uploads/2019/11/DHBW_Vortrag_Zukunftsforum_Handout_ZES_Kuck.html, October 21, 2019, Access May 12, 2020.

Kuck, André — (2017) Emergenzbasierte Anlagealgorithmen, Automatisierte Geldanlage auf Basis empirischer Gesetze über die emergente relative Vorteilhaftigkeit von Anlageregeln, November 2017.

Kuck, André
Kuck, Elsbeth
Jan Philipp Harries — (2015) Der Weg zu wahren empirischen Gesetzen und rationalem Handeln, in: https://www.udpl.info/wp-con-tent/uploads/2017/03/Kuck_Harries_WahreGesetze.pdf, March 30, 2015, Access March 19, 2020.

Kuck, André
Frischhut, Hans — (2017) The process of emergent law-based model building, Villingen-Schwenningen 2017.

Kuck, André
Frischhut, Hans — (2015) Die Suche nach emergenten deterministischen empirischen Gesetzen als Alternative zur wahrscheinlichkeitsbasierten Statistik, in: https://www.udpl.info/wp-content/uploads/2017/03/Kuck_Frischut_.pdf, 2015, Access May 6, 2020.

Kuck, André
Kratz, Norbert
Frischhut, Hans — (2018) Objektives Wissen und dominante Entscheidungsregeln auf Basis emergenzbasierten Maschinenlernens, in: https://www.udpl-.info/wp-content/uploads/2018/03/Vortrag-_Fachtagung_DH_ZES-.html, February 1, 2018, Access May 6, 2020.

Kumar, Naresh (2019a) Advantages and Disadvantages of KNN Algorithm in machine learning, in: http://theprofessionalspoint.blogspot.com/-2019/02/advantages-and-disadvantages-of-knn.html, February 23, 2019, Access April 4, 2020.

Kumar, Naresh (2019b) Advantages and Disadvantages of Logistic Regression in machine learning, in: http://theprofessionalspoint.blogspot.com-/2019/03/advantages-and-disadvantages-of.html, March 2, 2019, Access April 5, 2020.

Kumar, Naresh (2019c) Advantages and Disadvantages of Random Forest Algorithm in machine learning, in: http://theprofessionalspoint.blogspot-.com/2019/02/advantages-and-disadvantages-of-random.html, February 23, 2019, Access April 10, 2020.

Kumar, Naresh (2019d) Advantages of XG-Boost Algorithm in machine learning, in: http://theprofessionalspoint.blogspot.com/2019/03/advantages-of-XG-Boost-algorithm-in.html, March 9, 2019, Access April 15, 2020.

Kurt, Noemi (2020) Stochastik für Informatiker- Eine Einführung in einheitlich strukturierten Lerneinheiten, Berlin 2020.

Maklin, Cory (2018) Machine learning Algorithms Part 11: Ridge Regression, Lasso Regression and Elastic-Net Regression, in: https://medium.com/@corymaklin/Machine learning-algorithms-part-11-ridge-regression-7d5-861c2bc76, December 30, 2018, Access April 5, 2020.

Mandot, Pushkar (2017) What is LightGBM, how to implement it? How to fine tune the parameters? in: https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc, August 17, 2017, Access April 15, 2020.

Manure, Avinash
Singh, Pramod
(2020) LearnTensorFlow 2.0- Implement machine learning and Deep Learning Models with Python, New York 2020.

McDonald, Conor
(2017) Machine learning fundamentals (I), Cost functions and gradient descent, in: https://towardsdatascience.com/Machine learning-fundamentals-via-linear-regression-41a5d11f5220, November 27, 2017, Access April 22, 2020.

Minaee, Shervin
(2019) 20 popular Machine Learning Metrics, in: https://towardsdatascience.com/20-popular-machine-learning-metrics-part-1-classification-regression-evaluation-metrics-1ca3e282a2ce, October 28, 2019, Access June 24, 2020.

Mishra, Aditya
(2018) Metrics to Evaluate your machine learning Algorithm, in: https://towardsdatascience.com/metrics-to-evaluate-your-Machine learning-algorithm-f10ba6e38234, February 24, 2018, Access March 17, 2020.

Morde, Vishal
(2019) XGBoost Algorithm: Long May She Reign, in: https://towardsdatascience-.com-/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d, April 8, 2019, Access June 23, 2020.

Moro, Sergio et al.
(2012) Bank Marketing Dataset, in: https://archive.ics.uci.edu/ml/datasets/-Bank+Marketing, February 14, 2012, Access March 18, 2020.

Nagpal, Anuja
(2017) L1 and L2 Regularization Methods, in: https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c, October 13, 2017, Access April 22, 2020.

Narkhede, Sarang
(2018) Understanding AUC – ROC Curve, in: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5, January 26, 2018, Access March 19, 2020.

Navlani, Avinash	(2018) AdaBoost Classifier in Python, in: https://www.datacamp.com/community/tutorials/adaboost-classifier-python, November 20, 2018, Access April 11, 2020.
Neumann, Michael	(2019) k-Nächste Nachbarn in: Kersting, Kristian, Lampert, Christoph, Rothkopf, Constantin, Publisher, Wie Maschinen lernen, Wiesbaden 2019, p. 73 – 80.
Open Data Science (ODSC)	(2018) Gradient Boosting and XG-Boost, in: https://medium.com/@ODSC/gradient-boosting-and-XG-Boost-9b4a23b84944, October 1, 2018, Access April 14, 2020.
Oleszak, Michael	(2019) Regularization: Ridge, Lasso and Elastic Net, in: https://www.datacamp.com/community/tutorials/tutorial-ridge-lasso-elastic-net, November 12, 2019, Access April 7, 2020.
Olson, David Desheng, Wu	(2020) Predictive Data Mining Models- Computational Risk Management, Singapore 2020.
Paper, David	(2020) Hands-on Scikit-Learn for machine learning Applications- Data Science Fundamentals with Python, New York 2020.
Prabhakaran, Selva	(2020) Logistic Regression – A Complete Tutorial with Examples in R, in: https://www.machinelearningplus.com/Machine learning/logistic-regression-tutorial-examples-r/, 2020, Access May 9, 2020.
Prajapati, Kaushal	(2019) Decision Tree, in: https://medium.com/Machine learning-for-grandma/decision-tree-e6ab1037df16, August 24, 2019, Access April 9, 2020.
Rakhecha, Aditya	(2019) Importance of Loss Function in machine learning, in: https://towardsdatascience.com/importance-of-loss-function-in-Machine learning-eddaaec69519, September 12, 2019, Access April 22, 2020.

Ravindran, Prashanth	(2015) What are the differences between bagged trees and random forests, in: https://www.quora.com/What-are-the-differences-between-bagged-trees-and-random-forests, December 16, 2015, Access April 11, 2020.
Reuters	(2020) DAX Index, in: https://www.reuters.com/quote/.GDAXI, 2020, Access May 17, 2020.
Rottmann, Horst	(2020a) Zeitreihendaten, in: https://wirtschaftslexikon.gabler.de/definition/zeitreihendaten-51137, 2020, Access February 13, 2020.
Rottmann, Horst	(2020b) Paneldaten, in: https://wirtschaftslexikon.gabler.de/definition/paneldaten-und-paneldatenmodelle-52094, 2020, Access February 13, 2020.
Sberbank	(2016) Sberbank Russian Housing Market, in: https://www.kaggle.com/c/sberbank-russian-housing-market/data, 2016, Access May 3, 2020.
Scikit-Learn	(2020a) Nearest Neighbors, in: https://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbors-classification, 2020, Access April 4, 2020.
Scikit-Learn	(2020b) Support Vector machines, in: https://scikit-learn.org/stable/modules/svm.html#complexity, 2020, Access March 30, 2020.
Scikit-Learn	(2020c) AdaBoost Classifier, in: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html, 2020, Access April 4, 2020.
Scikit-Learn	(2020d) Decision Trees, in: https://scikit-learn.org/stable/modules/tree.html, 2020, Access April 11, 2020.

Scikit-Learn	(2020e) Ensemble methods, in: https://scikit-learn.org/stable/modules/ensemble.htm, 2020, Access April 11, 2020.
Singhal, Ashish	(2018) Machine learning: Ridge Regression in Detail, in: https://towardsdatascience.com/Machine learning-ridge-regression-in-detail-76787a2f8e2d, November 9, 2018, Access April 5, 2020.
Statista	(2020) Statistik-Lexikon: Definition Querschnittsdaten, in: https://de.statista.com/statistik/lexikon/definition/107/querschnittsdaten/, Access February 13, 2020.
Stolfi, Daniel	(2016) Parking Birmingham Dataset, in: https://archive.ics.uci.edu/ml/datasets/Parking+Birmingham, 2016, Access March 21, 2020.
Swalin, Alvira	(2018) CatBoost vs. LightGBM vs. XG-Boost, in: https://towardsdatascience.com/catboost-vs-light-gbm-vs-XG-Boost-5f93620723-db?gi=4c7109c8f7af, March 13, 2018, Access April 22, 2020.
Swaminathan, Saishruthi	(2018) Logistic Regression- Detailed Overview, in: https://towardsdatascience.com/logistic-regression-detailed-overview-46c-4da4303bc, March 15, 2018, Access April 5, 2020.
Tableau	(2018) Sample Superstore, in: https://-www.tableau.com/solutions/gallery/super-sample-superstore, 2018, Access March 20, 2020.
Tomczak, Sebastian	(2016) Polish companies bankruptcy data, in: https://archive.ics.uci.edu/ml/datasets/Polish-+companies+bankruptcy+data, April 11, 2016, Access February 28, 2020.

Tseng, Gabriel	(2018) Gradient Boosting and XG-Boost, in: https://medium.com/@gabrieltseng/gradient-boosting-and-XG-Boost-c306c1bcfaf5, April 13, 2018, Access April 14, 2020.
UDPL	(2020) UDPL.info, in: https://www.udpl.info/, 2020, Access May 12, 2020.
Vermeulen, Andreas	(2020) Industrial machine learning, New York 2020.
Yeh, I-Cheng	(2016) Default of credit card client's data, in: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients, January 26, 2016, Access February 28, 2020.
Zhou, Victor	(2019) A Simple Explanation of Gini Impurity, in: https://victorzhou.com/blog/gini-impurity/, March 29, 2019, Access April 9, 2020.

Declaration

I hereby certify that I wrote my bachelor's thesis on the subject:

Comparison of selected machine learning approaches-
An analysis based on different economic application fields

independently and have not used any sources or resources other than those specified. I also assure that the submitted electronic version matches the printed version.

Munich
Location

July 4, 2020 
Date Signature