Julius-Maximilians-Universität Würzburg
Wirtschaftswissenschaftliche Fakultät

Advanced Analytics in Operations Management and Information Systems: Methods and Applications

Inauguraldissertation
zur Erlangung des akademischen Grades
doctor rerum politicarum (Dr. rer. pol.)

vorgelegt von
Nikolai Werner Stein, M.Sc.
geboren in Würzburg

Name und Anschrift:	Nikolai Werner Stein Gertraud-Rostosky-Str. 54 97082 Würzburg
Erstgutachter:	Prof. Dr. Christoph M. Flath
Zweitgutachter:	Prof. Dr. Richard Pibernik
Datum der Einreichung:	14. August 2019

Abstract

The digital transformation of business and society presents enormous potentials for companies across all sectors. Fueled by massive advances in data generation, computing power, and connectivity, modern organizations have access to gigantic amounts of data. Companies seek to establish data-driven decision cultures to leverage competitive advantages in terms of efficiency and effectiveness. While most companies focus on descriptive tools such as reporting, dashboards, and advanced visualization, only a small fraction already leverages advanced analytics (i.e., predictive and prescriptive analytics) to foster data-driven decision-making today. Therefore, this thesis set out to investigate potential opportunities to leverage prescriptive analytics in four different independent parts.

As predictive models are an essential prerequisite for prescriptive analytics, the first two parts of this work focus on predictive analytics. Building on state-of-the-art machine learning techniques, we showcase the development of a predictive model in the context of capacity planning and staffing at an IT consulting company. Subsequently, we focus on predictive analytics applications in the manufacturing sector. More specifically, we present a data science toolbox providing guidelines and best practices for modeling, feature engineering, and model interpretation to manufacturing decision-makers. We showcase the application of this toolbox on a large data-set from a German manufacturing company.

Merely using the improved forecasts provided by powerful predictive models enables decision-makers to generate additional business value in some situations. However, many complex tasks require elaborate operational planning procedures. Here, transforming additional information into valuable actions requires new planning algorithms. Therefore, the latter two parts of this thesis focus on prescriptive analytics. To this end, we analyze how prescriptive analytics can be utilized to determine policies for an optimal searcher path problem based on predictive models. While rapid advances in artificial intelligence research boost the predictive power of machine learning models, a model uncertainty remains in most settings. The last part of this work proposes a prescriptive approach that accounts for the fact that predictions are imperfect and that the arising uncertainty needs to be considered. More specifically, it presents a data-driven approach to sales-force scheduling. Based on a large data set, a model to predictive the benefit of additional sales effort is trained. Subsequently, the predictions, as well as the prediction quality, are embedded into the underlying team orienteering problem to determine optimized schedules.

Kurzzusammenfassung

Die digitale Transformation der Gesellschaft birgt enorme Potenziale für Unternehmen aus allen Sektoren. Diese verfügen aufgrund neuer Datenquellen, wachsender Rechenleistung und verbesserter Konnektivität über rasant steigende Datenmengen. Um im digitalen Wandel zu bestehen und Wettbewerbsvorteile in Bezug auf Effizienz und Effektivität heben zu können müssen Unternehmen die verfügbaren Daten nutzen und datengetriebene Entscheidungsprozesse etablieren. Dennoch verwendet die Mehrheit der Firmen lediglich Tools aus dem Bereich „descriptive analytics“ und nur ein kleiner Teil der Unternehmen macht bereits heute von den Möglichkeiten der „predictive analytics“ und „prescriptive analytics“ Gebrauch. Ziel dieser Dissertation, die aus vier inhaltlich abgeschlossenen Teilen besteht, ist es, Einsatzmöglichkeiten von „prescriptive analytics“ zu identifizieren.

Da prädiktive Modelle eine wesentliche Voraussetzung für „prescriptive analytics“ sind, thematisieren die ersten beiden Teile dieser Arbeit Verfahren aus dem Bereich „predictive analytics.“ Ausgehend von Verfahren des maschinellen Lernens wird zunächst die Entwicklung eines prädiktiven Modells am Beispiel der Kapazitäts- und Personalplanung bei einem IT-Beratungsunternehmen veranschaulicht. Im Anschluss wird eine Toolbox für Data Science Anwendungen entwickelt. Diese stellt Entscheidungsträgern Richtlinien und bewährte Verfahren für die Modellierung, das Feature Engineering und die Modellinterpretation zur Verfügung. Der Einsatz der Toolbox wird am Beispiel von Daten eines großen deutschen Industrieunternehmens veranschaulicht.

Verbesserten Prognosen, die von leistungsfähigen Vorhersagemodellen bereitgestellt werden, erlauben es Entscheidungsträgern in einigen Situationen bessere Entscheidungen zu treffen und auf diese Weise einen Mehrwert zu generieren. In vielen komplexen Entscheidungssituationen ist die Ableitungen von besseren Politiken aus zur Verfügung stehenden Prognosen jedoch oft nicht trivial und erfordert die Entwicklung neuer Planungsalgorithmen. Aus diesem Grund fokussieren sich die letzten beiden Teile dieser Arbeit auf Verfahren aus dem Bereich „prescriptive analytics“. Hierzu wird zunächst analysiert, wie die Vorhersagen prädiktiver Modelle in präskriptive Politiken zur Lösung eines „Optimal Searcher Path Problem“ übersetzt werden können. Trotz beeindruckender Fortschritte in der Forschung im Bereich künstlicher Intelligenz sind die Vorhersagen prädiktiver Modelle auch heute noch mit einer gewissen Unsicherheit behaftet. Der letzte Teil dieser Arbeit schlägt einen präskriptiven Ansatz vor, der diese Unsicherheit berücksichtigt. Insbesondere wird ein datengetriebenes Verfahren für die Einsatzplanung im Außendienst entwickelt. Dieser Ansatz integriert Vorhersagen bezüglich der Erfolgswahrscheinlichkeiten und die Modellqualität des entsprechenden Vorhersagemodells in ein „Team Orienteering Problem.“

Acknowledgements

First and foremost, I would like to thank my doctoral advisor, Prof. Dr. Christoph M. Flath for his excellent supervision and his continuous guidance and support. He supported my research ideas and was always at my side with help and advice. I benefited a lot from his creativity and deep expertise, as well as his counseling and persistence. I am grateful for his patience and effort during countless weekends and nights of paper development and for consistently pushing me to improve my work and make this research a success. I truly could not have imagined having a better advisor and mentor for my research.

I would also like to thank my co-advisor, Prof. Richard Pibernik, not only for his insightful comments and encouragement but also for the hard questions which incented me to widen my research from various perspectives. Due to his commitment and constructiveness, our collaboration turned out to be very fruitful. I am very grateful for his counsel and effort during the last years.

Furthermore, I want to thank all my former and current colleagues at the Chairs of Information Systems and Information Management, Logistics and Quantitative Methods in Business Administration, Information Systems Engineering, and Business Administration and Information Systems, for creating a productive research environment and a pleasant workplace. I would like to express special thanks to Matthias Griebel, Patrick Föll, and Florian Imgrund for being awesome office mates. Matthias Hauser and Marcus Fischer for preventing sprawling lunch breaks. Jannis Hanke for our great ECIS trip to Guimaraes. Toni Greif, Jan Meller, Fabian Taigel, Adrian Krenzer, Christina Laake, Felix Oberdorf, and Alexander Dürr for the collaboration on our research papers. Konstantin Kloos, Felix Lauton, Pascal Notz, Felix Schmidt, Peter Wolf, and Jana Niemeyer for the interesting discussions and the feedback in our OPIM seminars. Marco

Wirth for helping me navigate the bureaucracy. Gregor Friedrich-Baasner for keeping me company in the yard and Adrian Hofmann for letting me use “The Beast” to run my optimizations. In particular, I want to thank Claudia Mack for reminding me to take my remaining vacation and for making everybody feel welcome in the office.

Finally, I would like to express my deepest gratitude to my family for their endless and invaluable support over the last years. Without their backing, it would not have been possible for me to realize this dissertation project.

Contents

1 Introduction

The digital transformation of business and society presents enormous potentials for companies across all sectors. Fueled by massive advances in data generation, computing power, and connectivity, modern organizations have access to gigantic amounts of data. According to a recent study by Dhawan, Hei, and Laczkowski (2018), this development is just starting to gain traction. As pointed out by the authors, the industrial sector is expected to see more disruption within the next five years than in the past 20 years combined. Following the European Commission, leveraging the available data sources holds the key to unlocking future growth in Europe. However, EU businesses are currently not taking full advantages of these possibilities.[1]

Consequently, there is a rising demand for new analytics tools to foster data-driven decision-making and increase both efficiency and effectiveness of business processes (Chen, Chiang, and Storey 2012). This growing need has led to the emergence of a plethora of new trends and buzzwords, such as big data analytics (Russom 2011), advanced analytics (Barton and Court 2012), business analytics (Kohavi, Rothleder, and Simoudis 2002), or data science (Provost and Fawcett 2013). All of these terms are used to summarize the efforts to drive decisions based on the extensive use of data. In this thesis, these terms are used interchangeably as there is still no clear distinction between them.

In the era of big data, advanced analytics has emerged as an essential area of study for both practitioners and researchers from various fields (Davenport 2006). As pointed out by Mortenson, Doherty, and Robinson (2015), analytics is a cross-disciplinary field at the intersection of technology, decision-making, and quantitative methods. As such, analytics offers tremendous research opportunities for traditional cross-disciplinary fields

[1] https://ec.europa.eu/growth/industry/policy/digital-transformation_en

operating at these intersections, such as information systems, operations research, and artificial intelligence (Figure 1.1).

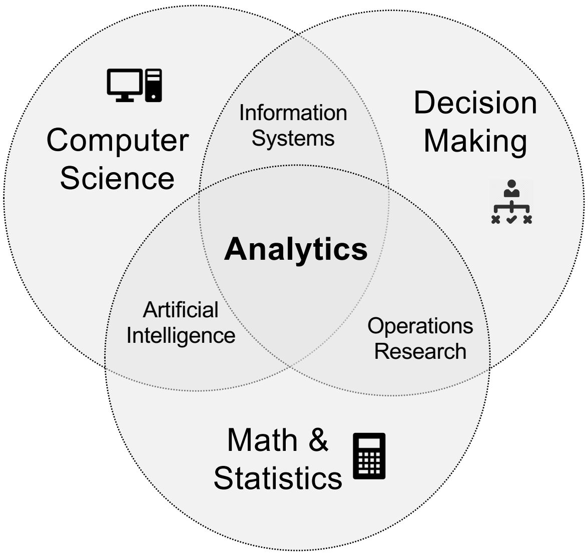


Figure 1.1: Advanced analytics as a cross-disciplinary field. This figure is inspired by Mortenson, Doherty, and Robinson 2015.

1.1 Operations Management & the Analytics Stack

Depending on the scope of the analysis, advanced analytics can be further refined into three categories (Lustig et al. 2010; Evans and Lindner 2012; Holsapple, Lee-Post, and Pakath 2014). So far, most business analytics applications used in practice aim at using data to understand the past and current business performance. To this end, data is summarized into meaningful charts and reports informing decision-makers. The term descriptive analytics summarizes these techniques. Going further, predictive analytics analyzes and extracts patterns from historical data. The uncovered relationships are extrapolated in time to make predictions on future business developments. Prescriptive analytics goes even further and provides guidance by evaluating possible scenarios and identifying optimal decision policies.

The interconnection between operations management problems and the

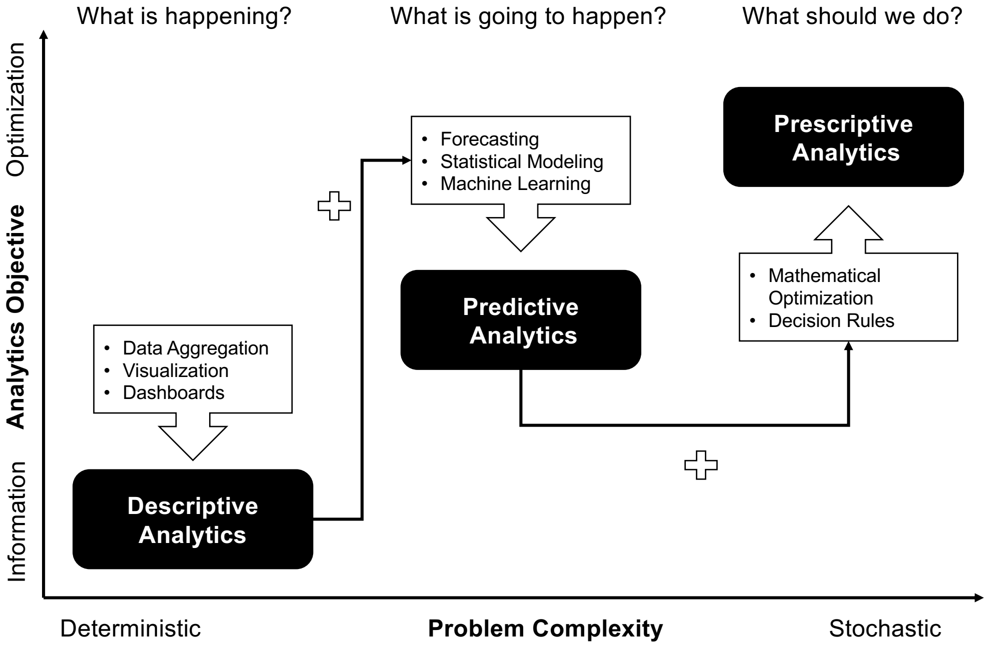


Figure 1.2: Analytics stack

analytics stack is visualized in Figure 1.2. In deterministic planning problems, descriptive analytics applications can provide real-time information based on big data and allow planners to make appropriate adjustments to the parameters of the planning models (Souza 2014).

However, operation managers face uncertainty in many real-world problems. Traditional planning models require either a point forecast or a known (or at least estimated) probability distribution of the unknown variables to solve these stochastic problems. Hence, statistical forecasting has traditionally played an important role in the operations management community (Stevenson, Hojati, and Cao 2007). Naturally, these applications can benefit from highly predictive machine learning techniques. Predictive analytics has proven useful in stochastic planning problems such as customer selection for direct marketing (Moro, Cortez, and Rita 2014) and customer churn prediction (Coussement, Lessmann, and Verstraeten 2017).

However, the way to derive optimal policies from forecasts is not straight forward for many applications. Here, prescriptive analytics has the potential to build on operations management techniques such as mathematical optimization and decision rules to determine optimized policies. Prescriptive analytics has been successfully applied in settings such as repair crew routing (Tulabandhula and Rudin 2014) and inventory management (Huang and Van Mieghem 2014). Overall, leveraging business analytics allows companies to reduce operational costs while maintaining or even increasing service levels simultaneously.

1.2 Research Questions

According to a recent study by Dresner Advisory Services (2017), the adoption of big data analytics reaches 53% across 4,000 interviewed companies and is expected to proliferate. However, currently, most companies focus on descriptive analytics, such as reporting, dashboards, and advanced visualization. In contrast, the adoption of predictive analytics is much slower, with only 23% of companies using it, a figure essentially unchanged from the prior year. This finding indicates that there is still a lack of research regarding new applications for predictive analytics. As indicated by a variety of calls for papers and special issues in leading journals (Giesecke et al. 2018; Hull et al. 2018; Sanders and Ganeshan 2015), the lack of research regarding prescriptive analytics is even bigger. This finding motivates the guiding research question of this thesis:

Guiding Research Question *How can information systems combine state-of-the-art machine learning techniques and operations management modeling to provide prescriptive analytics models that are robust to prediction errors?*

While trying to shed light on this overreaching research question, I structure the thesis along with the latter two steps of the analytics stack. As predictive models are a key component of any prescriptive modeling activity, the first two parts of this work try to explore and evaluate potential new applications for predictive analytics based on two different case studies. Thus, the first subordinate research question I aim to answer is:

RQ1 *What are appropriate machine learning setups (performance metrics and models) for different prediction tasks and how do these setups perform compared to traditional approaches?*

Building on the findings of the predictive analytics case studies, I develop a data science toolbox for prediction tasks in the manufacturing sector to bridge the gap between machine learning research and specific practical needs. To this end, I aggregate the previous findings and highlight key data preparation and analysis steps combining methods from machine learning and business information systems to guide the development of predictive analytics solutions. Consequently, the second subordinate research question of this thesis is:

RQ2 *What are guidelines and best practices for modeling, feature engineering and model interpretation in the context of industrial analytics applications?*

Based on these guidelines, predictive models in the context of industrial analytics can be designed. Providing additional information to decision-makers, these models themselves have business value for many companies. However, mere forecast information cannot be translated into valuable action in many manufacturing settings. To this end, new policies leveraging the information provided by the machine learning models have to be determined by means of prescriptive analytics. Therefore, the third subordinate research question is:

RQ3 *How can prescriptive analytics operationalize predictive models to provide sensor-based decision support for manual processes?*

In contrast to predictive forecasting models, the resulting prescriptive planning model can suggest optimized policies to a decision-maker based on raw sensor data. However, even sophisticated machine learning algorithms do not provide perfect forecasts in real-world applications. Accounting for the shortcomings of the underlying predictive models can improve the quality and robustness of prescriptive planning models. By explicitly integrating the quality of the forecasting model into the planning model, I answer the fourth subordinate research question:

RQ4 *How can prescriptive planning models take the uncertainty of the underlying forecasting models into account?*

1.3 Structure

While trying to explore the overreaching research question, this thesis is composed of four independent parts that have been published as research articles[2]. The first and the second article identify novel use-cases for predictive analytics and aim to answer the first subordinate research question (**RQ1**). Additionally, the second article aggregates the findings of the case studies and provides guidelines and best practices for predictive analytics in industrial applications in the form of a toolbox (**RQ2**). In the third article, we embed a predictive forecasting model in a mathematical optimization problem to evaluate a prescriptive analytics system in a manufacturing environment (**RQ3**). The last article aims at answering **RQ4**. To this end, we describe and evaluate a prescriptive planning model taking the underlying uncertainty of the forecasting model into account. Figure 1.3 relates the chapters of this thesis to the research questions.

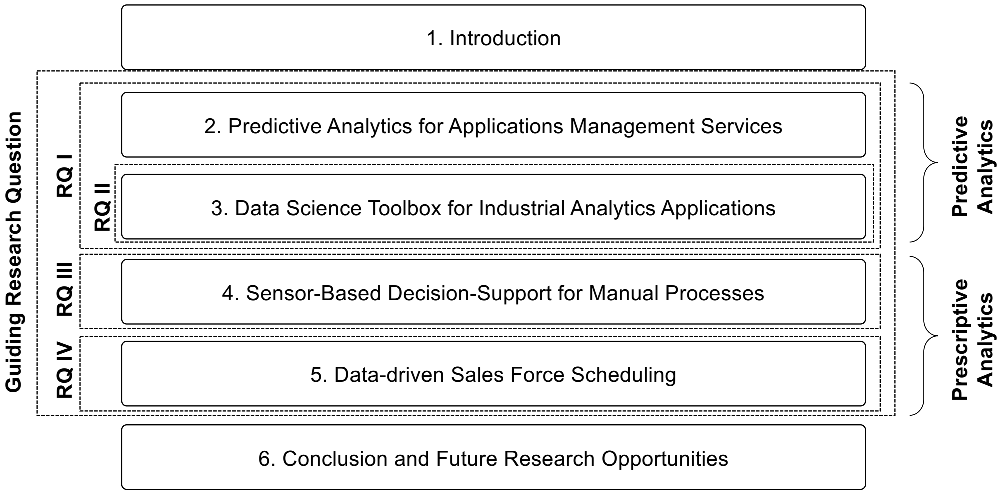

1. Introduction
2. Predictive Analytics for Applications Management Services
3. Data Science Toolbox for Industrial Analytics Applications
4. Sensor-Based Decision-Support for Manual Processes
5. Data-driven Sales Force Scheduling
6. Conclusion and Future Research Opportunities
Guiding Research Question
RQ I
RQ II
RQ III
RQ IV
Predictive Analytics
Prescriptive Analytics

Figure 1.3: Chapters and structure of the thesis

More specifically, the first article "Predictive Analytics for Application Management Services" (Chapter 2) analyzes the potential of predictive analytics in the context of capacity planning and staffing at an IT service desk. In this chapter, we collaborate with an IT service management firm to develop and evaluate an IT service demand forecasting model using machine learning techniques.

[2] See Appendix A for an exhaustive list of publications.

The second article “Towards a Data Science Toolbox for Industrial Analytics Applications” (Chapter 3) sheds light on predictive analytics applications in production environments. Specifically, we put forward guidelines and best practices for modeling, feature engineering, and model interpretation in this domain. Subsequently, we illustrate the usage of this toolbox utilizing a real-world manufacturing defect prediction case study.

The problem of deriving decision support for individual workers from predictive models is addressed in the third article “Big Data on the shop-floor: Sensor-Based Decision-Support for Manual Processes” (Chapter 4). To explore the potential of such prescriptive solutions, we illustrate the main steps and major challenges in developing and instantiating a prescriptive decision support system in a high-tech composite manufacturing setting. By leveraging techniques from statistical learning, we are able to identify the location of leaks at a high degree of confidence. Subsequently, we derive and evaluate optimized search policies by embedding the leak forecast into the underlying searcher path problem. However, the derived policy does not take the quality of the underlying forecasting model into account.

This problem is approached in the fourth article “Data-driven Sales-Force Scheduling” (Chapter 5). In this article, we present a novel data-driven approach to sales-force scheduling. On the example of a data set provided by DAW, a leading German manufacturer of paint and coating solutions, we introduce a machine learning model predicting the benefit of additional sales activity. Subsequently, we determine optimized sales force schedules based on the expected value of the sales effort. Hereby, we explicitly account for the uncertainty of the prediction model and benchmark the performance of the novel prescriptive policy against two baseline policies.

Finally, the thesis is summarized, future research opportunities are outlined, and the work is concluded.

2 Predictive Analytics for Application Management Services

With digitization efforts across all industries, IT consulting firms have enjoyed ever-increasing demand for their services. To cope with this demand surge, long-term hiring decisions, as well as short-term capacity planning and staffing, are of crucial importance for business viability. Predictive analytics methods offer enormous potentials to support planning and staffing of IT service desks to ensure both high capacity utilization and service levels. However, the current state-of-the-art for these planning activities still relies on traditional statistical forecasting methods. We collaborated with an IT service management firm to develop and evaluate an IT service demand forecasting using machine learning techniques. This approach allows us to improve planning accuracy by more than 30% compared to standard approaches.[3]

2.1 Introduction

Greater business data availability and IT ubiquity have created a growing need for useful theories and tools for information extraction (Fayyad, Piatetsky-Shapiro, and Smyth 1996). With constantly increasing computing power, new possibilities to gain insights from data have arisen (Gualtieri, Powers, and Brown 2015). Consequently, the buzzword “Big Data” has attained high attention in almost all areas of business. Companies have started to see the opportunities for turning data into a commodity of high

[3] This paper is published in the proceedings of the 26th European Conference on Information Systems (Stein, Flath, and Boehm 2018).

value for strategic and operative decision making and for providing competitive advantage (Waller and Fawcett 2013). Following Aggarwal (2015), the process of data mining is further gaining importance.

One important application of data mining is predictive analytics, the forecasting of future events by using past data. Modern, effective, and convenient tools have accelerated the popularity and use of predictive analytics throughout various business (Gualtieri, Powers, and Brown 2015). Time series forecasting constitutes a part of predictive analytics in which predictions are made for temporal data. Here, all recorded data is connected to a precise date and time, and the forecasts are predominantly made by using internal structures of the data such as seasonality and trend (Aggarwal 2015). While for time series forecasting tasks in business, statistical methods have been applied for many decades, machine learning (ML) models took root as a contestant for such tasks only in the last decade. Nowadays, ML methods play a significant role in the analysis of large amounts of data, as they can learn with low or even without supervision and improve with the amount of data they are fed (Alpaydin 2010). However, for forecasting temporal data, the approach of training an ML model is different from applying it to time-independent data (Bontempi, Taieb, and Le Borgne 2012). In principle, all ML models are able to perform time series prediction tasks. However, depending on the number of selected lags, the number of features can get very large, which is why models with faster training times are deemed beneficial.

In this work, we analyze how predictive analytics can be applied in order to forecast the future demand for the application management and support division of an SAP consulting company. Given digitization initiatives across all major industries, this firm has seen significant growth over the recent years. In turn, long-term hiring decisions, as well as short-term capacity planning and staffing, need to be able to keep up with this demand surge. On the operational level, detailed forecasting is essential for employee capacity and workload planning. On the strategic level, the business unit recognizes an opportunity for better gross profit estimation as well as budget planning. These tasks are of crucial importance for business viability and signify the importance of business intelligence applications (Popovič, Turk, and Jaklič 2010).

Along these lines, we set up a data science study addressing two guiding research questions:

RQ1.1 What is an appropriate machine learning setup (performance metric and model) to predict IT service management support demand?

RQ1.2 How does the machine learning setup perform on different aggregation levels and forecasting horizons compared to traditional forecasting models?

2.2 Related Work and Preliminaries

In the past decades, the importance of IT service management is constantly growing. Consequently, this field is gaining an increased amount of interest in the information system community (Iden and Eikebrokk 2013; Imgrund et al. 2017). Reviewing the relevant literature, a set of often discussed research questions can be identified. A variety of empirical studies focuses on the implementation strategies for IT service management and the success factors of these strategies (Cater-Steel, Tan, and Toleman 2006; Cater-Steel and McBride 2007; Cater-Steel 2009; Hochstein, Tamm, and Brenner 2005; Iden and Langeland 2010; Marrone and Kolbe 2011; McBride 2009). Possible outcomes and benefits of these implementations are analyzed by Disterer (2012), Hochstein, Tamm, and Brenner (2005), and Marrone and Kolbe (2011). In contrast, only a few studies related to IT incident management can be found in Business Intelligence literature. Here, most research focuses on the labeling of service requests. Maksai, Bogojeska, and Wiesmann (2014) and Diao, Jamjoom, and Loewenstern (2009) develop classifiers to reduce the manual labeling effort. Goby et al. (2016) use a combination of topic modeling and predictive analytics to identify relevant topics and assign them to help desk tickets automatically.

However, there is a lack of research regarding the application of business intelligence in order to forecast the number and workload of incoming service requests. Hence, we widen our search horizon to other related research streams by following a three-step approach: (1) exploring literature on related business topics; (2) analyzing literature on time series forecasting with machine learning and hybrid methods; (3) finding papers utilizing gradient boosting and in particular extreme gradient boosting (XGBoost). Analyzing the literature in detail, we can assign the relevant research papers to different application areas (Table 2.1).

Table 2.1: Literature grouped by area of application

Area of Application	Count	Examples
Business	8	call center arrivals, tourism demand, accounting data
Energy	8	electricity demand, status of water pumps, wind ramp events
Financial	7	stock & commodity prices, exchange rates
Science	7	breast cancer gene expression, sunspots, several competitions
Environment	2	metropolitan air pollution, waste generation
Engineering	2	compressor failures, fuel consumption

We find that business is one of the predominant sectors, with a total of eight relevant papers. Here, call center arrival forecasting through statistical forecasting methods is analyzed four times (Aldor-Noiman, Feigin, and Mandelbaum 2009; Shen and Huang 2008; Taylor 2008). Foster (1977) show that time series forecasting is not a new topic by forecasting the quarterly seasonal accounting data of several firms using ARIMA. Cankurt (2016) predict future tourist demand employing a random forest model. They show that ensembling methods outperform single models.

Energy is another popular sector, in particular, electricity demand forecasting. For this task gradient boosting is applied (Taieb and Hyndman 2014; Kim et al. 2015; Nassif 2016; Mayrink and Hippert 2016) as well as a statistical approach (Taylor 2010). Also in the energy sector, but for event classification and not for time-series forecasting, another two approaches which utilize gradient boosting are found (Arymurthy and Darmatasia 2016; Gupta et al. 2016).

Looking at financial time series forecasting, Krollner, Vanstone, and

Finnie (2010) conduct a study on machine learning methods applied in this area. They find that artificial neural networks are the predominant technique, particularly for the prediction of stock market movements. The oldest relevant paper on financial time series forecasting found in the literature search was published in 1992. Here, an ANN is exploited for the prediction of a multivariate time series of monthly flour prices (Chakraborty et al. 1992). Huang, Nakamori, and Wang (2005) apply support vector machines for stock market prediction. Tay and Cao (2001) and Pai and Lin (2005) compare the performance of support vector machines and artificial neural networks in terms of stock price and index predictions. The only work applying a boosting method in the financial sector is the prediction of gold price volatility (Pierdzioch, Risse, and Rohloff 2016).

2.3 Research Approach and Case Study Overview

Consulting firms collect large amounts of data, including customer-related as well as problem-related information. Our industry partner, an IT consulting firm specialized in the areas of retail and logistics, aims to precisely forecast future demand for its support unit. The firm seeks to generate business value from the available data and the forecasts by improving the staff assignment decisions in the short-term and the business development estimations in the long term.

We want to address this prediction task as a data science study following the guidelines for applying big data analytics (Müller et al. 2016). Correspondingly, we structure our analysis along with the proposed three phases:

Data collection For the extraction of the required data, we harness several internal databases. The resulting raw data set contains all employee bookings from September 2003 to December 2016, resulting in 358,184 entries and 23 variables. Each entry relates to a working hour booking of an employee on a specific project. For each booking, the date, time, and duration along with a name and abbreviation are given. Further, the employee and the project plus the associated customer are specified. Additionally, the data consist of ticket information, distinguishing between support cases and priorities along with annotations, customer information, and time-based corrections.

Data analysis We develop a predictive model to forecast future demand for support requests. Starting with four statistical time series models, we increase the performance through a hybrid model. To this end, we engineer new time-based and non-time-based features from the data set at hand. To leverage these features, we combine the statistical forecasts with an advanced XGBoost model (Chen and Guestrin 2016).

Result interpretation The system is evaluated in Section 2.6. On the one hand, the machine learning approach significantly improves the short-term forecast allowing the company a more efficient employee scheduling. On the other hand, the long-term forecast can be improved, allowing the company a more precise estimate of the business development.

2.4 Data Collection

In this section, we describe in detail the data collection process as well as the resulting data set. The resulting data set contains date-derived as well as non-date-derived variables describing the booking duration of support requests over the last 13 years. We harness several internal databases for the extraction of required data.

2.4.1 Data Extraction and Cleaning

In the problem at hand, support requests can be distinguished into either tickets or task. A customer initiates tickets while internal employees trigger tasks. Based on the urgency of a request, each ticket or task is assigned one out of five status codes.[4] Additionally, we can query information on the billing mode of a request. Here, a distinction between the three types can be

[4] Statuses by urgency (in descending order): Incident, Service Request, Problem, Request for Change, Change

made. For billable requests, the working hours are charged on the invoice of a specific customer while they are only listed and not charged for reportable bookings. Internal requests are invoiced internally and can either originate from customers with included hours or bookings for employee education and training. Additional customer information can be used to augment the data set. Each of the 33 customers in the data set is categorized into one of twelve businesses.[5] Each of the businesses is assigned to one of three sectors, namely manufacturing, services and retail. We include a dummy business and sector for internal in-house bookings. The resulting raw data set consists of 358,184 entries and 23 variables describing all employee bookings from September 2003 to December 2016.

The analysis of data revealed 114,093 missing values in the target variable *booking.duration*. These observations can not be imputed and have to be removed as they resulted from inconsistent database structures. Subsequently, we identify several non-relevant variables and remove them based on one of the following reasons: (1) redundancy, (2) inconsistency, and (3) sparsity.

2.4.2 Exploratory Data Analysis

In order to understand the underlying structure of the problem, we conduct an exploratory data analysis of the remaining data set (Tukey 1977). To understand the following analysis, it is important to note that the data for 2003 and 2016 is incomplete since the recordings started in September 2003 and ended in November 2016.

In Figure 2.1, we observe a constant increase in the number of bookings as well as in the total duration of bookings (i.e., workload). In recent years, tickets account for roughly 50% of the bookings but only about 30% of the workload. Hence, we conclude that support cases triggered by internal employees require significantly more processing time than the tickets invoked by external customers. Zooming into the business and sector gives further inside into the data. Most of the workload can be assigned to the four businesses engineering, food, IT, and OEM. While the former two are in the

[5] Business to Business (B2B), Business to Consumer (B2C), Automotive, Chemical, Construction, Defence, Engineering, Food, Pharmaceutical, IT, Logistics and Original Equipment Manufacturer (OEM)

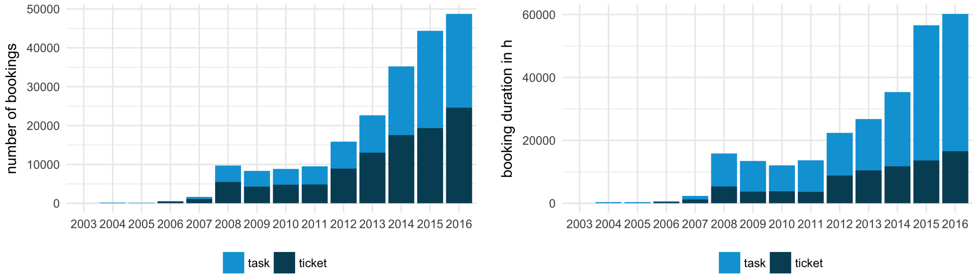


Figure 2.1: Number of bookings vs. total booking duration

sector manufacturing and the latter two in the sector services. Especially in the last two years, we observe an increase in the manufacturing sector and a decrease in the service sector. We also observe heterogeneity in the processing behaviors of customers. The number of ticket hours is relatively low for OEMs and nonexistent for construction clients. Opposed to that, the logistics and IT sector have a much higher amount of working hours book on tickets.

In addition to the analysis of the differences between tasks and tickets, we also analyze the observable differences in the billing modes. Here, the manufacturing sector has the highest total booking duration. It is formed of 43% billable, 10% reportable, and 47% internal bookings. Services constitute the second-largest sector. Here, internal bookings account for 73% of the total workload. Remaining modes are billable at 25% and reportable at 2%. The dummy sector for in-house bookings constitutes the third largest sector. By definition, it only reports internal bookings as the support requests can not be assigned to a client. The least amount of workload is booked in the retail sector. Here, the majority of requests (61%) is booked internally, followed by 30% reportable and only 9% billable bookings. Additionally, we also observe a heterogeneity regarding the billing modes inside the sectors on a business level. The distribution across the sectors and billing modes is summarized in Table 2.2.

Table 2.2: Billing modes across sectors

Sector	Billable	Reportable	Internal	Total
Retail	9%	30%	61%	0.8%
In-house	0%	0%	100%	10.3%
Manufacturing	43%	10%	47%	56.2%
Services	25%	2%	73%	32.7%

Going further into detail, we observe a long tail distribution of working hours on the customer level. While the three biggest customers are responsible for over 50% of the workload, there is a multitude of customers with very sparse support requests.

Analyzing employee development, we find that the booking hours, as well as the number of support requests, are increasing over proportional to the number of workers. Hence, an increase in productivity due to process improvement can be concluded. Additionally, idle times can be reduced due to better planning. Therefore, employees can handle not only more requests but also a higher workload. These findings are visualized in Figure 2.2.

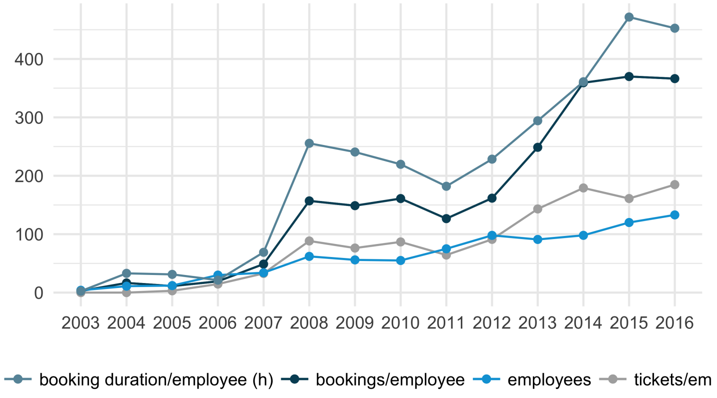


Figure 2.2: Development of employee number and average booking count and booking duration per employee

Extracting the recurring seasonal patterns from the time series, we find that the seasonal fluctuation is approximately 500 hours per month.

Comparing this to the booking duration of 7000 at the end of 2016, the seasonality component only captures about 7% of the variance.

In order to train predictive models, it is imperative to aggregate the data on a specific level (Geurts 2002). Since our objective is to support the decision making on different management levels, forecasts for periods ranging from one day to one year have to be determined. Hence, we define three levels of aggregation: daily, weekly, and monthly. We aggregate the target variable *booking.duration* for each level by summing up the single observations.

2.5 Model Setup

Prior to any modeling activities, a suitable evaluation metric has to be chosen. This metric has to account for the specific properties of the problem at hand. Following Davis et al. (2007), the metric selection is fundamental for the success—or failure—of every data science project. Regarding time series forecasting a variety of different metrics with different strengths and weaknesses is available. According to a classification broad forward by Hyndman and Koehler (2006), each measure is either a "scale-dependent measure," a "percentage error measure" or a "relative error measure."

In the forecasting task at hand, the scale of the workload varies over time. Hence, we chose to select a "scale-dependent measure." While the mean absolute percentage error (MAPE) is the most utilized quality measure, it comes with several weaknesses. According to Tofallis (2015), the MAPE tends to prefer models underestimating the realized values. Hence, Armstrong (1978) suggest a symmetric version of MAPE called sMAPE. Since the original sMAPE has a range of $[-\infty, \infty]$, Hyndman and Koehler (2006) suggest using absolute values in the denominator, which is the version used in this paper. The metric is specified as

$$sMAPE = \frac{200}{n} \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|}$$

with y_i being the actual value, $\hat{y}_i$ the prediction and n the number of fitted points. Since for $y_i, \hat{y}_i = 0$ the function is undefined, the sMAPE is suggested to be only used for predicting values which are significantly higher than zero (Hyndman and Koehler 2006). The range of sMAPE is $[0, 200]$. While this metric does not allow for an easy interpretation, it is well suited to compare the performance of different predictive models.

2.5.1 Modeling Approach

Having established a suitable evaluation metric, a set of appropriate forecasting models has to be selected. To this end, we first select a set of statistical forecasting methods that later serve as a benchmark for the proposed machine learning approach. Following Bontempi, Taieb, and Le Borgne (2012), the literature on time series analysis and forecasting is mostly based on these methods. Following Hyndman and Athanasopoulos (2014), two simple approaches can be identified for statistical time series forecasting. The average or mean method is a simple approach mostly used as a benchmark for comparing it to more sophisticated models. It is also referred to as the historical average as the prediction of future outcomes is equal to the average of the historical outcomes. As another simple benchmark approach, the naïve method sets all forecasts to the last observed value.

Furthermore, the seasonal naïve approach can be utilized for highly seasonal data. Here, the forecasts are set to the values of the last value observed in the same season, which for instance could be the last observed value for the same month in the previous year. The autoregressive integrated moving average (ARIMA) is a more sophisticated approach of statistical time series forecasting. It is often applied in the relevant literature (Ho, Xie, and Goh 2002; Zhang 2003; Khashei and Bijari 2011). However, ARIMA models can only be applied to stationary time-series without missing data Khashei and Bijari (2011). Being able only to capture linear patterns, the application of these models in real-world problems does not always reveal good forecasting results Zhang (2003).

To overcome this shortcoming, we suggest applying advanced machine learning (ML) models. Such techniques are often referred to as black-box or data-driven models. They represent non-parametric and non-linear models that learn stochastic dependencies between historical and future data. While most ML methods can theoretically be applied for time series prediction tasks, only a small set of algorithms has gained popularity in this field. Friedman, Hastie, and Tibshirani (2001) and Hastie (2004) compare the performance of the five most popular methods (Artificial Neural Networks, Support Vector Machines, Multivariate Adaptive Regression Splines, k-Nearest Neighbours and Gradient Tree Boosting) regarding a set of characteristics (natural handling of data of "mixed" type, handling of missing values, robustness to outliers in input space, insensitivity to monotone transformations of inputs, computational scalability (large N), ability to deal with irrelevant inputs, interpretability, predictive power). For each of the characteristics, the performance of the models is rated. While each of the methods shows strengths and weaknesses in their performance, the gradient tree boosting approach exhibits the best performance overall. Additionally, the implementation XGBoost introduced by Chen and He (2015) is known for fast training times and high accuracy in predicting real-world problems Hu et al. (2017) and Taieb and Hyndman (2014). For example, on the data science competition platform Kaggle, out of all winning teams of the overall 29 challenges in 2015, in total 17 used XGBoost. Another advantage of this model other than the fast training times is the included automatic variable selection (Taieb and Hyndman 2014). In particular, we decided to utilize XGBoost for the prediction task at hand for the following key reasons:

- Efficient training process
- High quality of predictions
- Robustness to overfitting

2.5.2 Traditional Forecasting as Benchmark

In order to define a competitive benchmark for the machine learning model, the four different statistical forecasting methods discussed in Section 2.5.1 are applied to the data set at hand. To this end, we use a rolling horizon evaluation with several forecasting horizons for all models on all three aggregation levels. The model quality is assessed for a one-day, three-day and seven-day ahead forecast for the daily data, a one-week, two-week and four-week forecast for the weekly data and a one-month, six-month and twelve-month forecast for the monthly data.

Table 2.3 summarizes the sMAPE metric for the different models and aggregation levels. Looking at the overall results, we see the out-performance of the naïve and the ARIMA methods over the other forecasts. This finding even holds for all aggregation levels and forecasting horizons. While a big improvement in forecast accuracy is generated by moving from daily to weekly aggregation, we see mixed results if the aggregation periods get longer. In general, we conduct that using a statistical forecasting method allows us to improve on the simple average method by up to 35% in terms of the sMAPE.

Aggregation	**Forecast Horizon**	**Average Method**	**Naïve**	**Seasonal Naïve**	**ARIMA**
daily	1 day	115.40	83.99	104.66	77.48
	3 days	116.14	103.01	102.09	81.52
	7 days	120.70	75.29	93.70	79.78
	mean	117.41	87.43	100.15	79.59
weekly	1 week	42.95	27.28	55.67	24.97
	2 weeks	43.45	29.84	54.69	27.17
	4 weeks	43.88	35.02	52.41	29.54
	mean	43.43	30.71	54.26	27.23
monthly	1 month	42.43	16.40	43.70	17.03
	6 months	48.04	21.41	42.94	25.05
	12 months	53.94	31.12	42.94	44.46
	mean	48.14	22.98	43.19	28.85
total	**mean**	**69.66**	**47.04**	**65.87**	**45.22**

Table 2.3: Rolling forecasting evaluation of statistical models

2.5.3 Machine Learning and Feature Engineering

So far, our analysis only relied on past realizations of the target variable. To further increase the quality of the forecast, we now utilize a machine learning approach by building an XGBoost model. This model can generate valuable information based on multiple input variables. The process of developing these features is summarized under the term feature engineering. Following Domingos (2012), this process is critical to ensure the success of any data mining project. Going beyond basic raw features requires a significant portion of business and process understanding as well as creativity and luck. In a time-series setting, the features can either be time-based or non-time based. At first, we create the following set of time-based variables which serve as the basis for the different aggregation levels:

Table 2.4: Time based variables

Feature	**Description**
day, *month*, *year*, *weeknumber*	Features derived from the date variable.
weekday	Weekday of each date as an ordered factor.
isWeekend	Due to the observations of differing workloads, the distinction between weekends and weekdays is a promising feature (Mayrink and Hippert 2016).
holiday	We find a strong correlation between the booking duration and official German holidays. Additionally, bridge days appear to have an impact on the booking duration.
seasonal, *trend*	Seasonality and trend component of the daily time series.

For the weekly and monthly scales, the variables above have to be further aggregated:

Table 2.5: Weekly and monthly aggregation

Feature	**Description**
holidays.in.week, *holidays.in.mth*	Total number of holidays in a week or month.
bridgedays.in.week, *bridgedays.in.mth*	Total number of bridge days in a week or month.
weekdays.in.mth, *weekenddays.in.mth*	Number of weekends and weekdays in a month (only for the monthly aggregation).
seasonal, *trend*	Seasonality and trend component of the weekly or monthly time series.

In addition to the time-based variables, we create an additional set of explanatory features to increase the predictive power of the model. Hence, the features summarized in Table 2.6 are calculated for the relevant aggregation periods.

Table 2.6: Additional variables

Feature	**Description**
isTicket.true, *isTicket.false*	Count of bookings that are a ticket or a task.
total.bookings	Total number of bookings for each day.
customer.bookings	Total number of bookings booked on customers. By implication, bookings on internal projects and tasks are excluded.
customer.durations	Total time booked across customers (w/o internal bookings).
*billing.mode.**	Booking duration for each billing mode (*).
*sector.**	Booking duration for each sector (*).
*business.**	Booking duration for each business (*).

Since for the predictions of time series data solely date-derived features are known in advance, the forecasting can only be based on the behaviour of past data (Bontempi, Taieb, and Le Borgne 2012). Hence, all variables that are not derived from the date, including the target variable *booking.duration.*, need to be shifted in time for the model to find stochastic dependencies between past and future data. Said shifting is predominantly referred to as lagging. In order to find the optimal lags, the autocorrelation for each aggregation level is analyzed.

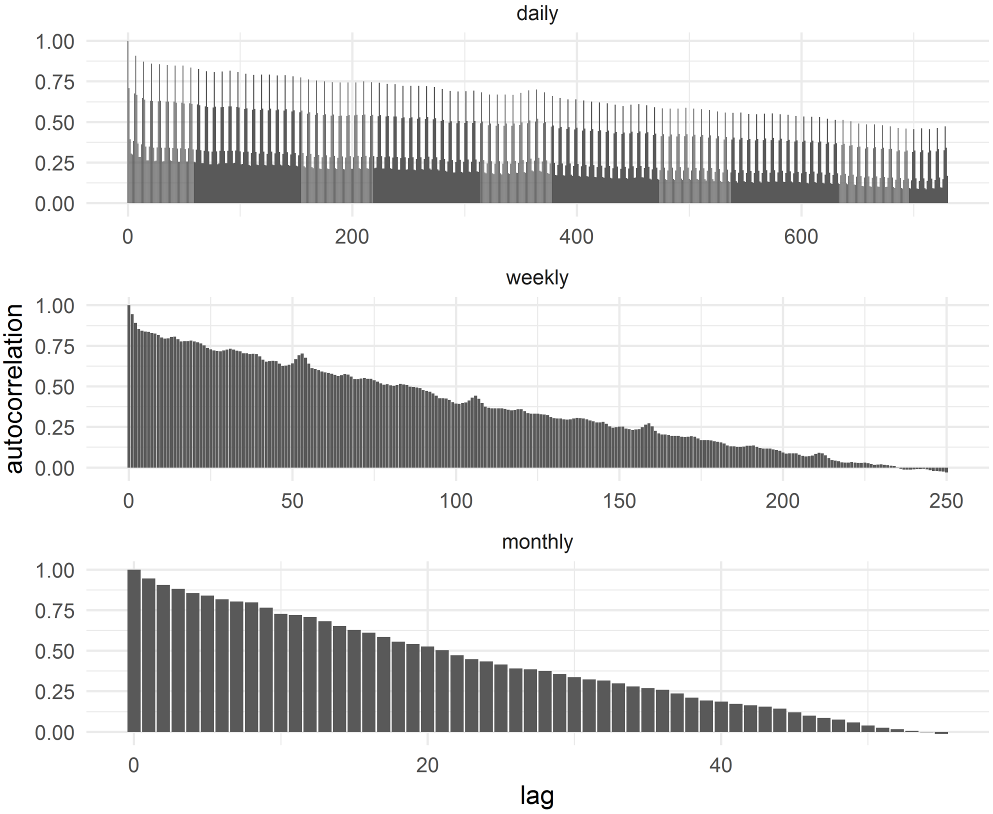


Figure 2.3: Autocorrelation of booking duration variable for each aggregation level

As visualized in Figure 2.3, the daily model shows a high autocorrelation every seven days, which endorses the weekly seasonal pattern. When looking at a higher lag horizon in the daily lags, the plot shows a mostly linear decline while keeping the overall seven-day pattern. Looking at the weekly booking time data, every 52 weeks or respectively one year, a slight increase in the autocorrelation can be observed. The graph shows an overall linear decline down to the 235-week lag. Opposed to the prior aggregation levels, in the monthly booking hour data, no repeating peaking pattern can be observed. The graph shows an overall linear decline down to the 55-month lag. For all aggregation levels, the autocorrelation values start to get negative at about a 4.5-year horizon.

Based on this analysis, we choose the lags summarized in Table 2.7 for further modeling.

Table 2.7: Specified lags for each model

model	**lags**
daily	7, 14, 21, 28, 91, 182, 365, 730, 1,095, 1,460 days
weekly	8, 16, 52, 104, 156, 208 weeks
monthly	12, 24, 36, 48 months

The use of high lags for all models contradicts the correlation analysis. However, since only the autocorrelation of the target variable and not the individual feature correlations are analyzed, and XGBoost includes an automatic feature selection, we decide to use several higher lags.

Following Krollner, Vanstone, and Finnie (2010), the combination of several forecasting methods in order to achieve higher prediction quality is a promising approach. Hence, we decide to add the forecasts of ARIMA –the best performing statistical model– as an additional feature to the ML approach.

2.5.4 Model Refinement

Based on the features created above, we train nine separate ML models to tackle the prediction problem at hand. To leverage the full potential of the suggested approach, a single model is trained for each aggregation level (daily, weekly, monthly) and forecasting horizon. Additionally, the models are updated for each subsequent evaluation step following the rolling horizon approach. While this procedure provides us with well-tuned models, it is computationally expensive due to the size of the data set at hand.

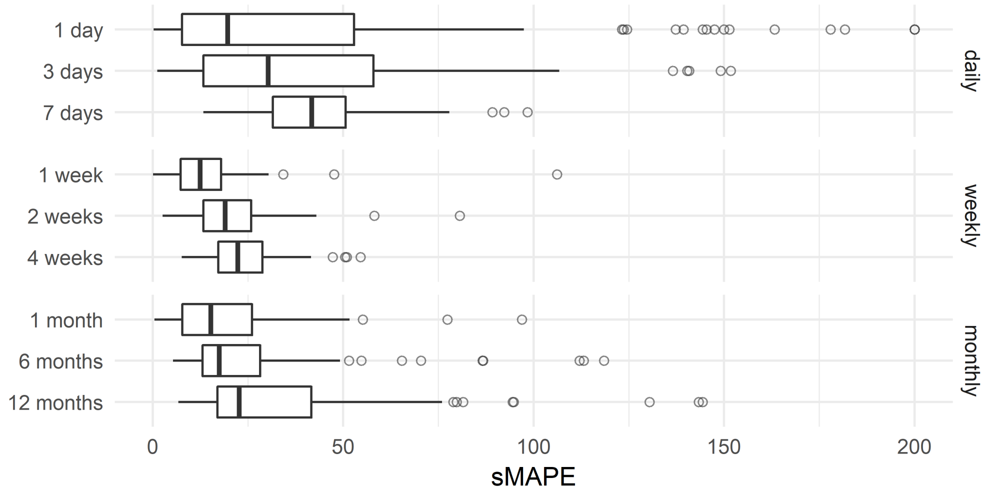


Figure 2.4: Comparison of sMAPE metrics throughout the models

The performance of the models can be further improved by selecting a good set of model parameters. In order to find a good set of parameters, we perform a hyper-parameter grid search. To this end, over 400 possible parameter combinations are tested for each aggregation step. Due to computational limitations and in order to avoid overfitting, we decide to apply the same set of parameters for each forecasting horizon. Hence, the final hyper-parameter grid search has to be performed three times, resulting in a total of 1,200 trained models. Table 2.8 summarizes and describes the tunable parameters, the tested ranges, and the selected values.

The quality of the nine ML models is evaluated using the same rolling horizon as in the statistical forecasting approach. The results are summarized in Figure 2.4. As expected, a decreasing accuracy can be observed with increasing forecast horizons. Additionally, the forecasting quality is better on higher aggregation levels. In terms of the sMAPE, the weekly and monthly models perform roughly twice as good compared to the daily aggregation level. This finding also holds in terms of forecast reliability, as the variance of the sMAPE is significantly reduced for the two higher aggregations.

Table 2.9 compares the performance of the best statistical forecasting model ARIMA and the suggested ML approach. On the daily aggregation level, the biggest performance increased can be reported. Here, the average sMAPE is reduced by roughly 46%. With a reduction of roughly 27%, we also observe substantial improvements on the weekly level. On the other hand, ARIMA and XGBoost perform at about the same quality on monthly aggregated data. Here, a significant increase in forecasting accuracy can only be observed for long forecasting horizons. The overall average performance in terms of the sMAPE shows a 34% better performance for the ML model.

2.6 Result Discussion and Interpretation

We developed and evaluated a system for support request forecasting in an IT consulting setting. We first determined several traditional statistical forecasts on several aggregation levels that serve as benchmarks for the later evaluation. Subsequently, we derived a meaningful feature set for a sophisticated ML forecast and performed hyper-parameter optimization.

In the age of big data, researchers, as well as practitioners, can no longer rely exclusively on standard statistical methods (e.g., ARIMA) to generate business insights from large data sets. Instead, the use of machine learning becomes inevitable as these approaches are better suited to handle thousands of variables or work with unstructured data. Breiman (2001b) and Shmueli et al. (2010) show that these approaches are of paramount importance in studies aiming at prediction instead of description. The main advantage of state-of-the-art ML algorithms is that they make less rigid statistical assumptions and can work with data sets of very high dimensionality. Additionally, these methods cannot only capture non-linear relationships but also pick up higher-order interaction effects between variables. On the downside, these black-box algorithms (e.g., gradient boosting machines) typically generate incomprehensible models and rules. However, the interpretability of the rules used by the algorithms is important if subsequent actions based on the predictions are to be taken by human decision-makers (Martens and Provost 2014; Diakopoulos 2014).

Answering the need for comprehensible prediction models as identified by Breuker et al. (2016), we analyze the importance of the features through the information gain. Figure 2.5 visualizes 20 most important features for all aggregation levels.

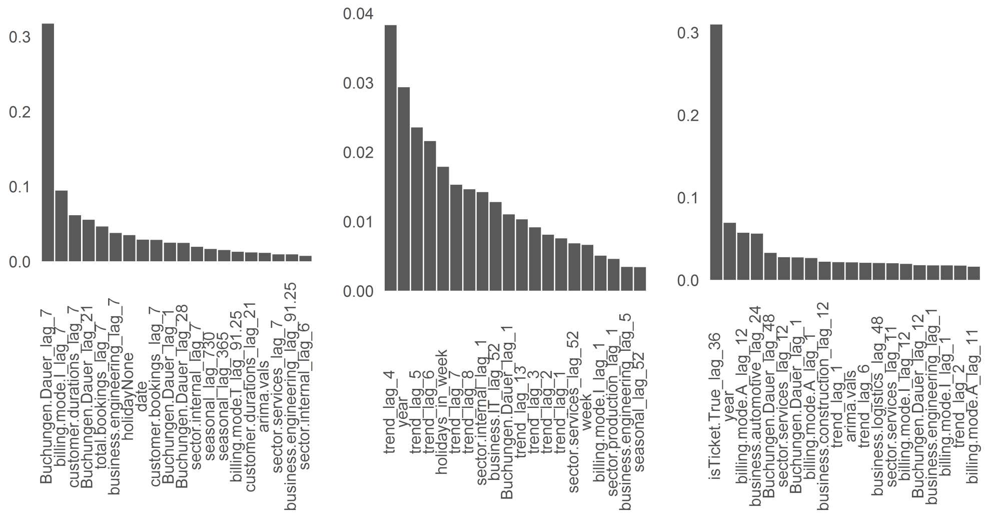


Figure 2.5: Feature importance for each aggregation level

Looking at the feature importance of the daily model, the importance of the seven-day-lagged booking duration (*Buchungen.Dauer*) provides the most information gain. In the weekly model, more features are needed to leverage the full potential of the suggested approach. While the four-week-lagged *trend* is the most important feature, a total of nine *trend* features with different lags can be found in the top 20 features. The information gain in the monthly model shows a distribution similar to the daily aggregation with one feature holding the most importance. Here, the variable *isTicket.True* with a 36-month lag has the highest importance, followed by the year variable.

For a better understanding of the model performance, the average employee-per-day error for the XGBoost model is calculated. For this, we assume an employee with an eight-hour workday on five weekdays (40 h/week). Calculating the mean absolute error per day, we find that the daily models average at an error of $\sim$ 2.9 employees per day. The weekly models result in an average of $\sim$ 3.4 employees per day and the monthly models at an average of $\sim$ 2.6 employees per day. Considering the total of 137 employees in the business unit, the forecast should enable a high-quality planning process.

2.7 Conclusion and Outlook

Using a large data set from an IT consulting company, we showcased the development, refinement, and evaluation of a machine learning-based forecasting system for incoming support tasks. Here, the objective is to improve short term as well as long term planning processes to improve capacity utilization and service levels.

After the extraction of raw booking data from several databases, we perform an extensive exploratory data analysis to identify patterns informing the subsequent modeling phase. During this analysis, we clean the data set and remove redundant and sparse variables. Subsequently, an appropriate metric as well as appropriate statistical forecasting methods and a machine learning approach are chosen. Leveraging a powerful feature set, we show the out-performance of the suggested machine learning approach in comparison to traditional forecasting methods. On average, the machine learning model is able to increase the forecast accuracy by 34% depending on the forecast horizon and the aggregation level. Especially for short-term operational planning, the machine learning approach is far superior.

In future work, we intend to extend our case study in the following directions. First, the accuracy, as well as the robustness of the predictions, could be further improved by leveraging additional internal and external data sources (e.g., financial data or press releases) and creating more explanatory features. Second, manual model adjustments could be allowed in order to incorporate human knowledge into the model. To this end, the feature set could be enhanced by adding a variable with expert estimations. Third, the performance of the XGBoost model can be compared to other machine learning algorithms such as geometric semantic genetic programming (Castelli et al. 2016).

Parameter	Description	Test Range	Daily	Weekly	Monthly
learning_rate	Shrinks the feature weights	0..1	0.5	0.01	0.005
gamma	Minimum loss reduction for partitioning tree leaf node	1..10	5	8	4
max_depth	Maximum depth of a tree	2..5	5	4	4
min_child_weight	Minimum number of instances in each node	1..10	7	6	5
subsample	Ratio of data to use for training	0..1	0.6	0.6	0.6
colsample_bytree	Ratio of columns to use for training	0..1	0.6	0.6	0.6

Table 2.8: Hyper-parameter optimization

Table 2.9: sMAPE comparison of XGBoost and ARIMA

Aggregation	**Forecast Horizon**	**ARIMA**	**XGBoost**
daily	1 day	77.48	42.53
	3 days	81.51	42.33
	7 days	79.78	43.30
	mean	79.59	42.72
weekly	1 week	24.97	14.42
	2 weeks	27.17	20.99
	4 weeks	29.54	23.94
	mean	27.23	19.78
monthly	1 month	17.03	18.80
	6 months	25.05	25.32
	12 months	44.46	25.95
	mean	48.14	22.98
total	**mean**	**45.22**	**29.48**

3 Data Science Toolbox for Industrial Analytics Applications

Manufacturing companies today have access to a vast number of data sources providing gigantic amounts of process and status data. Consequently, the need for analytical information systems is ever-growing to guide corporate decision-making. However, decision-makers in production environments are still very much focused on static, explanatory modeling provided by business intelligence suites instead of embracing the opportunities offered by predictive analytics. We develop a data science toolbox for manufacturing prediction tasks to bridge the gap between machine learning research and concrete practical needs. We provide guidelines and best practices for modeling, feature engineering, and interpretation. To this end, we leverage tools from business information systems as well as machine learning. We illustrate the usage of this toolbox by means of a real-world manufacturing defect prediction case study. Thereby, we seek to enhance the understanding of predictive modeling. In particular, we want to emphasize that simply dumping data into "smart" algorithms is not the silver bullet. Instead, constant refinement and consolidation are required to improve the predictive power of a business analytics solution. [6]

3.1 Introduction

In the last decade, the manufacturing sector has seen a tremendous digital transformation. Wireless connectivity, as well as cost decreases for

[6] This paper is published in *Computers in Industry* (Flath and Stein 2018).

sensors and data storage, have paved the way towards a next-generation industrial infrastructure. In particular, there has been a considerable convergence of industrial IT systems and shop-floor automation (see Figure 3.1). Going forward, ubiquitous IT on the shop-floor will be instantiated by self-monitoring production equipment and networked production systems (Reddy 2016). Unsurprisingly, manufacturing companies today have access to a vast number of data sources providing gigantic amounts of process and status data. Manyika et al. (2011) estimate that the manufacturing sector generated more than two exabytes of data in 2010. This data ranges from production status and utilization data to continuous tool and machinery condition monitoring. Yet, creating ever-growing data dumps will not contribute to business value generation. However, if appropriately managed data can be a highly valuable resource that is becoming more and more critical to worldwide business operations. This finding has led to widespread agreement that *data is the new oil* in future IT-augmented systems (Rotella 2012).

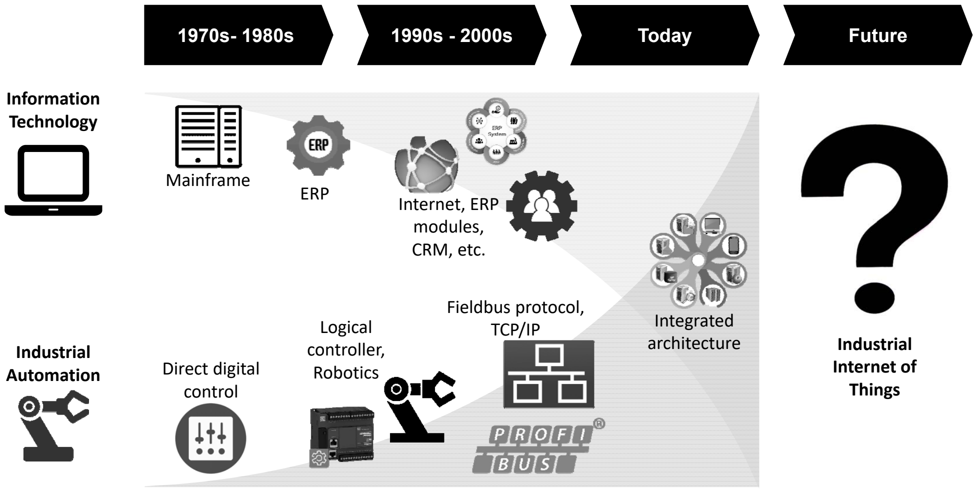


Figure 3.1: Convergence of industrial IT systems and shopfloor automation (adapted from (IoT Analytics 2016))

In turn, companies are hard-pressed to establish novel analytics tools and use cases to benefit from their data treasures. Leveraging this data utilizing new analytics tools offers opportunities to foster data-driven decision-making and increase both efficiency and effectiveness of existing business processes. Such approaches have been discussed in both academic and practitioner literature (Sharma, Mithas, and Kankanhalli 2014; Chen, Chiang, and Storey 2012). Revisiting the new oil analogy, an analytics solution resembles an oil refinery which turns a basic resource into a valuable products (IoT Analytics 2016, p.11).[7]

While a plethora of IT consultants has been courting companies to buy into the "Big Data Revolution", companies are often disappointed by the outcomes and overwhelmed by the amount and variety of data (LaValle et al. 2011; McAfee et al. 2012). For now, the promise of industrial analytics mostly remains a mixture of promises, visions, and pilot projects instead of large-scale implementations. To become an indispensable part of the manufacturing engineer's toolbox, it still has a long way to go. The recent influx of machine learning research has brought forward a host of capable algorithms and tools but has not equipped operators and decision-makers with the necessary work-flows and tools. Consequently, there is an urgent need for tool-kits and templates which assist manufacturing decision-makers in navigating through a world of new opportunities.

This paper seeks to address this gap by compiling and explicating a data science toolbox for prediction tasks in manufacturing systems. We highlight key data preparation and analysis steps. In particular, we combine methods from machine learning and business information systems to guide the development of predictive analytics solutions. We subsequently apply the toolbox to a case study from a major manufacturing company. Thereby, we illustrate how a predictive analytics solution can be set up, refined, and evaluated. Prediction tasks in other manufacturing settings will face very similar challenges. Therefore, we are confident that these research questions and our results can be generalized and applied beyond the specific case at hand.

3.2 Related Work and Preliminaries

Data ubiquity due to the integration of networked machines as well as the rise of machine learning algorithms leads to a transformational change

[7] In a recent IoT Analytics study 15% of respondents consider industrial data analytics as a crucial success factor today. Additionally, 69% think it will be crucial in 5 year's time.

throughout all major industries. Recent research conducted by General Electric, Accenture (2015) estimates that the Industrial Internet offers a $15 trillion opportunity due to reduced costs, productivity gains, and new products. They show that the need to leverage the potential of the available data sets is of high urgency in the manufacturing sector. However, modern manufacturing environments are characterized by large amounts of sensors leading to data sets that are complex in terms of volume and variety. In the upcoming section, we show the different techniques that are available to tackle these problems.

3.2.1 Data Science in Manufacturing

With the rise of data ubiquity, the desire to generate insights and business value from this data is ever-growing. Hence, the idea of "business analytics" describing "data science" in a business context (Chen, Chiang, and Storey 2012) has experienced rapid growth over the last years.

Shmueli and Koppius (2011) carve out the difference between explanatory statistical modeling and predictive modeling. They emphasize that explanatory power derived from traditional models does not imply predictive power. Consequently, predictive analytics is needed not only to create models for practical applications but also for theory building and theory testing. Manufacturing companies need to embrace business analytics in order to remain competitive in the global marketplace (Lee et al. 2013). Historically, manufacturing firms have relied on observable process outcomes through shop-floor initiatives like standardized work or continuous improvement. By incorporating advanced analytics, they can also address unobservable problems like machine degradation or hidden defects.

Recent research regarding machine learning applications for manufacturing mainly focuses on technical solutions that are used to identify relevant information from large data sets with many variables. To predict the level of machine degradation, Mosallam, Medjaher, and Zerhouni (2016) apply unsupervised learning to select essential variables from a set of monitoring data. The authors report good results in a turbofan engine as well as a battery health setting. Sipos et al. (2014) design an information system using multiple linear classifiers to predict failures of medical equipment based on log data. Bleakie and Djurdjanovic (2013) propose a method that is capable of predicting system condition by comparing the similarity of recent sensor readings with known degradation patterns. They successfully apply this method in a semiconductor manufacturing setting.

To this end, the existing research mainly provides solutions for specific problems in case study settings. Hence, the goal of this paper is to provide a toolbox for the implementation of data-driven approaches in various manufacturing settings.

3.2.2 Machine Learning

The algorithms behind predictive manufacturing applications can be assigned to the field of data mining. Unlike "normal" algorithms it is the data that tells these data-driven algorithms what the good answer is. In a manufacturing setting, a traditional approach would try to define a set of variables (e.g., weight and form) that identifies defective parts. In contrast, a machine learning algorithm does not need such coded rules but would learn them by examples. These learning techniques can be either unsupervised or supervised. In unsupervised machine learning, the observations have no "labels." Hence, an algorithm is used to identify hidden patterns in the input variables.

In contrast, supervised learning is the task of inferring a function from labeled training data. In supervised learning, each example is a pair consisting of an input object (in most cases a vector) and an output value. Problems with a continuous output space are summarized under the term regression problems while classification describes problems with a discrete output space.

Unsupervised Learning

Unsupervised learning summarizes machine learning algorithms that find hidden structures in unlabeled data (Hastie, Tibshirani, and Friedman 2013). While there are many possible applications in different fields (e.g., association rule mining for recommender systems, generative adversarial networks for image generation (Goodfellow et al. 2014)) we focus mainly on the algorithms used for dimensionality reduction as they are of particular interest in manufacturing settings with increasing amounts of high dimensional monitoring data.

Principle components analysis (PCA) is a popular and well-studied method to transform high-dimensional data sets into low-dimensional data sets. PCA converts a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables. To this end, it finds the n principal axes in the original m-dimensional space where the variance between the points is the highest. By selecting the axes that explain most of the variance, the number of variables is reduced from m to n. Thereby, the bulk of information is preserved as the new variables are combinations of the old variables (Hotelling 1933). However, PCA reaches its limitations if the relationships between the variables are non-linear. This shortcoming is tackled by the recently developed method t-distributed stochastic neighbor embedding (t-SNE). This technique takes a set of points in a high-dimensional space and embeds them in a lower-dimensional space by solving a problem known as the crowding problem (Maaten and Hinton 2008). Due to its flexibility, t-SNE is often able to find structures in data sets where other dimensionality-reduction algorithms fail. However, this advantage comes at the costs of a decreased interpretability as well as the need for a complex hyper-parameter tuning (Wattenberg, Viégas, and Johnson 2016). The selection of the best dimensional reduction algorithms is typically a trial and error process as both algorithms have different advantages and limitations.

After applying one of the above methods, clustering algorithms can be used to group and identify data-inherent structures. These algorithms set the clusters in a fashion to maximize the intra-cluster distance and minimize the inter-cluster distance. Depending on the clustering process a further distinction between hierarchical algorithms (e.g., Ward's method or single-linkage (Ward Jr 1963; Murtagh and Contreras 2012)) and partitioning algorithms (e.g., k-means or k-medoids (MacQueen et al. 1967; Kaufman and Rousseeuw 2009)) can be made.

Supervised Learning

Supervised learning is the machine learning task of inferring a function from labeled training data (Mohri, Rostamizadeh, and Talwalkar 2012). Learning tasks with quantitative labels (such as machine degradation) are summarized under the term regression problems while tasks with categorical labels (such as defective product/non-defective product) are summarized as classification problems (Hastie, Tibshirani, and Friedman 2013). Both problems can be approached using various white-box models and black-box models. White-box models offer greater transparency concerning the rules used to generate predictions while black-box models can often increase the prediction quality at the cost of reduced interpretability.

White-box models The oldest and probably best studied white-box methods are linear regressions for regression problems and logistic regression for classification problems (Galton 1886; Hastie, Tibshirani, and Friedman 2013). They rest on the assumption of a linear relationship between a set of input variables and a single output variable. The linearity assumption is simultaneously the major strength and weakness of this method. On the one hand, it renders the model very simple to understand and efficient to learn. On the other hand, it constrains the predictive power of the model, as many statistical relations are non-linear. To overcome this shortcoming, decision tree model can be considered. They map multiple input variables with an output variable through a tree structure. Thereby, they are able to take account for potential non-linear relationships in the data. While this model class has, in general, a higher predictive power than linear regression, it is also prone to fail to generalize from the training data (Quinlan 1986). Hence, Gaussian process models can be applied. This model class assumes that the output variable follows a Gaussian process fully defined by a mean and a covariance function. The covariance function expresses the expected covariance between the output variable and the input variables. Thus, linear and non-linear relationships in the data can be learned. These models are better suited for complex data compared to linear and logistic regression models. Additionally, the prediction of these models does provide not only a point estimate but also uncertainty information (Rasmussen 2006).

Black-box models More complex and less interpretable models are summarized under the term black-box models. Here, we first consider support vector machines viewing each observation as a vector of all input variables. During the model training, the hyperplane that best separates the different output variables depending on the input vectors is determined. In order to incorporate non-linear relationships, the dimensionality of the input vector is augmented using the kernel-trick if no hyperplane separating all different output variables exists. A major benefit of support vector machines is the good generalization ability and therefore the low susceptibility to over-fitting even for small training data sets (Smola and Schölkopf 2004). Another popular class of black-box models are artificial neural networks. These models consist of several layers of artificial neurons. Each neuron is connected with many other neurons and processes incoming information and propagates the results to other neurons. Neural networks are powerful, very adaptable, and used for many different applications in various fields (Hastie, Tibshirani, and Friedman 2013). A model class recently enjoying increasing popularity are gradient boosting machines. They ensemble many weak learners to a strong predictive model in a sequential fashion. Gradient boosting increases the weight of the samples misclassified by the first model and decreases the weight of the samples that are classified correctly to train another decision tree. This step is repeated n times, where n is the number of boosting iterations. In this way, the algorithm always trains models using data samples that are difficult to learn in the previous round, which results in an ensemble of models that are good at learning different parts of the training data (Friedman 2002). Due to the highly efficient learning as well as the good generalization, these models are predestined for settings with a large number of features and observations frequently arising in modern manufacturing environments.

Evaluation Metrics

Prior to any modeling activities, a suitable evaluation metric for the task at hand has to be chosen. Understanding the importance of the evaluation metric is fundamental for the success—or failure—of every data science project (Davis et al. 2007). Typically, the prediction errors are minimized for regression problems. To this end, the mean absolute deviation (MAD) or the mean squared error (MSE) could be applied.

In contrast to the straight-forward metrics mentioned above, the selection of an appropriate evaluation criterion is more complicated in classification settings. In general, the performance measures used for these problems are derived from the confusion matrix accounting for the number of wrong—false positives (FP) or false negatives (FN)—and correct—true positives (TP) and true negatives (TN)—classifications. However, the metric selection has to account for the specific properties of the given scenario, e.g., skewed classes and misclassification cost distributions (Flach 2003). This is of particular importance for manufacturing scenarios as they are typically characterized by high class imbalanced due to low failure rates.

Standard evaluation metrics for classification problems like accuracy fail in these settings (i.e., a simple model predicting "non-defective" for all parts achieves high accuracy even though it has no predictive power). Following Powers (2011), the Matthews correlation coefficient (MCC) is considered to be robust against class imbalances. It measures the correlation coefficient between the observed and predicted binary classification and returns values between -1 and +1. A value of +1 indicates a perfect prediction, while a value of 0 is a random prediction and a value of -1 indicates a total disagreement between prediction and observation. Thereby, the MCC takes into account the true and false positives and negatives. For a given model with fixed predictive power, the MCC can be optimized by minimizing the product of false positives and false negatives. An MCC-optimized model will either have a low false-positive and high false-negative rate or vice versa. The respective cost for the errors determines the optimal trade-off. In settings with high costs for non-detected defects (e.g., product recalls) the number of false negatives will be minimized, in settings with high costs for wrong alerts (e.g., complex quality control) the number of false positives will be minimized. The ability to adapt to the cost structure of the underlying problem is an additional benefit of the MCC metric in the manufacturing context.

Feature Engineering

Feature engineering describes the process of aggregating information in the data or adding even new information by using domain knowledge. This is an important creative step before useful patterns can be discovered (Dhar 2013). Domingos (2012) highlights that this phase is critical to ensure the success of any data mining project. Similarly, Halevy, Norvig, and Pereira (2009) emphasize that good features enable a simple model to outperform a more complex model significantly. Hence, feature engineering should be performed thoroughly even though it can be a long-winded time-consuming task. Features can be derived from the underlying process structure as well as the data structure through automatic feature extraction via variable ranking procedures, manual feature construction by domain experts and mixtures of the two. Going beyond basic raw features requires a significant portion of business and process understanding as well as creativity.

3.2.3 Business Process Mining

Today, modern manufacturing systems store large amounts of sensor readings and events in some structured form. Following Van Der Aalst et al. (2011), process mining is another way to restore the inherent process knowledge in these data sets. These algorithms are used to identify patterns as well as outliers in the data and restore the different process flows. Aalst et al. (2007) successfully apply this technique in an industrial application. Schwegmann, Matzner, and Janiesch (2013) design a predictive analytics tool combining business intelligence and real-time process monitoring for a maintenance application scenario. By following an event-driven approach, this tool can reduce the lag between event observation and the decision-maker's response. Breuker et al. (2016) integrate process-mining and predictive modeling techniques to streamline operational business processes. Process-mining reveals business process models from historical transaction data. Subsequently, predictive analytics approaches facilitate the prediction of the future behavior of currently running process instances. They illustrate how this approach can be used to monitor the likelihood of negative events or detect fraudulent behavior in real-time.

3.3 Data Science Toolbox

As shown above, the process of successfully implementing a predictive information system in a manufacturing system is a complex task. Mayor challenges such as the acquisition of relevant data, data pre-processing, algorithms selection, and result interpretation are identified by Wuest et al. (2016). Therefore, we propose a five-step approach based on the guidelines for applying big data analytics put forward by Müller et al. (2016).

1. The analysis process starts with the data collection phase. Here, structured data from various legacy systems such as enterprise resource planning and production planning systems as well as unstructured data from various sources such as sensors is collected and aggregated (Chen, Chiang, and Storey 2012).

2. Subsequently, exploratory data analysis (EDA) is performed on the raw data. Here, the properties of the data set should be identified. Thereby, low-information features should be identified and treated accordingly using unsupervised learning methods. Additionally, the structures of the underlying processes can be retrieved using business process mining.

3. After data exploration, an appropriate evaluation metric for the problem at hand has to be selected. This metric should account for the data-properties identified during the EDA such as class imbalances.

4. Now, a suitable learning algorithm has to be chosen. This selection depends on the task at hand (e.g., regression vs. classification) as well as the number of observations and the dimensionality of the data.

5. Next, new features have to be derived from the results of the EDA process. These features have to be evaluated iteratively by training new models. Subsequently, a hyper-parameter optimization of the supervised learning algorithm has to be performed to increase the predictive power of the model further. Finally, the results of the analysis have to be interpreted and verified. This phase is of special importance if black-box models are applied.

Our data science toolbox is visualized in Figure 3.2.

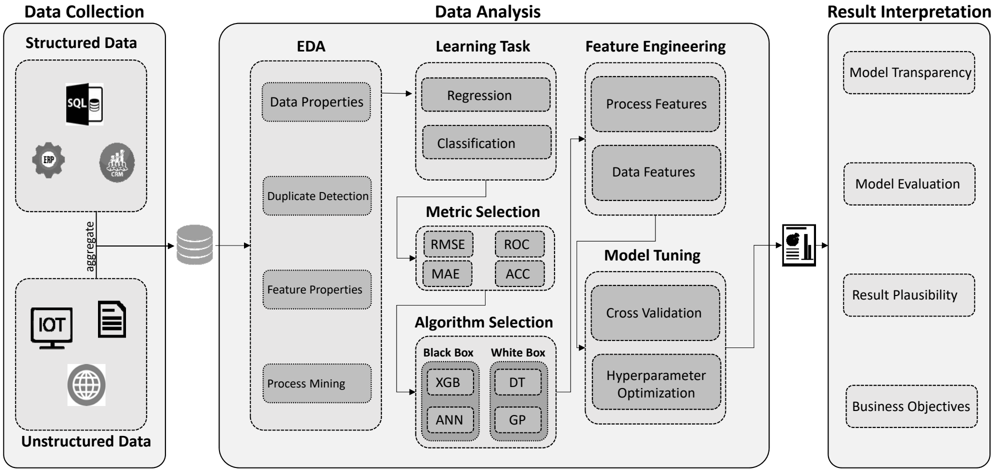


Figure 3.2: Data science toolbox

3.4 Case Study

Bosch, one of the world's leading manufacturing companies, hosted a data science competition on Kaggle, the leading crowd-sourcing platforms for predictive modeling (Kaggle.com 2016). This competition features a very large data set with anonymized measurements of production jobs moving through different manufacturing lines and stations. In addition to the measurements, the result of an ex-post quality control process is provided. To generate business value from the available data, participants were challenged to predict the defectiveness of individual production jobs.[8] We want to address this prediction task as a data science study applying our proposed toolbox.

3.4.1 Exploratory Analysis

At first, we perform an exploratory data analysis to identify the properties of the data set at hand. To this end, we describe the data set, identify correlations and low-information features, and the underlying process structure.

[8] The competition has spurred ample contributions by competition participants as well as academic publications (Pavlyshenko 2016; Maurya 2016; Mangal and Kumar 2016; Stein and Flath 2017).

Data Set Properties

The data set was collected in a manufacturing environment. It comprises a total of roughly 2.4 million manufacturing jobs. Each job has a unique id and 4,264 anonymized features. These features can be split into 968 numeric, 2,140 categorical and 1,156 time variables that are measured along 52 stations on four different manufacturing lines. Due to data anonymization, no information on the meaning of the numeric and categorical variables is available, and only the manufacturing line and the station of the feature recording can be retrieved from the variable name. The time variables indicate when each measurement was taken. To ensure the generalizability of the predictive algorithms, the data set is split equally into a training and a test data set. Jobs in the training set are labeled with $\mathit{response} = 1$ for products failing quality control and 0 otherwise. In the validation set, no information on product quality is provided as the response variable is to be predicted. Process quality is very high: Failures only occur in 0.58% of the cases while 99.42% of the observed jobs pass quality control.

Duplicate Detection

Identifying duplicates is the first step to reduce the number of features. However, column-wise comparisons are computationally expensive and not feasible due to the size of the data set. Hence, we use digest hashing for data de-duplication. To this end, a 32-bit hash is calculated for each column. Subsequently, duplicate features can be identified and removed by a fast pairwise comparison of the hashes. We find that the timestamp variables are recorded for some of the features on a station at the same time. Hence, 1,030 of the timestamps are redundant and can be removed.

Feature Properties

A first analysis shows that there are hardly any linear correlations between the response variable and the numerical variable present in the data set. Additionally, the features with the highest correlation coefficients are missing for many observations. These findings are illustrated in Figure 3.3.

The low correlation besides the high number of variables suggests that many features have a low-information value. To substantiate this

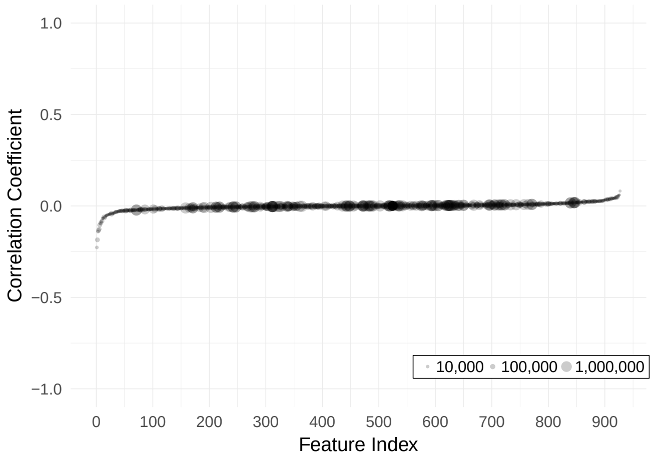
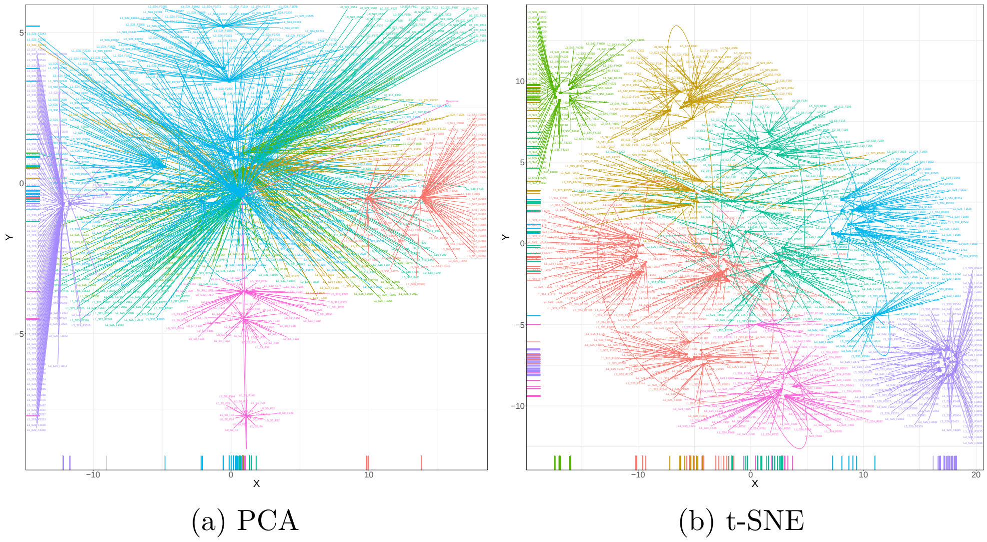


Figure 3.3: Ordered linear correlation coefficients between response variable and numerical variables (*Node size indicates the number of observations with this feature*)

assumption, we perform a PCA as well as a t-SNE. The results visualized in Figure 3.4 show, that t-SNE is able to perform a much better split on the high dimensional data at hand. Additionally, it becomes evident that many variables are holding similar information. Keeping this finding in mind, we will be able to perform feature reduction steps in later steps of the modeling process.



(a) PCA

(b) t-SNE

Figure 3.4: Dimension reduction approaches for identifying feature similarity

Process Structures

To obtain a deeper understanding of the underlying processes, we apply process mining to identify relevant patterns. Following Van Der Aalst et al. (2011), this approach can help reveal a process model without any a-priori information. This is especially valuable for the anonymized data set at hand. To proceed, we filter the individual job data for non-empty features to identify the stations that each job passes through. Subsequently, the stations are ordered by ascending time to create a network representation of the jobs. Figure 3.5 shows the production network from different perspectives.

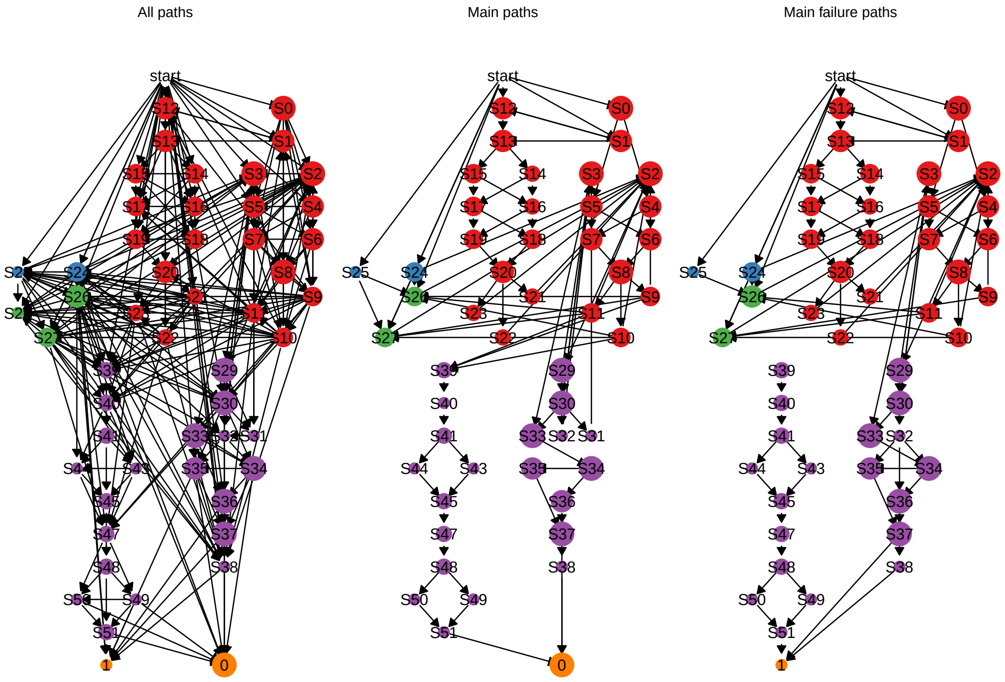


Figure 3.5: Shopfloor process visualization and illustration of predictive process patterns (*node colors indicate the line, node size indicates usage frequency of the given station*)

First, the complete graph with all occurring edges is visualized. Most parts follow a sequential path through two of the four production lines before they are classified as defective or non-defective. Next, the paths that only occur sporadically are removed by filtering for edges with a frequency exceeding the first quartile. Given the base failure-rate of 0.58%, the remaining main paths through the manufacturing network all lead to a non-defect classification. Some of the stations perform parallel operations (e.g., S14→S18) while others have to be visited in sequential order (e.g., S12→S13). The last graph visualizes the main process paths resulting in defective products.

3.4.2 Modeling and Feature Reduction

Given the insights concerning feature similarity, we first seek to reduce the data set by identifying and removing non-informative variables. To identify non-informative variables, we train separate boosting models using either the numerical or the categorical features. To reduce the computational load and speed up the process, the training is performed on samples of 200,000 rows. Subsequently, the importance of the features is determined by calculating the Kullback-Leibler divergence, also referred to as *information gain* in the machine learning context (Friedman, Hastie, and Tibshirani 2001). The sum of the information gains for all features always equals one. Therefore, this metric evaluates relative variable predictiveness as opposed to offering an absolute value. Figure 3.6 summarizes the cumulative information gains for the numeric and the categorical data. It becomes obvious that a relatively small set of features carries the bulk of the relevant information while the biggest part can be considered noise. In the case of the numeric variables, most information is captured by a subset of only 150 of the 968 features with about 80% captured by the first 50 features. Even more dramatically, out of the 2,140 categorical variables, all information gain is captured by only 27 variables with about 80% being condensed in only one variable. We remove all variables without information gain and reduce the number of variables to 150 numerical and 27 categorical features.

The reduced data size allows us to run a lightweight *R* implementation with extreme gradient boosting (XGB) developed by Chen and Guestrin (2016). XGB is a state-of-the-art gradient boosting implementation, offering superior speed by exploiting the sparsity of feature matrices. This more efficient implementation allows training of models with thousands of boosting iterations within less than a day facilitating efficient hyper-parameter optimization. Furthermore, XGB facilitates direct integration of custom

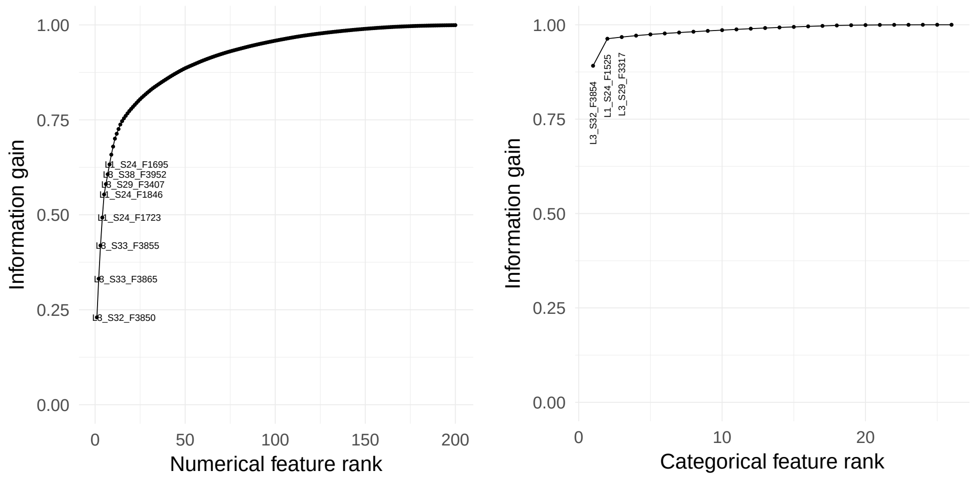


Figure 3.6: Relative information gain of numerical and categorical features

evaluation metrics instead of standard metrics. Consequently, we directly modeled the MCC score as the base for the machine learning algorithm. In combination with the removal of duplicated features, the new model realized an MCC score of 0.24 corresponding to a 9% increase in predictive power.

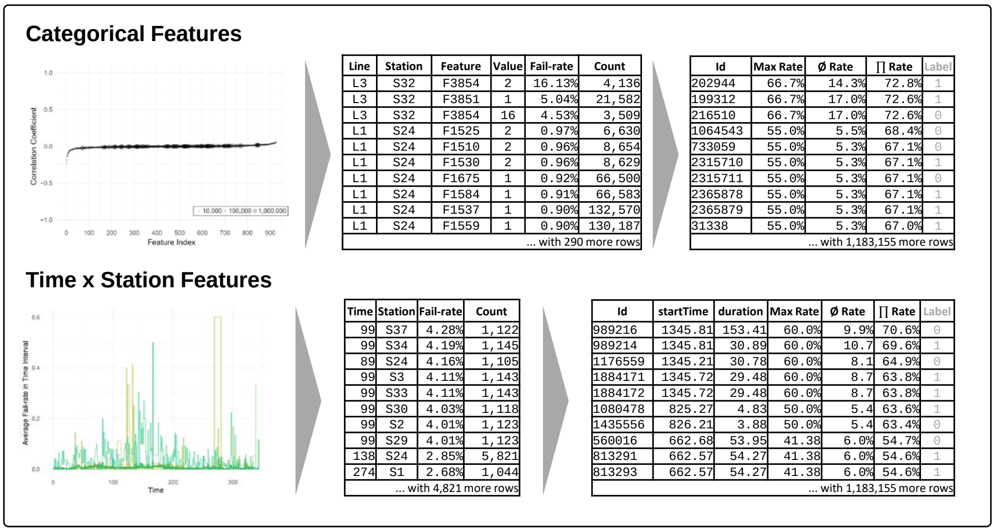
3.4.3 Feature Engineering

So far, our analysis only relied on features provided in the raw data set. To further increase the quality of the model, new features have to be developed. Going forward, we first retrieve the process structure from the anonymized data set. This information is used to refine the predictive model iteratively. To this end, we aggregate the existing raw variables to more powerful features by modeling system failure rates and approximating individual production lots.

Failure Rate Features

We can leverage the manufacturing process flows to develop stronger features for the machine learning approach. In particular, we are interested in combining multiple individually weak features into strong combined features.

While individual categorical features are fairly non-predictive (Figure 3.3), we can condense their joint information content by means of aggregating approaches akin to Hauser et al. (2015). To this end, we determine failure-rates for any given realization of the different categorical variables (including the frequent absence of a variable signified by an "NA" coding). We find that defect-rates are significantly increased for some (possibly seldom) categorical variable values. For example, jobs featuring the value "2" for feature "F3854" have a failure-rate of 16.13% compared to the base rate of 0.58%. Using path-wise aggregation along with the process flow, we can derive meta-features from the individual defect-rates FR_i. We apply three different aggregation schemes, namely the maximum failure-rate $\max_i FR_i$, the mean failure rate $\frac{1}{|i|}\sum_i FR_i$ as well as the compound rate $\Pi_i(1 - FR_i)$. This aggregation approach is illustrated in the top panel of Figure 3.7. The table in the top-right illustrates that the meta-features exhibit a much higher correlation with the target label than the original set of unprocessed categorical features. This suggests that the meta-features succeed in distilling the information content from the raw feature realizations.



Categorical Features

Line	Station	Feature	Value	Fail-rate	Count
L3	S32	F3854	2	16.13%	4,136
L3	S32	F3851	1	5.04%	21,582
L3	S32	F3854	16	4.53%	3,509
L1	S24	F1525	2	0.97%	6,630
L1	S24	F1510	2	0.96%	8,654
L1	S24	F1530	2	0.96%	8,629
L1	S24	F1675	1	0.92%	66,500
L1	S24	F1584	1	0.91%	66,583
L1	S24	F1537	1	0.90%	132,570
L1	S24	F1559	1	0.90%	130,187
... with 290 more rows					

Id	Max Rate	Ø Rate	∏ Rate	Label
202944	66.7%	14.3%	72.8%	1
199312	66.7%	17.0%	72.6%	1
216510	66.7%	17.0%	72.6%	0
1064543	55.0%	5.5%	68.4%	0
733059	55.0%	5.3%	67.1%	0
2315710	55.0%	5.3%	67.1%	1
2315711	55.0%	5.3%	67.1%	0
2365878	55.0%	5.3%	67.1%	1
2365879	55.0%	5.3%	67.1%	1
31338	55.0%	5.3%	67.0%	1
... with 1,183,155 more rows				

Time x Station Features

Time	Station	Fail-rate	Count
99	S37	4.28%	1,122
99	S34	4.19%	1,145
89	S24	4.16%	1,105
99	S3	4.11%	1,143
99	S33	4.11%	1,143
99	S30	4.03%	1,118
99	S2	4.01%	1,123
99	S29	4.01%	1,123
138	S24	2.85%	5,821
274	S1	2.68%	1,044
... with 4,821 more rows			

Id	startTime	duration	Max Rate	Ø Rate	∏ Rate	Label
989216	1345.81	153.41	60.0%	9.9%	70.6%	0
989214	1345.81	30.89	60.0%	10.7	69.6%	1
1176559	1345.21	30.78	60.0%	8.1	64.9%	0
1884171	1345.72	29.48	60.0%	8.7	63.8%	1
1884172	1345.72	29.48	60.0%	8.7	63.8%	1
1080478	825.27	4.83	50.0%	5.4	63.6%	1
1435556	826.21	3.88	50.0%	5.4	63.4%	0
560016	662.68	53.95	41.38	6.0%	54.7%	0
813291	662.57	54.27	41.38	6.0%	54.6%	1
813293	662.57	54.27	41.38	6.0%	54.6%	1
... with 1,183,155 more rows						

Figure 3.7: Failure rate feature generation through aggregation of categorical and temporal features

We apply an analog procedure to capture temporal and station-level failure behavior (lower panel of Figure 3.7). Item fail rates vary depending on the time they went through a given station. This may be due to machine wear, operator fatigue, material problems, or other external influences. Direct encoding of station-time stamp pairs would again yield an enormous number of weak features. By determining station-level fail rates with subsequent path-wise aggregation, we can again create condensed and predictive meta-features. These features are complementary to the categorical fail rates and combined with the base model boost predictive power to an MCC value of 0.28.

Manufacturing Batch Features

Sequence dependencies are commonplace in manufacturing settings due to the grouping of production jobs into batches or production lots. Consequently, failure information on individual items from a lot may be relevant for failure detection for other lot members.

To approximate batches in the data set, we follow two avenues: First, we use the timestamps to approximate individual production lots in the data. To this end, the data set is ordered by the start time and the time difference between two subsequent parts is calculated. Small differences suggest that two parts are part of the same lot, while bigger differences indicate different lots. A more focused approach relies on the assumption of the data set Id column not being random but actually revealing information on the underlying process. By filtering the data to only feature pairs of subsequent Ids (approximately half of the data), we can analyze this hypothesis. Table 3.1 presents the results of this sequence-level analysis.

Table 3.1: Failure rates dependent on previous observation

Current label	**Probability of subsequent "1"**	**Count**
"0"	0.55% $\approx$ base-rate	6,479
"1"	5.79% $\gg$ base-rate	398

(a) Jobs sorted by start time

Current label	**Probability of subsequent "1"**	**Count**
"0"	0.52% $\approx$ base-rate	3,055
"1"	10.08% $\geqslant$ base-rate	345

(b) Subsequent Ids sorted by Id

Both approaches reveal significantly increased fail-rates of the subsequent job if the current job is labeled defective—a 10-fold increase in the coarse start time approach and a 20-fold increase with the Id-approach. Incorporating this additional information in the form of lagged feature variables dramatically improves the model's predictive performance: A minimal model with raw variables and the sequence features yields an MCC score of 0.36, the combination of sequence features and previously developed features achieves an MCC of 0.44.

We also tried to incorporate more distant pairings besides direct sequences. However, these additional sequence feature did not improve model performance but rather deteriorated predictive power.

Data Anomaly Features

Going beyond the more natural information sources offered by process and measurement data, a more untypical source of predictive features are data anomalies. In the data set a hand, an initial screening had highlighted the presence of duplicate entries exhibiting identical numerical feature values despite having different Ids. Such row duplicates can arise in manufacturing systems in the context of communication crashes. SCADA systems will usually repeat the last seen value, so the measurements associated with a given sequence of part numbers correspond to the last correctly received, until the communication is recovered. If such communication failures are triggered by external events (such as power outages), they may also affect the quality of currently manufactured parts. To explore this hypothesis, we created row-wise hashes across all numerical features to efficiently detect duplicate rows in the large data set (Elmagarmid, Ipeirotis, and Verykios 2007).[9] This thorough search for duplicates confirmed the initial observation of anomalous rows. In total there were 90,000 duplicate rows present in the data. Even more surprising, 3,293 of all 6,879 defective jobs originated from the duplicate data. In turn, the non-duplicate data subset has a corrected fail-rate of 0.33% while the duplicate subset has an eleven times higher fail-rate of 3.63% (see Table 3.3). The inclusion of the duplicate feature enhanced the predictive performance of our model to reach an MCC score of 0.47.

3.4.4 Result Discussion and Interpretation

In the previous sections, we developed a failure-detection system for a manufacturing process. We removed non-essential features and extracted information on the manufacturing process through process mining. This

[9] The hash creation for the full data set took about 40 minutes.

Table 3.3: Fail-rate of duplicated vs. non-duplicated data rows

Duplicate row	Fail-rate	Count	Number of Failures
FALSE	0.33%	1,092,975	3,586
TRUE	3.63%	90,772	3,293

enabled us to determine failure rates on a station and time level and to identify manufacturing batches. Meta-features derived from these failure rates and the batches further increase the predictive power of our model significantly. In the last step, data anomalies occurring in manufacturing systems are identified. Utilizing this feature boosts the system's MCC to 0.47. Recognizing the incremental nature of the individual improvement steps, it becomes evident that successful predictive modeling is not a one-shot endeavor but rather necessitates diligent and persistent development.

Answering the problem of machine learning opacity raised by Burrell (2016), we recombine the trees determined by our gradient boosting machine to one aggregated tree. To this end, we make use of the fact that all 7,000 binary trees of the final model have the same depth and therefore, the same number of nodes. Consequently, each node has 7,000 representations. We can determine the importance of a feature by counting how often it appears on a particular node. Figure 3.8 visualizes the aggregated tree with the three most frequent features at each node. As in standard decision trees, variables occurring earlier in the tree are more important than variables appearing at the end. To this end, the value of the engineered features becomes evident. The defect-rates on a machine level as well as the production lot approximation emerge as highly predictive while the raw features show up deeper in the tree.

Furthermore, it becomes evident that black-box machine learning models and process mining approaches can work in unison. For instance, the boosted trees identify station 33 as a possible weak point with the feature recorded on this station ("L3-S33-F3857"). Looking at the shop-floor process structure (Figure 3.5) confirms this station's central role in the manufacturing process.

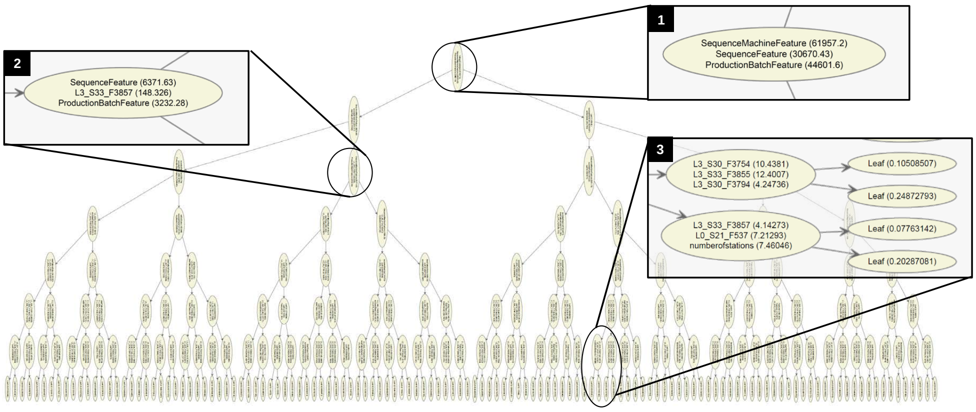


Figure 3.8: Retreiving system insights from black box models

3.5 Conclusion and Outlook

We develop and explicate a data science toolbox for manufacturing decision-makers. We showcased the application of this toolbox using a large data set from a major manufacturing company. In particular, we illustrate the development, refinement, and evaluation of a predictive analytics system. Although we showcased the implementation with a data set from a specific manufacturing process, we are confident that the approach is generic and straight-forward to transfer and implement for other use cases as the data was anonymized and no additional information about the underlying process was available.

In the age of big data, researchers, as well as practitioners, can no longer rely exclusively on standard statistical methods (e.g., linear regression) to generate business insights from large data sets. Instead, the use of machine learning becomes inevitable as these approaches are better suited to handle thousands of variables or work with unstructured data. Breiman (2001b) and Shmueli et al. (2010) show that these approaches are of particular importance in studies aiming at prediction instead of description. The main advantage of state-of-the-art machine learning algorithms is that they make less statistical assumptions and can work with data sets of very high dimensionality. Additionally, these methods cannot only capture non-linear relationships but also pick up higher-order interaction effects between variables. On the downside, these black-box algorithms (e.g., gradient boosting machines) typically generate incomprehensible models and rules.

Yet, the interpretability of the rules used by the algorithms is important if subsequent actions based on the predictions are to be taken by human decision-makers (Martens and Provost 2014; Diakopoulos 2014).

Our findings regarding the application of big data analytics are twofold. It becomes evident that simply plunging a vast amount of data into “smart” algorithms is not the silver bullet a lot of researchers and practitioners expect it to be. Instead, we show that constant improvement, feature engineering, and consolidation will complementarily improve the predictive power of a business analytics system. In order to further increase the predictive power, higher-level modeling approaches could be applied. A first step would be the training of two distinct models for the duplicates and non-duplicates identified during the anomaly detection. Going further, a set of different black-box models should be trained and combined to generate predictions from stacked predictors. Such an ensemble would come at the cost of interpretability and necessitate new methods to answer the need for comprehensibility. The increasing complexity in the data and the successful combination of process mining and machine learning emphasize the need for analytic skills as well as business understanding and showcases the comparative advantage of industrial analytics as a cross-disciplinary application of machine learning (Wuest et al. 2016).

4 Sensor-Based Decision-Support for Manual Processes

Analytics applications are becoming indispensable in today's business landscape. Greater data availability from self-monitoring production equipment allows firms to empower individual workers on the shop-floor with powerful decision-support solutions. To explore the potential of such solutions, we replicate an important manual leak detection process from high-tech composite manufacturing and augment the system with highly sensitive sensors. Based on this setup, we illustrate the main steps and major challenges in developing and instantiating a predictive decision support system. By establishing a scalable and generic feature generation approach as well as leveraging techniques from statistical learning, we are able to improve the forecasts of the leak position by almost 90%. Recognizing that mere forecast information cannot be evaluated with respect to business value, we subsequently embed the problem in an analysis of the underlying searcher path problem. We compare predictive and prescriptive search policies against simple benchmark rules. The data-supported policies dramatically reduce the median as well as the variability of the search time. Based on these findings we posit that prescriptive analytics can and should play a greater role in assisting manual labor in manufacturing environments.[10]

[10] This paper is published in the *Journal of Business Economics* (Stein, Meller, and Flath 2018).

4.1 Introduction

Despite a shift in jobs and capital towards the service industry, the manufacturing sector still serves as a critical backbone of many leading economies around the globe. Recently, this sector has seen a tremendous digital transformation: Due to the introduction of “smart”, i.e., self-monitoring production equipment paired with significant cost decreases for sensors and data storage, modern manufacturing companies have access to a considerable pool of process data. For the year 2010, Manyika et al. (2011) estimated that the manufacturing sector generated two exabytes of data (e.g., production status, equipment utilization, machinery condition). The combination of data abundance on the one hand and new analysis tools on the other provides opportunities to foster data-driven decision making helping to increase both efficiency and effectiveness of existing processes.

The widespread use of analytics applications in managerial decision making is limited by data availability. For this reason, data-driven decision support has initially been exclusively used for problems on the organizational level. This is because these problems primarily rely on aggregated data, which is easily accessed through standard enterprise resource planning systems. With increasing availability of data tailored to the needs of specific departments within the company, advanced analytics solutions were introduced on the departmental level. Yet, individual workplace decision support is still limited due to the lack of fine-grained data, especially in manufacturing. We put forward a taxonomy of data-driven analytics based on the functional level where the tools are applied. This categorization complements the recent descriptive-predictive-prescriptive framework suggested by Lustig et al. (2010).

Recognizing a lack of contributions on individual decision support in manufacturing industries, we then explore this novel application area for advanced analytics through a case study. Specifically, we illustrate the development and evaluation of a sensor-based decision support tool for a manual leak search process in the aerospace industry. We split this task into two distinct research tasks which address the following research questions:

RQ3.1 How much can the accuracy of leak localization be improved by using high-resolution sensor data and predictive analytics?

RQ3.2 To what extent can better location predictions be translated into an economic benefit as measured by the labor cost of searching?

To address the questions, we first develop a *predictive* framework for localizing the leak position based on sensor data. Subsequently, we complement this positional information with *prescriptive* decision support informing the actual search policy. Using these two components, we can quantify the value of leveraging sensor data in manual shop-floor processes.

4.2 Data-Driven Decision Support

With greater data availability, the desire to extract insights and business value is ever-growing. As a consequence, we note a steady increase in data-driven decision support systems that also affect the way managerial decisions are prepared nowadays. Serving as an objective grounding for informed actions, such systems can be employed in a wide range of different planning situations. Analytics is an umbrella term for activities that guide decision-making processes by means of analyzing business and process data and in turn deriving functional insights.

4.2.1 A Granularity-Oriented Taxonomy for Advanced Analytics

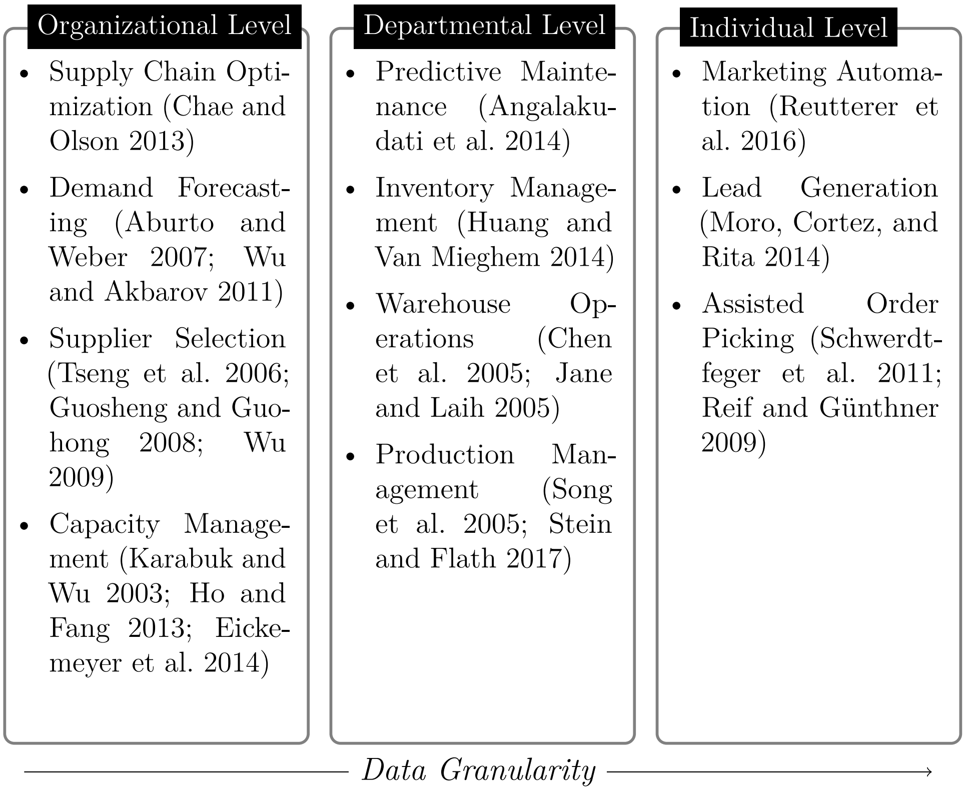
A widely adopted classification of analytics is the distinction between descriptive, predictive, and prescriptive approaches based upon the reach and methodological scope of the particular application (Lustig et al. 2010).

This “classic” split promotes a view on applications from a foremost methodological point of view, i.e., according to this taxonomy, an analytics solution would be categorized depending on whether the final decision maker receives a descriptive analysis, a prediction of future observations or a prescribed action as an outcome to solve his planning problem.

While acknowledging the simplicity and intuitiveness of this well-established taxonomy, we posit that particularly for business practitioners as well as researchers from the management science domain, a taxonomy with a stronger scope on the actual planning problem to solve would provide valuable insights. We argue that the requirements and conditions to provide decision support for particular planning problems on different functional levels differ considerably. As an example, despite being both classified as *prescriptive analytics*, it is undoubted that considerable differences exist between an analytics application for a warehouse location problem and an assisted order picking system. While the former involves a planning horizon of several years in advance and is hence confronted with high uncertainty in terms of future demand patterns, the latter is a stationary optimization problem with all relevant information available to provide a deterministic, optimal solution. For this reason, we bring forward a new taxonomy that complements the classic view with a more decision problem-oriented, functional scope. A closer look at the recent literature suggests three different planning levels on which companies employ such data-driven decision support systems.

On the most aggregate level, firms use data analytics to support decisions on the organizational or inter-organizational level. These strategic decisions are taken a very long time in advance, and hence, relevant data is considered only on a very aggregate level. Decision support systems for this setting have been developed over decades within the operations management community. Example applications in an industrial context can be found as solutions to classic problems such as supplier selection or network planning.

In recent years, companies tremendously grew their capabilities in collecting and storing their internal transaction-level and process data. With the availability of such detailed and near-real-time data, analytics can also be employed for a different set of problems. Most of the recently developed solutions provide decision support on a departmental level helping mid-level management to optimize the operational planning for one or more teams. Representative tasks include production scheduling, maintenance management, or warehouse operations with a tactical planning horizon. Going forward, increasing amounts of sensor and real-time transaction data will enable decision support to not only guide planning tasks but to optimize the operational work of individual employees, e.g., optimize the execution of single manual tasks in production environments. Figure 4.1 provides an overview of this classification scheme.



Organizational Level

- Supply Chain Optimization (Chae and Olson 2013)
- Demand Forecasting (Aburto and Weber 2007; Wu and Akbarov 2011)
- Supplier Selection (Tseng et al. 2006; Guosheng and Guohong 2008; Wu 2009)
- Capacity Management (Karabuk and Wu 2003; Ho and Fang 2013; Eickemeyer et al. 2014)

Departmental Level

- Predictive Maintenance (Angalakudati et al. 2014)
- Inventory Management (Huang and Van Mieghem 2014)
- Warehouse Operations (Chen et al. 2005; Jane and Laih 2005)
- Production Management (Song et al. 2005; Stein and Flath 2017)

Individual Level

- Marketing Automation (Reutterer et al. 2016)
- Lead Generation (Moro, Cortez, and Rita 2014)
- Assisted Order Picking (Schwerdtfeger et al. 2011; Reif and Günthner 2009)

Figure 4.1: Taxonomy of data-driven decision support

4.2.2 Analytics for Organizational-Level Decision Problems

Supporting managerial decision making through mathematical modeling has been the goal of operations research for a long time. To illustrate this, we look at analytics solutions for organizational decision making in forecasting, supplier selection, and capacity management.

In the context of supply chain management, accurate forecasting of demand plays a major role in the efficient management of operations. As the knowledge about end consumer demand is distorted due to asymmetric information, manufacturers typically face substantial variations in demand and hence have to plan their production based on highly inaccurate demand forecasts. To improve forecasting performance facing such distorted information, Carbonneau, Laframboise, and Vahidov (2008) compare different advanced analytics approaches with traditional forecasting methods. They find that these new machine learning models excel in terms of performance.

To better capture time-series information in comparison to a pure machine learning-based approach, Aburto and Weber (2007) propose a hybrid system which combines Auto-regressive Integrated Moving Average models with neural networks to forecast demand. In the context of manufacturing/warranty providers, Wu and Akbarov (2011) use a system based on support vector regression to better forecast warranty claims.

Several authors (Aviv 2007; Cui et al. 2015) argue that by combining information from different players along the supply chain, overall performance can be improved. To integrate data from these complementing sources, Chae and Olson (2013) propose a framework for a business analytics solution. They identify three central capabilities of such a tool, namely data management, analytical processing, and supply chain performance management.

Different application areas are purchasing and sourcing. Tseng et al. (2006) develop a machine learning approach to guide this process. They test their model in a supplier selection setting and derive a set of decision rules and a preferred supplier prediction model. They argue that supply chain experts would be able to apply those decision rules and hence improve the supplier selection process. Guosheng and Guohong (2008) employ a similar approach based on support vector machines to evaluate and select suppliers based on a predicted score and compare their performance to a different selection method. Wu (2009) proposes a hybrid model consisting of data envelopment analysis, decision trees, and neural networks to assess supplier performance.

Another important planning problem affecting the organization is the determination of optimal capacity levels. Karabuk and Wu (2003) introduce an approach for capacity planning in the semiconductor industry. They consider uncertainty about both demand and capacity, as well as strategic capacity decisions and operational capacity adjustment options through a recourse variable. Their model includes demand forecasting as well as proactive market development strategies to maximize revenue. Ho and Fang (2013) focus on the capacity allocation problem for a manufacturing system that produces multiple products. They provide a mathematical model solving this problem based on marginal profit, inventory holding and shortage cost, loss of excess production, and market demands. Eickemeyer et al. (2014) show that data-driven models can support organization-wide capacity management decisions. Their approach builds upon a data fusion strategy with the help of Bayesian networks to integrate available information into the model as soon as available. They evaluate their solution with empirical data.

4.2.3 Analytics for Departmental-Level Decision Problems

According to the literature review of Choudhary, Harding, and Tiwari (2008), analytics applications can contribute to various enterprise functions such as design, logistics, production, and marketing.

Amongst these, a particularly prominent area is maintenance management. The idea of monitoring machines to assess and predict their degradation until a failure occurs has been present for decades (e.g., Létourneau, Famili, and Matwin 1999; Grall, Bérenguer, and Dieulle 2002; Dieulle et al. 2003). However, only with the widely available sensory information generated by modern production equipment, predictive maintenance systems that can link this real-time condition monitoring information with degradation information are being established. This development provides planners with an improved forecast of maintenance demand and allows optimized scheduling of maintenance crews. Raheja et al. (2006) propose a predictive maintenance system architecture for such a prescriptive solution in maintenance management. In a gas pipeline maintenance context, Angalakudati et al. (2014) develop a prescriptive maintenance management solution to schedule maintenance activities and allocate maintenance resources to these events for a large gas utility. Stein and Flath (2017) apply advanced analytics to predict manufacturing failures in a production system.

The efficient use of available capacity is another carry-over from the organizational level. Chen et al. (2005) analyze the incoming order process to derive decision rules for efficient order batching. Concerning inventory management, the analysis of available data related to the actual demand provides an enormous potential for cost reductions. Jane and Laih (2005) show how cluster algorithms can be used to balance the workload among pickers in a pick-by-light system in order to reduce the time needed for fulfilling each requested order. Huang and Van Mieghem (2014) analyze click-stream data for a company featuring its products online but selling them offline. By matching the collected online data with offline purchases and integrating the information into a dynamic decision support model for their inventory management, they achieve cost reductions between 3 and 5% in specific scenarios. Another contribution to the data-driven inventory management literature is due to Beutel and Minner (2012): They propose a regression-based decision support system that directly links auxiliary data to the final inventory decision whilst balancing overage and underage costs for excess respective missing units.

In production management, analytics can be employed to find decision rules to schedule production on a single machine based on raw production data (Li and Olafsson 2005). Song et al. (2005) apply machine learning to assess the feasibility of resource usage plans in re-manufacturing settings. They show that their approach is viable in large-scale problems and enables firms to determine good plans even in very complex settings. Another sector facing complex resource conflicts is the semiconductor industry. To achieve high manufacturing performance against changing environments, appropriate dispatching rules have to be selected. Wang, Chen, and Lin (2005) show that a stack of machine learning models can be used to encounter this task. They apply a decision tree model to select the most suitable rule, while a neural network predicts the expected performance of the selected rule.

4.2.4 Analytics for Individual-Level Decision Problems

Marketing was one of the first functions to embrace analytics on the individual level. The notion of “marketing automation” summarizes decision support systems in this domain. These systems include tools assisting or automating processes on the level of an individual employee (Heimbach, Kostyra, and Hinz 2015). A prominent example are recommendation engines automating customer support processes. Reutterer et al. (2016) identify measures to analyze multi-category purchase histories and provide customer relationship agents with recommendations for targeted marketing actions in the grocery sector. Moro, Cortez, and Rita (2014) introduce analytics to predict the success of marketing calls for the banking industry based on features about the targeted customer as well as details about the proposed offer. They compare the performance of different machine learning models for the prediction and point out the importance of such a decision support tool for client selection. Logistics is another department with the potential to leverage decision support systems on an individual level. While pick-by-light systems enable more efficient warehouse operations on an individual level, additional efficiency gains can be enabled by pick-by-vision systems (Schwerdtfeger et al. 2011). Reif and Günthner (2009) show that providing workers decision support via head-mounted displays significantly reduces the required time for the picking process. However, we can note that while prescriptive analytics are commonly applied for larger planning problems on a group or departmental level, only a few approaches can be found that provide decision support on a workplace level guiding employees for a specific task. Furthermore, these decision support systems currently focus mostly on non-manufacturing supporting functions. Our work contributes to the literature by establishing and evaluating an integrated prescriptive analytics framework for manual processes in manufacturing environments.

4.3 Case Study Overview and Research Approach

We design an analytics-aided production system on the example of the vacuum resin infusion process (Williams, Summerscales, and Grove 1996). This process is of particular importance in practice as it is used in a variety of high-value manufacturing industries (e.g., automotive, aerospace, marine, infrastructure). During infusion the part molds are placed in vacuum bags, subsequently evacuated and infused into the workpiece. The final part is removed after the resin is fully cured. During the process, any leakages lead to sub-par product quality, which is why their detection is of the highest priority. Yet, leakage scanning is performed manually with ultrasonic microphones or thermal cameras. As the time required for the manual search is convex-increasing in the dimensions of the search area, the production of large components (e.g., aircraft wings) becomes extremely expensive. To assist manufacturing processes in general and improve the leakage scan process in particular, we design and evaluate an analytics-aided production system. To this end, the original approach is augmented by generic, multi-use vibration sensors collecting data during the production process. Subsequently, we apply machine learning algorithms to train predictive models based on the sensor data. The output of these models is used to derive an individual prediction for a given search task. Subsequently, an optimized searcher path in the spirit of Trummel and Weisinger (1986) is determined using a set of heuristics. This way, data is leveraged to achieve better decision making. We illustrate and evaluate our conceptual approach through a laboratory experiment replicating the leak prediction process and a simulation study accounting for varying search strategies and parameterization. To tackle the problem at hand, we apply a design-oriented research approach and position our artifact design along with the guidelines put forward by Hevner et al. (2004).

- **Problem Relevance:** In order to remain competitive, manufacturing companies are hard-pressed to improve the efficiency of manual processes. The suggested analytics-aided production system is one possible way to improve decision making on this level based on novel data sources.

- **Research Rigor:** We base our artifact on different existing models and research articles. Following Grabocka, Wistuba, and Schmidt-Thieme (2015) we apply polynomial curve fitting for the feature generation. The forecasting module deploys several well studied white-box and black-box machine learning models, while the prescriptive module showcases several policies based on the optimized searcher path (Trummel and Weisinger 1986).

- **Design as a Search Process:** To design an analytics-aided production system, we follow the “Cross Industry Standard Process for Data Mining” (Chapman et al. 2000). We first present our experimental setup and the resulting raw data. Second, we elaborate on the data preparation phase. This phase includes the cleaning of the data set, as well as the development of new predictive features. Third, we model the location problem at hand and apply various machine learning techniques to the one with the highest predictive power. Subsequently, we move forward from predictive to prescriptive analytics. To this end, we show how the available information can be operationalized in optimized search policies.

- **Design as an Artifact:** We design an IT artifact with two modules. The forecasting and the policy module are described in detail and implemented using *R*.
- **Design Evaluation:** We assess the artifact in a simulation study using real-world leakage and sensor data and different parameter scenarios.
- **Research Contribution:** The main contribution of this research is the transformation of raw sensor data into solid decision support utilizing big data analytics as well as operations management techniques. This research informs technical as well as managerial audiences. While the formal models may primarily appeal to audiences with a more technical focus, the economic results address managers and policymakers.

The over-arching structure of our study is shown in Figure 4.2.

4.4 Experimental Design and Data Understanding

We present the development and evaluation of a prototype for data-driven decision support on the individual workplace level. Our application is motivated by the vacuum resin infusion process as described above and uses sensor data to improve the detection of minor leaks within the utilized vacuum bags.

4.4.1 Experimental Design and Setup

Following Basili (1996), the quality and efficacy of a system have to be rigorously demonstrated by means of an appropriately selected evaluation

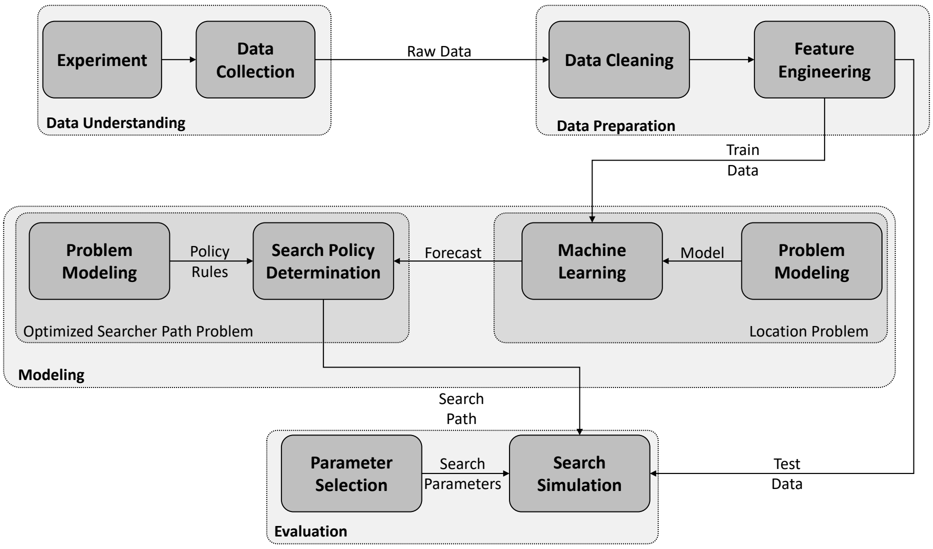


Figure 4.2: Research approach

method. Our case study relies on an experimental replication of the real-world vacuum resin infusion process to collect data and evaluate our data-driven leakage detection approach. To this end, we apply eight multi-use structure-borne noise sensors between a rectangular part mold of size $120cm \cdot 60cm$ and the vacuum bag to measure vibrations on the bag. The experimental setting is illustrated in Figure 4.3.

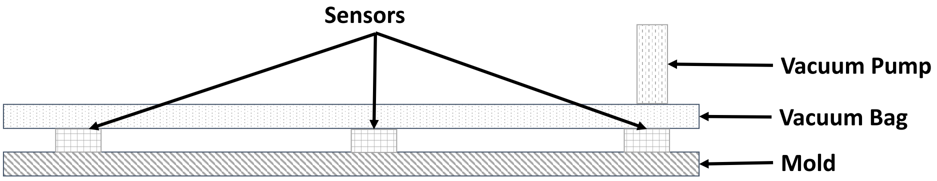


Figure 4.3: Experimental setup

Subsequently, the bag is punctured and the evacuation process is started. During the evacuation, the vibrations on different positions of the bag are measured by the sensors and the readings are recorded with a very fine granularity of 4,000 readings per second. In total, we repeat the evacuation process for 166 holes at different locations to collect the final data set. The absence of a leak is trivially detected by noticing that the bag pressure remains constant. Therefore, we did not have to consider scenarios without leaks.

4.4.2 Data Understanding

This phase involves describing and exploring the data set at hand. We rely on a lightweight SQLite (Hipp and Kennedy 2016) database to store the raw data collected from the experiment. Given the number of different sensors and their extremely high temporal resolution, the complete data table spans 12 variables with roughly 10 million rows. Each observed leak has a unique id $leak_{id}$. The sensor readings of sensor n at time t are stored in the variable s_n. Additionally, to facilitate supervised learning, the coordinates of the leak are stored as label variables. Table 4.1 summarizes the variables and the according data types.

Table 4.1: Data properties

Variable	**Data Type**	**Description**
$leak_{id}$	int	Unique observation id for each leak
s_n	num	Reading of Sensor n. One variable for each sensor
x	num	x-Axis leak coordinate
y	num	y-Axis leak coordinate
t	time	Time (nano seconds) of the sensor reads in variables s_n

In order to get a better picture of the data set at hand, examples of the raw data and the corresponding process steps are shown in Figure 4.4. The y-axis shows the readings of the sensors. A value of 0 indicates no vibrations while the value -5 is the maximum vibration level that can be recorded by the sensor. Values outside of these boundaries can be considered noise as the sensors in use can not detect them. We see from the figure above that there are no vibrations when the recording of the sensor data is started. Vibrations peak as soon as the evacuation process begins and come back to halt during the process. The time needed for each sensor to react to the evacuation depends on the location of the observed leak.

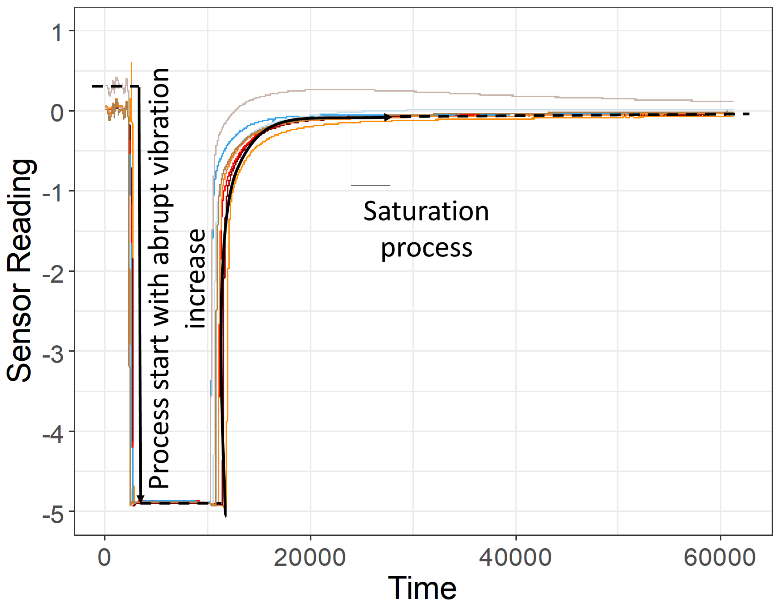


Figure 4.4: Exemplary sensor readings

4.4.3 Data Preparation

In this phase, the final data for the development of the machine learning models is compiled. The expected quality of the subsequent modeling efforts is constrained by the quality of the data preparation (Zhang, Zhang, and Yang 2003). Extracting the relevant information from the sensor reads is a non-trivial task due to the data volume and the inherent noise. Because of its complexity, on the one hand, and its inherent importance on the other, data preparation regularly requires the greatest efforts within a data mining project (Yan, Zhang, and Zhang 2003). Against this backdrop, we follow a two-stage approach.

We initially focus on data pre-processing tasks to handle technical issues such as missing or noisy data (Yu, Wang, and Lai 2006). The reasons for these issues lie in the experiment execution. Due to the highly accurate sensors, even minor disturbances of the experiment setup can result in significant noise in the readings. During the experiment, we experienced problems with a change of the room temperature due to bright sunshine, which corrupted the data. By visualizing the data, we identified the corresponding 33 invalid observations and removed them from the data set. As we want to observe the evacuation process, the sensor readings recorded before the vacuum pump is started need to be removed. To this end, we filter and remove all data which is recorded before all sensors reach their minimum level. Subsequently, the time variable t is normalized to ensure comparability across experiment runs.

The second stage of the data preparation phase is the data transformation or feature engineering stage. Following Domingos (2012), this is another decisive success factor of any machine learning project. In the problem at hand, each sensor reading can be viewed as a potential feature. However, given the massive number of readings (compared with the number of trials), we need to condense the available information into a compact feature set. This aggregation step has to identify the characteristic sensor profiles for every sensor and each leak. These profiles describe the curves in terms of a limited set of variables. Note that the idea is not to replicate the curve in the spirit of curve-fitting but rather to summarize it with a limited but not too limited number of features. Consequently, fitting logarithmic or square root functions is of limited help as there are too few free parameters. Instead, we follow Grabocka, Wistuba, and Schmidt-Thieme (2015) who demonstrate that polynomial curve fitting is a balanced and at the same time computationally efficient approach to extract information from time series with high dimensionality. The time series of the sensor readings s_n is defined as the sum of the products of the coefficients β and the predictor values t (see Equation 4.1). As time series are described, the time of the sensor readings t is used as predictor value. Depending on the degree d of the polynomial, $d + 1$ coefficients $\beta \in \mathbb{R}^{d+1}$ are required.

$$\hat{s}_n = \sum_{j=0}^{d} \beta_j t^j \tag{4.1}$$

Besides computational efficiency, the proposed method offers additional benefits. On the one hand, this approach is generic and not limited to the introduced use-case. Hence, a successful implementation can be transferred to other applications with limited effort. On the other hand, polynomial curve fitting offers the possibility to adapt the number of features dynamically. Where a higher degree will achieve a better fit of the characteristic sensor profiles to the real data (Figure 4.5), this entails the risk of over-fitting the machine learning algorithms to the training data. To mitigate this risk while still tapping into all available information, the degree of the polynomial needs to be chosen in accordance with the number of observations. Due to the limited size of our study, we do not explore this design choice and fix the degree at four.

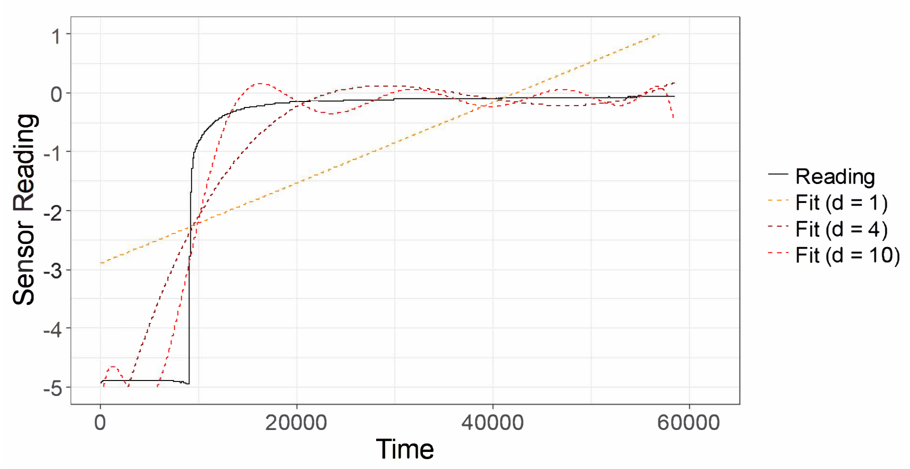


Figure 4.5: Different levels of polynomial curve fitting

The proposed method condenses the information resulting in a significant reduction of the data set. Initially, the raw data features a total number of $|T| \cdot |N|$ data points per observed leak. In contrast, the characteristic sensor profiles describe each sensor curve using the $d + 1$ coefficients of the polynomial. Hence, the size of the data is reduced to $(d + 1) \cdot |N|$ data points per leak. In our example with eight sensors over roughly 65,000 timestamps, this implies a reduction of over 99%.

In the following sections, we leverage this prepared train data set to develop a collection of *predictive* models that help to forecast the leak position. To this end, we apply various white-box and black-box machine learning algorithms and evaluate their performance. Subsequently, several policies are developed to determine optimized paths yielding *prescriptive* decision support for the search task.

4.5 Predictive Modeling: Locating the Leak

During the modeling phase, a predictive model is built for the problem at hand. Our goal is to determine the leak locations. To this end, we apply appropriate prediction techniques. Our data set is a case of labeled training data (leak positions are known across the training set). Hence, we can apply supervised learning techniques to forecast the leak locations.[11] As positions can be characterized by their spatial coordinates, we can further refine the localization problem as two regression tasks geared towards forecasting continuous response variables. The first model predicts the first coordinate of a leak from the features, i.e., the set of the characteristic sensor profiles. Naturally, the same features can be used to predict the second coordinate. However, in order to incorporate additional information in the model and increase the predictive power, one should additionally embed the prediction for the first coordinate (e.g., $\hat{x}$) as an additional input variable to incorporate dependencies between the coordinates. Our experimental setup features more sensors along the x-axis and we therefore first predict the x and then the y coordinate. We approach this nested regression task by first fitting white-box models which offer greater transparency with respect to the rules used to generate predictions. Subsequently, we apply black-box models, which can often help to further increase the prediction quality at the cost of reduced interpretability.

4.5.1 White-Box Models

The first white-box method we use is a multiple linear regression. This method established by Galton (1886) more than 200 years ago is probably the best studied form of statistical learning (Friedman, Hastie, and Tibshirani 2001). It rests on the assumption of a linear relationship between a set of input variables and a single output variable. The linearity assumption is

[11] The problem seems to be suited for application spatial regression techniques. However, this approach requires the availability of independent variables over the whole search space. In our experimental setup, we only record sensor readings at the edges of the mold. Furthermore, Tobler's law of spatial auto-correlation (Tobler 1970) is violated in the experimental data: In each run, there is a single leak and thus a leak at a given point has no direct effect on the probability of a leak occurring in neighboring positions. Hence, classic spatial regression approaches such as Kriging or geographically weighted regression cannot be applied for the problem at hand.

simultaneously the major strength and weakness of this method. On the one hand, it renders the model very simple to understand and efficient to learn as well as robust to outliers. On the other hand, it constrains the predictive power of the model, as many statistical relations are nonlinear. To overcome this shortcoming, we consider a decision tree model. Decision trees predicting continuous variables are also referred to as regression trees. They map multiple input variables with an output variable through a tree structure. Thereby, they are able to take account of potential non-linear relationships in the data. Basic decision tree algorithms tend to select variables with many possible splits or missing values, leading to reduced predictive power of the model. To mitigate this selection bias, we apply conditional inference trees which use significance tests in order to select variables (Hothorn, Hornik, and Zeileis 2006). Figure 4.6 visualizes one of these trees predicting the leak's x-coordinate. For each leaf, we report the point estimate as well as a boxplot visualization of the underlying distribution of realizations. While this model class has, in general, a higher predictive power than linear regression, it is also prone to fail to generalize from the training data (Quinlan 1986). Finally, we apply a Gaussian process model as the third white-box method. This model class assumes that the output variable follows a Gaussian process fully defined by a mean and a covariance function. The covariance function expresses the expected covariance between the output variable and the input variables. Hence, linear and non-linear relationships in the data can be learned. These models are better suited for complex data compared to linear regression models. Additionally, the prediction of these models does not only provide a point estimate but also yields a quantification of uncertainty (Rasmussen 2006).

4.5.2 Black-Box Models

We first consider support vector machines. Here, each observation is viewed as a vector of all input variables. During the model training, the hyper-plane that best separates the different output variables depending on the input vectors is determined. In order to incorporate non-linear relationships, the dimensionality of the input vector is augmented using the kernel-trick if no hyperplane separating all different output variables exists. A major benefit

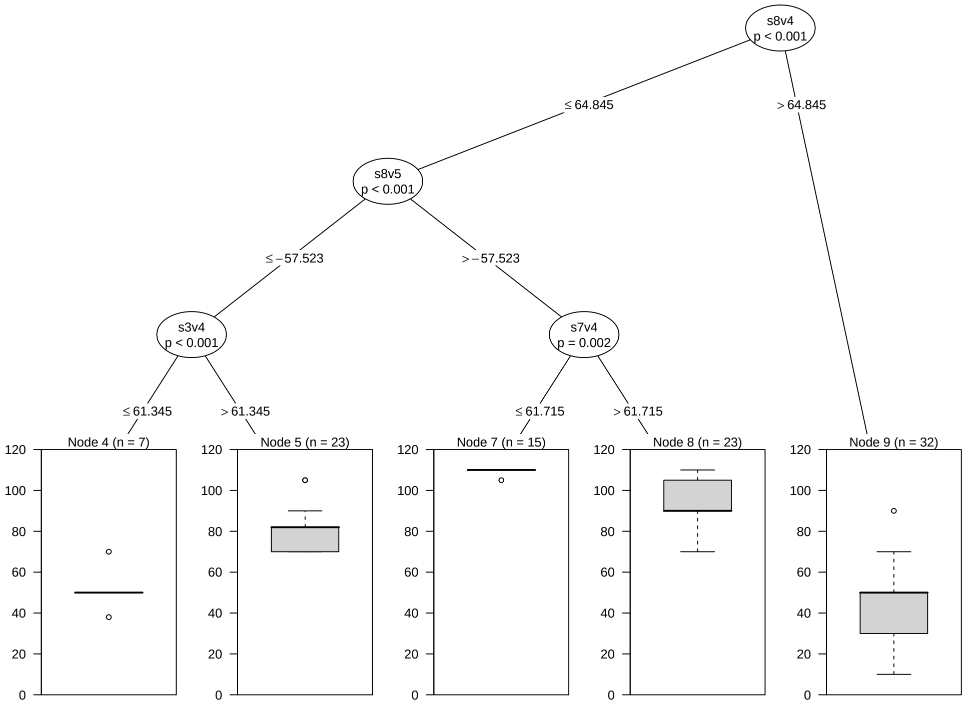


Figure 4.6: Regression tree for x-Axis

of support vector machines is the good generalization ability and therefore the low susceptibility to over-fitting even for small training data sets (Smola and Schölkopf 2004).

Additionally, we implement a Bayesian regularized neural network. Just as standard artificial neural networks, these models consist of several layers of artificial neurons. Each neuron is connected with many other neurons and processes incoming information and propagates the results to other neurons. Regular neural networks are powerful and very adaptable but tend to be prone to over-fitting. Bayesian regularized neural networks partly mitigate this risk by restricting the magnitude of the weights for each neuron (Foresee and Hagan 1997). The last black-box model we use is a generalized boosted regression and belongs to the class of gradient boosting machines. This model class combines many weak learners to a strong predictive model in a sequential fashion. This way, it is able to alleviate some of the shortcomings of the weak models while increasing the predictive power. On the downside, gradient boosting is prone to over-fitting (Friedman 2002).

Except for the linear regression and the Gaussian process, a set of hyper-parameters has to be defined in advance to train the models described above. These parameters contain, but are not limited to, the depth of the regression tree, the polynomial degree of the kernel for the support vector machine, the number of layers and neurons of the artificial neural network and the number of boosting iterations for the gradient boosting machine. Choosing a suitable parameter set is essential as the quality of the models strongly depends on them. Traditionally, performing hyper-parameter optimization was a time-consuming task as only a few trials where possible due to computational limitations (Bergstra et al. 2011). With increasing available computational power, more model training iterations can be performed. We follow Bergstra and Bengio (2012) and use random search hyper-parameter optimization, which was shown to be a very efficient method.

4.5.3 Training and Evaluation

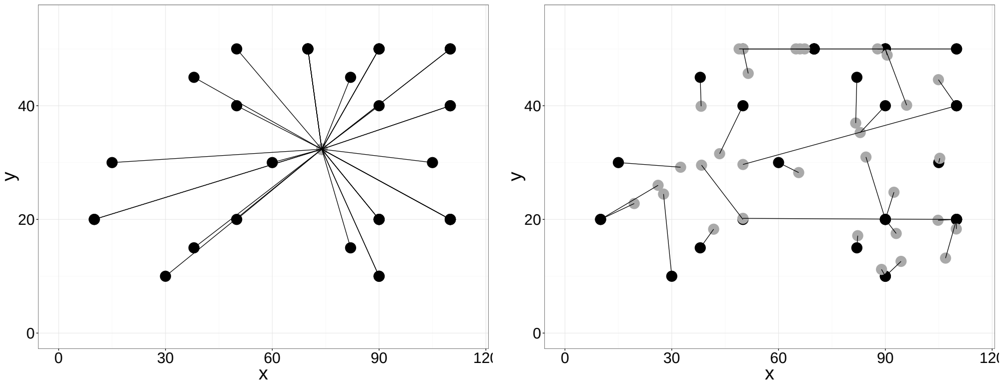
We split the data set into 75% training and 25% test data. Model training is performed using 10-fold leave group out cross-validation. Here, a group of observations is randomly selected from the training set and the model performance is evaluated using the left out observations. This process is repeated ten times to avoid over-fitting to the data and hence to increase the generalizability of the predictor. Conversely, the test set is never seen in the training phase and only used for the final evaluation shown in Table 4.2. The model quality is evaluated by means of the median absolute prediction error in comparison to the benchmark, which is calculated as the mean of the x- and y-coordinates of the training data. This method allows us to incorporate all information on non-centered data and strengthen the benchmark performance for a fairer comparison.

The evaluation shows that all models are able to outperform the naïve benchmark greatly. For the x-axis, the gradient boosting machine achieves the best results. While the complex black-box models perform best on the x-axis, the lightweight Gaussian process provides the best prediction on the y-axis. We surmise that this is due to the nesting of x-predictions within the y-forecast. Finally, we combine these two models and use a nested

Table 4.2: Relative error compared to benchmark

Model	**x**	**y**
Linear Regression	0.549	0.416
Regression Tree	0.191	1.291
Gaussian Process	0.259	**0.109**
Support Vector Machine	0.348	0.377
Neural Network	0.350	0.462
Gradient Boosting	**0.106**	0.261

two-staged machine learning model to predict the coordinates of the leaks. This approach has a mean error of 13.62 cm and a median error of 6.39 cm, implying an error reduction of 56% for the mean and 74% for the median compared to the benchmark. The results are visualized in Figures 4.7a and 4.7b. Here, the black circles show the actual location of the leaks while the connected grey circles symbolize the predicted locations.



(a) Benchmark

(b) Machine learning

Figure 4.7: Results

4.6 Prescriptive Modeling: Optimizing the Search Process

Our above analysis yields a predictive model for forecasting the leak location from the vibration sensor readings. This establishes a data-driven predictive decision support tool for the manual search process. However, our ultimate goal is to offer prescriptive decision support. To this end, the predictive model needs to be operationalized by means of a search policy. In the following, we develop a set of prescriptive search policies and compare them to simpler benchmark strategies.

4.6.1 Search Duration Problem and Simulation

To evaluate the efficacy of decision support for a search process, we need to determine the time until leak detection. This corresponds to the second statement of the optimal searcher path problem put forward by Trummel and Weisinger (1986).[12] Denoting the probability that search path ψ finds the leak by time k as $P_k(\psi)$, they established that the minimization of the expected search time

$$\min_{\psi \in \Psi} \sum_{k}^{\infty} k \left(P_k \left(\psi \right) - P_{k-1} \left(\psi \right) \right)$$

over the set of possible search paths Ψ is NP-hard. Consequently, we do not pursue an optimal solution to the problem but rely on heuristic approximations.[13]

In order to simulate a physical search process, we discretize the rectangular mold by splitting the surface into a grid of 72 square fields with a width of 10 cm each. Based on this search area, we compare search policies by calculating the total time until leak detection t. As shown in Equation 4.2, this time can be separated into the time for actually searching the grids and the time to move between the grids. The time spent searching the grids is calculated as the search time t_s times the number of searched grids

[12] The alternative formulation of maximizing the likelihood of detection is not applicable in our manufacturing scenario.

[13] Another difference to this canonical formulation is the absence of step-level probabilities due to our point forecasts.

n. The total time spent moving is the sum of the travel time t_t times the euclidean distances $d_{i,j}$ between the grids searched on each policy's route R until the leak is detected.

$$t = t_s \cdot n + \sum_{i=2}^{n} t_t \cdot d_{R_{i-1},R_i} \tag{4.2}$$

We repeat this procedure for all leaks in our data set and their corresponding position forecasts.

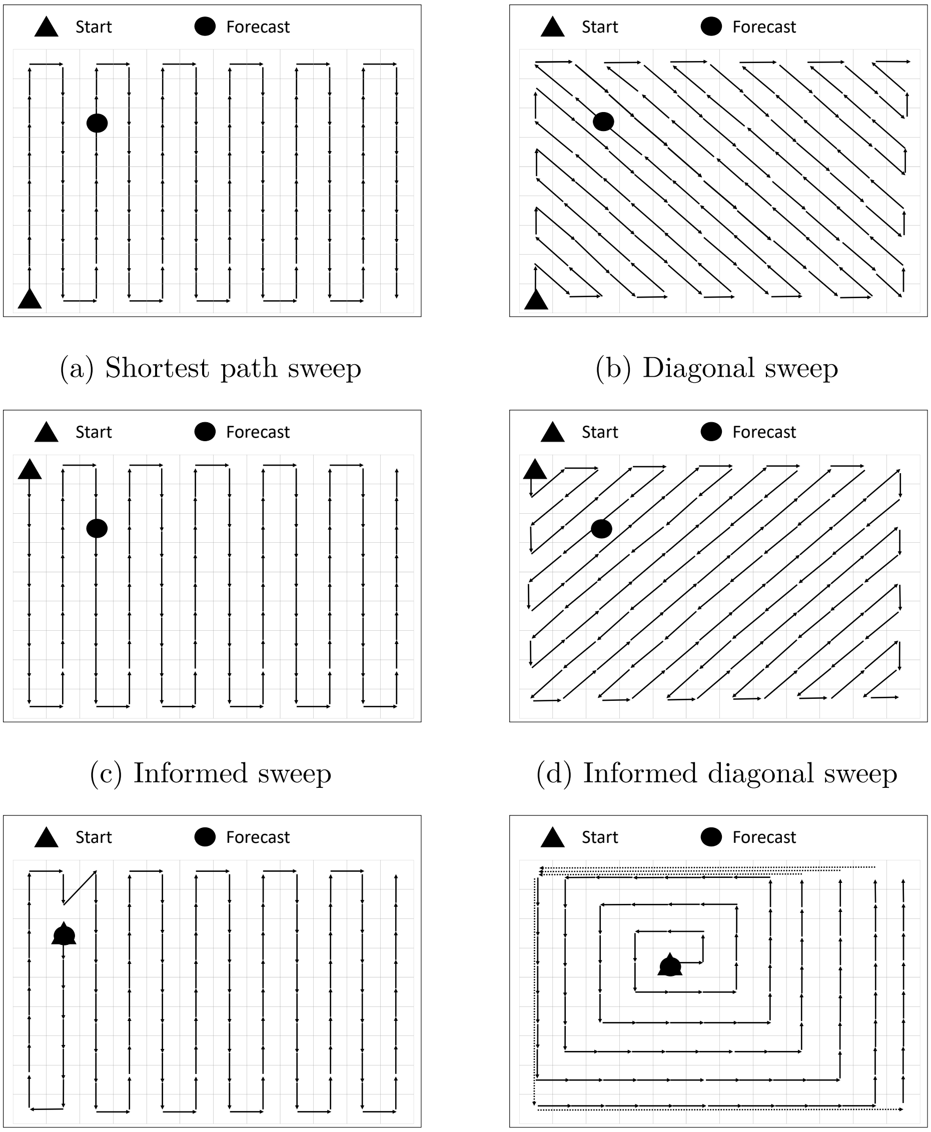
4.6.2 Search Policies

For our comparison, we consider a set of six policies for the searcher path determination. These policies can be differentiated depending on the way they translate the forecast into search paths.

The first two policies correspond to the current industry practice and serve as benchmark strategies. They do not account for any forecast information on the leak location and collapse to simple sweeps across the grid. To this end, both policies start the search in a random corner. From here on, the first strategy performs a simple sweep across the grid (Figure 4.8a). This sweep minimizes the walkway, which is one component of the expected search time. Still, this policy will often result in prolonged search times when the starting point is on the wrong side of the grid. In contrast, the second benchmark strategy performs a diagonal search across the grid. This strategy will faster search the grids close to the starting point, while the resulting search path will be longer (Figure 4.8b).

The next step is to integrate the leak position forecast from the predictive model into the search policies. In these *informed* search policies, the naïve sweeps are started in the corner closest to the predicted leak position. This approach retains the short walkways as well as the simple path structures of the benchmark approaches while incorporating information on the leak position. The informed sweep and the informed diagonal sweep are visualized in Figures 4.8c and 4.8d.

Finally, we develop two strategies to approximate the optimal searcher path, starting from the forecasted leak position. The first policy (Figure 4.8e solves a TSP problem originating at the leak position to determine the shortest route visiting all remaining fields of the grid. While this policy will result in a walkway minimizing path, it will typically visit fields close to the predicted location relatively late. To overcome this issue, the second prescriptive policy (Figure 4.8f) moves in circles around the forecasted leak location. Thereby, fields close to this position will be visited first. However, this faster coverage of close positions comes at the cost of longer paths through the whole grid.



(a) Shortest path sweep

(b) Diagonal sweep

(c) Informed sweep

(d) Informed diagonal sweep

(e) TSP search

(f) Circle search

Figure 4.8: Illustration of search policies

4.6.3 Evaluation

Considering the total time (searching plus walking) until leak detection measure, we compare the efficiency of the search process based on the different decision support instances in different settings. In particular, we want to account for the fact that depending on the scenario at hand searching or moving may be relatively more time-consuming. To this end, we report normalized total search times using an increasing ratio of search to travel times. The performance of the different policies is visualized in Figure 4.9.

The two benchmark strategies perform worst in all evaluated parametrizations. In settings with relatively low search times, the shortest path property of the simple sweep allows this strategy to outperform the diagonal sweep. On average, the diagonal sweep has 33% ($s_s = 0$) and 29% ($s_s = 1$) longer search times. However, the diagonal sweep can leverage its faster coverage property in the high search time settings and outperforms the simple sweep by 7% ($s_s = 5$) and 15% ($s_s = 10$).

The informed search policies are able to leverage the additional information and outperform the benchmarks in all settings. Unlike in the uninformed case, the informed diagonal sweep shows mostly better performance than the informed simple sweep except for the $s_s = 0$ setting. In all other settings, the faster coverage of the relevant search area can be leveraged better than the shorter path length leading to search time reductions of 51% vs. 45% ($s_s = 1$), 57% vs. 51% ($s_s = 5$) and 60% vs. 52% ($s_s = 10$).

Analyzing the prescriptive search policies, the importance of the coverage property becomes even more evident. While both policies outperform all other policies in each setting, the circle sweep remains the best strategy across all search times. On average, it leads to search time reductions of 82% vs. 54% ($s_s = 0$), 83% vs. 56% ($s_s = 1$), 85% vs. 59% ($s_s = 5$) and 86% vs. 61% ($s_s = 10$) compared to the simple sweep benchmark.

Aggregating the above results (Table 4.3), we conclude that the optimal selection of the search policy depends on the grade of available information. Without available data, the simple benchmark strategy should be preferred in settings with relatively high travel and low search times (e.g., leaks are

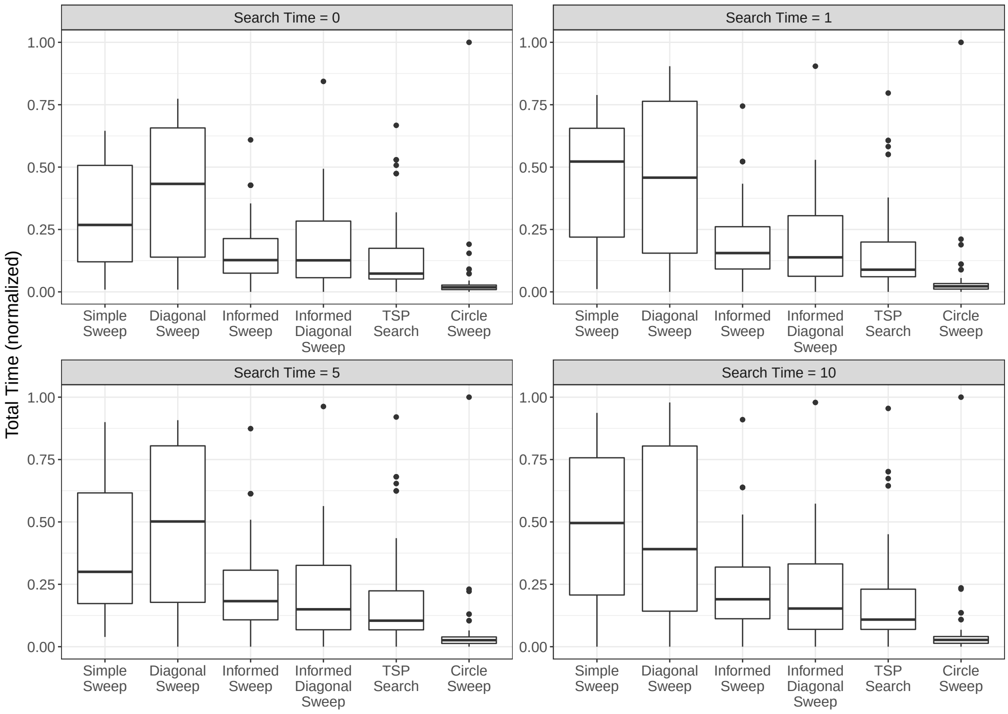


Figure 4.9: Search policy comparison

relatively easy to detect). The performance gains of the informed policies emphasize the potential value of incorporating predictive information in shop-floor processes. If sensor data is available and search times are not negligible, the diagonal policy should be selected. While the prescriptive search policies require additional support tools to guide the searcher, our results indicate that these tools might be worth the effort in many settings. Especially the circle sweep leads not only to significant search time reductions but also to reduced process variability. Hence, the proposed prescriptive decision support tool is able to decrease throughput times while increasing process stability in the shop-floor setting at hand. This finding is of particular importance for highly synchronized production lines whose performance can significantly deteriorate in the presence of process time variability (Tan et al. 1998).

Our results illustrate the value of incorporating sensor information in decision support. They also show that the predictive benefits of an improved point forecast can be sustained even when considering the actual underlying business process. Furthermore, we see promising evidence that there is

Table 4.3: Aggregated model performance

Policy	**Mean Change** **(vs. Simple Sweep)**
Circle Search	-84%
TSP Search	-57%
Informed Diagonal Sweep	-53%
Informed Sweep	-50%
Diagonal Sweep	+10%

an actual need for combining predictive analytics on the one hand and operations management techniques on the other hand in order to develop prescriptive decision support systems which can fully leverage the potential of big data.

4.7 Conclusion

Our research addresses the interface between predictive analytics and decision support systems. Having reviewed the literature on business applications of data-driven decision support, we highlighted the relative absence of decision support on the individual level in the manufacturing sector. To explore the potentials of such systems in this field, we augmented a labor-intensive manual process from high-tech manufacturing with highly sensitive sensors. Based on this setup, we illustrate the main steps and major challenges in developing and instantiating a data-driven prescriptive decision support system. By establishing a scalable and generic feature generation approach and using it in combination with techniques from statistical learning, we are able to achieve a very accurate localization of the leak. Recognizing that mere forecast information cannot be evaluated with respect to business value, we subsequently embed the problem in an operations management analysis of the underlying searcher path problem. We benchmark predictive and prescriptive search strategies against a naïve sweep strategy and find that sensor-based decision support dramatically reduces search times as well as the variability of the process.

4.7.1 Research Contribution

Based on the above findings, we posit that analytics information systems can and should play a greater role in assisting manual labor in manufacturing shop-floors. In particular, we make the following contributions to practice and theory: For practitioners, we highlight typical steps in developing a data-driven decision support artifact. In particular, we want to underline the generality of our approach: Although we showcased this analytics solution by means of the vacuum resin infusion process, there are many other settings which rely on extensive manual search activity. With the availability of appropriate training data, the feature generation and learning framework, as well as the prescriptive analytics component for determining search paths, should be straight-forward to transfer and implement. Concerning contributions to theory, we conceptualized a complementary categorization of analytical decision support applications along with the organizational scope. Thereby, we highlighted a current lack of manufacturing applications, which is a theoretical motivation for our case study. Furthermore, we put forward a simple feature generation approach for continuous (sensor) data based on the results from polynomial fitting. Finally, by combining the classic optimal searcher path problem from operations research with methods from statistical learning, we help to bridge the gap between optimization and data analytics. This is crucial for the establishment of prescriptive analytics solutions.

Faced with progressing digitization across many domains and functions the importance of analysis tools and application solutions extracting business value from these new data sources will continue to increase.

4.7.2 Limitations and Opportunities for Future Research

Clearly, an experimental analysis is subject to certain limitations. Due to the relative complexity and costliness of the recording process, the number of observations of our test data is somewhat limited. To ensure greater reliability of the results, an application in a real process setting would require additional trials before implementation. Yet, our experiment highlights the fact that big data does not always originate from high process volume but can also be a consequence of high-resolution recording. Another limitation concerning generalizability arises from the reduced size of the experimental setup. Therefore, the tested sensor setup may not be directly applicable to larger search areas. However, for very large components already small improvements to leak position forecasts would translate into considerable absolute search time improvements. Similarly, we only considered test settings with a single leak which may not be applicable for large components. To account for multiple leaks, the analytics process would have to adopt a two-step pattern where first the number of leaks is estimated and then their relative positions. This offers a promising avenue for future research. Another research opportunity would be to replace point estimate by a spatial probability distribution assigning leak probability to individual grid fields. Such a setup would necessitate a hierarchical search path determination balancing detours against higher detection probabilities.

Finally, we evaluated the decision support tool only by means of a simulation analysis. However, the actual implementation could be a major roadblock for bringing advanced decision support into practice as for example operators may not properly execute the proposed search routine leading in turn to deteriorated performance. In particular, this may imply a relative advantage of simpler policies (e.g., informed diagonal sweep) over more complicated ones. Therefore, future research needs to explore the intricacies of designing helpful interfaces (e.g., vision or speech) for embedding decision support in manual operations.

5 Data-Driven Sales Force Scheduling

Across various industries, companies need to decide how to best employ their capacity-constrained sales force. This means having to decide which prospective projects to primarily target in order to maximize expected future profits. Typically, these projects not only differ in terms of their profitability, but also have distinct characteristics, e.g., the specific type of product or service, the location, or past interactions with the prospective customer. It is reasonable to assume that these characteristics are predictive of the companies' chances of winning a particular project. In turn, these characteristics may also help quantify to what extent exerting additional sales effort influences the probability of winning a tender for a project ("the uplift"). This way, one can determine the marginal benefit of a sales representative visit to a potential customer. Building on top a large data set of successful and unsuccessful projects, we combine machine learning techniques for uplift prediction with routing and scheduling models to establish a novel *data-driven approach to sales-force scheduling.* In particular, this approach accounts for the fact that uplift predictions are imperfect and that the arising uncertainty needs to be considered when scheduling a sales force.[14]

[14] This working paper is currently under preparation for publication (Stein et al. 2019).

5.1 Introduction

In many industries (e.g., pharmaceuticals, construction, industrial services) companies spend significant shares of their marketing budget on sales force activities. To plan and schedule the activities of its sales force, companies would like to identify and prioritize those projects where additional sales efforts lead to the highest additional expected revenues. In recent years companies have reduced the size of their sales teams and, at the same time, have invested into digital technologies to improve the efficiency of the remaining sales agents (Zhang, Ohlmann, and Thomas 2014), in particular improved customer targeting (Albers, Raman, and Lee 2015). The typical result is a "priority list" of the customers who can be converted with the greatest likelihood. However, such information is insufficient to schedule sales activities if the required sales effort is not constant across all prospective projects. A case in point are traveling times in the case of physical sales meetings. Travel efforts will depend on the geographic location of clients as well as other scheduled clients. In turn, sales force scheduling becomes a multiple salesmen routing problem with a revenue maximization objective and uncertain input parameters. Yet, the required optimization framework combining prediction and prescription for sales-force management has so far not been considered in the literature.

In this paper we seek to address this research gap and propose a data-driven approach for tackling the integrated targeting and sales force scheduling. Our work is motivated by a research project with DAW, a leading German manufacturer of paint and coating solutions. In its direct sales channel DAW interacts with various customers, e.g., painters, processors or planners, to win tenders for supplying paint, mortar and other related products to construction projects. The projects not only differ in terms of their potential revenue, but also have distinct characteristics, e.g., the specific type of product or service, the project's location, or past interactions with the involved partners. It is reasonable to assume that these characteristics are, at least in part, predictive of the companies' chances of winning a particular project—even without exerting any additional sales effort. However, it is difficult to assess the probability of winning a project depending on its specific characteristics and, more so, to predict, how a certain sales activity will increase this probability (the “uplift”)—which may again be influenced by project-related characteristics.

The scheduling task is not only difficult because the company does not have accurate information about the uplift, but also, because the capacity required to visit different customers varies across projects. This is mainly because the sites of the potential customers are geographically dispersed—depending on the location of the customer and the home base of a sales rep, a customer visit can require more or less of the sales reps’ time. Hence, when scheduling its sales force, the company has to consider that visiting a “promising" customer far away from the home base may limit the number of other visits of customers that may seem less promising, but are closer to the home base.

Throughout the last years, our partner has collected extensive data on past project tenders (both successful and non-successful). Based on this data, we develop an end-to-end solution which uses state-of-the-art machine learning techniques for predicting uplifts and solves a routing problem to determine the optimal sales force schedule.

A crucial input for solving this problem is the predicted uplift for a project associated with an additional customer visit. To estimate the uplifts, we propose a two-step approach: We first train a predictive classification model to evaluate the probability of winning a specific project. This predictive model is then leveraged as a building block for our uplift approximation procedure. Under realistic circumstances we cannot assume our uplift predictions to be perfectly accurate. Consequently, the sales force scheduling model has to take the residual uncertainty of our uplift predictions into account. To this end we propose a novel weighting scheme motivated by decision analysis considerations. Thereby this approach is able to explicitly control for the level of confidence which is attributed to the predictive model. We provide an extensive numerical analysis of this proposed scheme. Figure 5.1 summarizes our approach.

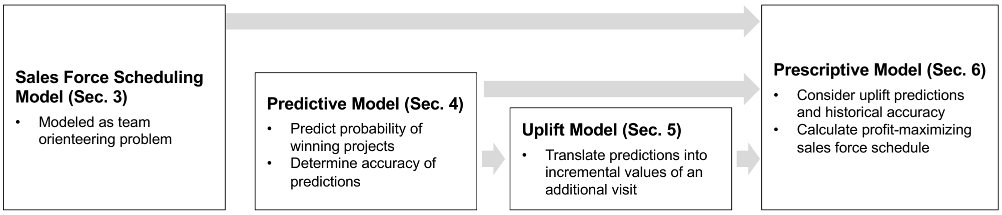

Sales Force Scheduling Model (Sec. 3)
• Modeled as team orienteering problem
Predictive Model (Sec. 4)
• Predict probability of winning projects
• Determine accuracy of predictions
Uplift Model (Sec. 5)
• Translate predictions into incremental values of an additional visit
Prescriptive Model (Sec. 6)
• Consider uplift predictions and historical accuracy
• Calculate profit-maximizing sales force schedule

Figure 5.1: Overview of the proposed data-driven approach

As a distinctive feature, our solution accounts for the uncertainty in the uplift predictions. Thereby we acknowledge the inherent probabilistic nature of the uplift predictions which may cause the sales force scheduling model to falsely prioritize customers with a too optimistic uplift prediction over customers with a correct or too conservative prediction. Using real-world data, we demonstrate the usefulness of this approach which should be applicable for many other companies facing sales force scheduling problems. From a more general perspective, our approach establishes a crucial link between marketing/sales analytics and traditional operations management problems.

5.2 Literature Review

Our work is related to two distinct literature streams. On the one hand, we are concerned with a traditional OM problem, the determination of optimal routes for a group of salesmen, which is similar in structure to a vehicle routing problem with profits. On the other hand, we draw from research on machine learning applications in marketing. More specifically, our setting necessitates *uplift* modeling techniques to predict the benefit of performing an action compared to not performing this action (e.g., visiting a potential customer).

We approach the sales force scheduling problem with a specific version of the vehicle routing problem with profits (cf. Feillet, Dejax, and Gendreau 2005). In particular we are considering the *team orienteering problem (TOP)* (Gunawan, Lau, and Vansteenwegen 2016). In these problems each stop is associated with a specific profit and capacities are constrained such that the team cannot “visit” all potential locations. In our specific setting, a TOP has to be solved for each day in order to generate a schedule for a capacity-constrained sales force in the construction industry where additional profits are associated with a visit to a potential customer. Most of the work on TOPs assumes that all relevant parameters (including the profits) are known to the decision maker (e.g., El-Hajj, Dang, and Moukrim 2016; Archetti, Carrabs, and Cerulli 2018). As we will describe below, the profits in our model are not certain, but rely critically on uncertain estimates of "uplifts"—that is the increases in the probabilities of winning projects associated with additional customer visits —that are obtained by means of predictive machine learning models. Most contributions that have considered parameter uncertainty in TOPs focused on uncertainty in service and travel times (e.g., Campbell, Gendreau, and Thomas 2011; Evers et al. 2014; Papapanagiotou, Montemanni, and Gambardella 2014; Zhang, Ohlmann, and Thomas 2014), which is less of an issue in our particular context. However, only few researchers have considered uncertainty with respect to the profits at each stop which is a crucial issue in our analysis. Ke et al. (2013) argue that parameter uncertainty unavoidably exists but oftentimes, reliable intervals of the actual parameter values can be provided. They adapt the TOP to consider these interval estimates for all model parameters, i.e., profits, service times, and travel times, and solve the resulting problem via a robust optimization approach. Ilhan, Iravani, and Daskin (2008) formulate and solve an orienteering problem that intends to maximize the probability of collecting more than a pre-specified target profit level assuming that the collected profits at each individual location follow a normal distribution. In contrast to these approaches, we estimate success probabilities for collecting individual profits and propose a novel way of dealing with the residual model uncertainty. In turn, we put forward a data-driven, non-parametric approach.

Tulabandhula and Rudin (2014) propose a data-driven approach to a related problem where the probability of service events in the power sector has to be estimated in order to optimally route repair crews. They employ logistic regression to predict the probabilities and by means of a regularization term derive cost-optimal maintenance policies based on these predictions. The key difference of our setting to the one examined by Tulabandhula and Rudin is the possibility of stops not being successful: Whereas a repair crew in their case will reliably repair an asset, a salesperson in our setting can only improve the probability of winning a project. Hence, we focus on estimating the change of probability as a consequence of such a stop.

Estimating the effect of sales efforts from historical data is a task the marketing literature has been concerned with for some time and which is frequently referred to as uplift or incremental value modeling. There are numerous approaches for uplift modeling (e.g., Hansotia and Rukstales 2002; Rzepakowski and Jaroszewicz 2012; Guelman, Guillén, and Pérez-Marín 2015; Zhao, Fang, and Simchi-Levi 2017; Wager and Athey 2018) which all have in common that they require data from two distinct groups, the *treatment* group and the *control* group. For the treatment group a certain sales activity has been performed, while this action has not been taken for the control group. Unfortunately, when dealing with a numerical action variable such as the number of customer visits, this logic is no longer applicable because we cannot distinguish between instances with "action performed" and "no action performed". Manchanda and Chintagunta (2004) overcome this problem and derive uplift estimates for numerical action variables by means of a hierarchical Bayesian count data model. However, this approach is not applicable if the uplifts depend on complex interrelations between multiple variables. Therefore, our work builds upon the methods presented in Foster, Taylor, and Ruberg (2011) and van de Geer, Wang, and Bhulai (2018) who both deal with the issue of not having learning data from distinct treatment and control groups. In a clinical setting, Foster, Taylor, and Ruberg (2011) identify subgroups of patients for which a specific treatment is particularly successful by predicting individual treatment effects. To this end, they consider the treatment indicator as a binary feature variable and learn a random forest model to predict an individual success probability. To derive estimates for the treatment effect of a single patient they then calculate the success probabilities for both expressions of the treatment indicator (1 and 0) and subtract the results. Recently, this approach was extended to also account for numerical action variables (e.g., the number of outbound calls) instead of binary treatment indicators. van de Geer, Wang, and Bhulai (2018) consider the case of a debt collector who can influence the probability of a debtor settling her debt through direct interactions between collector and debtor. Similarly to Foster they learn a prediction model that considers the number of past collector-debtor interactions to estimate recollection probabilities of single debtors. To predict the uplift of an additional interaction – say an additional call – they recalculate these recollection probabilities with the number of past collector-debtor interactions increased by one and subtract the base probability from this estimate. Very recently, a similar approach was applied in a setting where the value of online advertisements should be predicted by attributing the conversion credit of customer purchases to advertisements in different marketing channels (Wang 2019).

In contrast to the aforementioned literature, in our setting it is not obvious how to translate the predicted uplift values into actionable insights, i.e., how to derive optimal tours for the salespeople. Therefore, we combine the uplift estimates with a complex scheduling problem. In turn, our main contribution is based on the combination of these two literature streams. Since we cannot assume perfect predictions from a data-driven model, we provide a new approach to control for the reliability of these uplift values.

5.3 Problem Description

Consider a company bidding for various projects $k \in K$ with different profitabilities χ_k. Each project k is associated with a customer $c \in C$. Large construction companies are typically engaged in many projects so that a customer can be associated with multiple projects. We denote the set of projects of a particular customer c by K_c. Let p_k denote the company's probability of winning a project k. To increase the chances of winning a project, the company can exert sales effort in the form of a visit to the potential customer. We denote the uplift (e.g., the change in probability of winning project k) after an additional visit by Δp_k.

The company's sales force consists of sales reps positioned in a common home base. Each sales rep has a capacity (e.g., working hours per day), which is denoted by η^{max}. We denote the duration of a visit at customer c by τ_c^{visit}. Furthermore, the travel time between a pair of locations (customers or home base) i and j is given by τ_{ij}^{travel}.

The company's goal is to determine a sales force schedule π that maximizes the expected additional profits, denoted by $v(\pi)$, associated with this schedule. To obtain a schedule π, the company first determines a set of tours T based on the uplifts Δp_k, the profits χ_k and the available sales force capacity. Each tour $t \in T$ defines the order in which a sales rep visits a set of customers on a particular day. We denote by x_{ijt} the binary variable that indicates if the travel segment from location i to location j is a part of the tour t and by y_{ckt} the binary variable that indicates if customer c is visited on tour t to pitch project k. The decision variables x_{ijt} and y_{ckt} fully specify the tours $t \in T$ and the subset of customers to be visited. We assume that each sales rep can perform exactly one tour with capacity η^{max}, so the overall sales force capacity for the next day is $|T|\eta^{max}$. In a second step, a sales rep is assigned to each tour $t \in T$. In our analysis, we assume that sales reps are homogeneous in their preferences and that the uplifts Δp_k are independent of the sales rep who visits the customer. Therefore, any sales rep can be assigned to any tour $t \in T$, and a schedule π corresponds to a set of tours T. To determine the optimal schedule π^*, the company solves the following optimization problem:

$$\max v(\pi) = \sum_{c \in C} \sum_{t \in T} \sum_{k \in K_c} \Delta p_k \chi_k y_{ckt} \tag{5.1}$$

subject to the following set of constraints:

$$\sum_{j \in C} x_{0jt} = 1 \qquad \forall t \in T \tag{5.2}$$

$$\sum_{i \in C} x_{i0t} = 1 \qquad \forall t \in T \tag{5.3}$$

$$\sum_{j \in C, j \neq c} x_{ijt} \geqslant y_{ckt} \qquad \forall i \in C \ \& \ \forall k \in K_c \ \& \ t \in T \tag{5.4}$$

$$\sum_{j \in C, j \neq c} x_{jit} \geqslant y_{ckt} \qquad \forall i \in C \ \& \ \forall k \in K_c \ \& \ t \in T \tag{5.5}$$

$$\sum_{s \in T} y_{ckt} \leqslant 1 \qquad \forall c \in C \ \& \ \forall k \in K_c \tag{5.6}$$

$$\begin{aligned} &\sum_{j \in C} x_{0jt} \tau_{0j}^{travel} + \sum_{i \in C} x_{i0t} \tau_{i0}^{travel} + \\ &\sum_{c \in C, j \in C} x_{cjt} \tau_{cj}^{travel} + \sum_{c \in C, k \in K_c} y_{ckt} \tau_c^{visit} \leqslant \eta^{max} \qquad \forall t \in T \end{aligned} \tag{5.7}$$

Constraints (5.2) and (5.3) ensure that each tour s starts and ends at the home base 0. Constraints (5.4), (5.5) and (5.6) ensure that each customer location has one in-going and one out-going connection if a sales representative pitches at least one project related to the customer on a given tour t and no connections otherwise. The maximum tour lengths are ensured by constraint (5.7). Additional sub-tour elimination constraints are required but omitted for sake of brevity. Given the necessary input parameters, optimal schedules π^* for realistic problem sizes can be calculated in reasonable time using commercial MIP solvers.[15]

Table 5.1: Input parameters and decision variables

T	Set of tours.
C	Set of customer locations.
K	Set of projects.
K_c	Set of projects associated with customer c.
η^{max}	Maximum tour length (time).
τ_c^{visit}	Duration of a visit at customer c.
τ_{ij}^{travel}	Travel time between a pair of locations i and j.
Δp_k	Uplift generated by visiting customer c to discuss project k.
χ_k	Profit of winning craft k.
$y_{ckt} \in \{0,1\}$	Indicates if customer c is visited on tour t regarding project k.
$x_{ijt} \in \{0,1\}$	Indicates if travel segment between locations i and j is scheduled on tour t.

While the above formulation reflects a somewhat simple instance of the problem, our model can be extended to take more complex settings into account. First, the sales force scheduling is currently performed on a day-to-day basis—that is, a sales force schedule is determined each day for the next day. Our approach does, however, extend naturally to a planning horizon of multiple periods (days). Second, we currently assume that sales reps are

[15] Using Gurobi, (Gurobi Optimization 2016) realistic problem instances (e.g. 50 projects and 5 tours) can be solved to optimality in less than 30 minutes on a 12 CPU system.

homogeneous in terms of their preferences and capabilities. However, in real-world applications, the uplifts might depend on the person performing the visit. We could extend our model to account for heterogeneous uplifts if historical data on the sales rep level were available. Third, we assume that customers can be visited at any time during a day. In practice, however, there may be restrictions to when a customer can be visited. To account for such restrictions, we can formulate and solve a team orienteering problem with time windows (Vansteenwegen, Souffriau, and Oudheusden 2011).

5.4 Predictive Modeling

To determine an optimal schedule for the sales representatives the planner requires the uplift values Δp_k per project. These uplift values cannot be observed – and consequently, we cannot train a machine learning model to predict such values. However, we can train a predictive model from historical data in order to estimate the probabilities of winning a particular project. This predictive model then serves as a building block for the subsequent uplift approximation procedure. In the following, we first describe the available data and our feature engineering approach. Subsequently, we show which machine learning approaches we apply and compare their performances in order to choose the best approach for the subsequent uplift approximation in Section 5.5.

5.4.1 Data Set

DAW, our partner company, can revert to a database containing information about development projects in Germany from January 2015 through May 2017. Figure 5.2 provides an overview of the underlying relational data structure.

Building Table Clearly, for a large development project, multiple subprojects such as interior and exterior painting can be performed. For this reason, one table contains general information on the development project such as its type or its location.

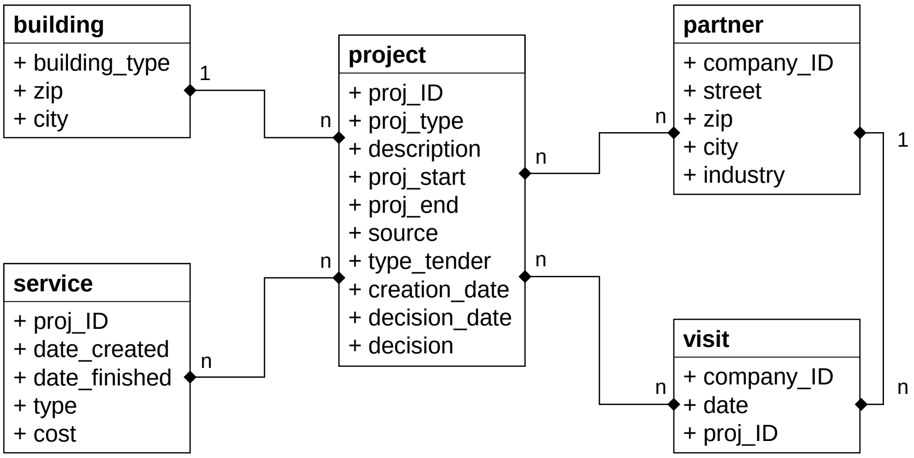


Figure 5.2: Overview of the existing data structure.

Project Table This central table of the database contains information about specific work packages of large development projects. Since these work packages lie at the center of our subsequent analyses we simply refer to them as "projects" in the following. Each data record comprises a description and categorization of the type of project (e.g., painting works, plastering works, repairs), timing information such as construction start and end dates as well as the characteristics of the project assignment (e.g., public tender, direct assignment). Moreover, the final decision of whether a project was won or lost is stored.

Partners Table Large projects typically involve several distinct partners who might also collaborate on a number of different projects. The stored information about these companies involves location information as well as the industry and the specific function in which the partner is involved at a particular project.

Company Visits Table A fourth table stores interactions between the company and its partners in the past. Each data entry represents an appointment at a specific date. For some but not all of these appointments, more specific information about the discussed project is included.

Services Table Finally, our partner company, DAW, offers several services to potential partner companies such as providing color samples or a personalized color consultation. These services are typically project-specific and can hence be rather attributed to a particular project instead of a partner company. Information about the performed services contain the type of service, the date when it was performed as well as the value of the performed services.

Data Cleaning and Processing On this raw data set, the following basic data cleaning routines were performed. For a large part of the $46,913$ included projects, no final decision was stored. For some of them, we were able to impute the decision by assuming that projects being not assigned within 365 days were lost. Then, the rest of the projects without assignment decision was discarded. Going forward, we then removed duplicate entries. As visualized in Figure 5.3, the outcome of $2,929$ of the $8,444$ remaining projects is decided within less than one week after the record is being created in the system. Additionally, only 73 of these $2,929$ instantaneously decided projects are lost. This finding indicates that a significant number of projects is decided–and oftentimes won–at the first contact. We exclude those entries from our data set, as such projects are never candidates for possible visits and would therefore add an unnecessary bias to our model. The final data set for the subsequent analyses then contains $5,515$ projects out of which DAW had won a supply contract in $3,828$ of the instances.

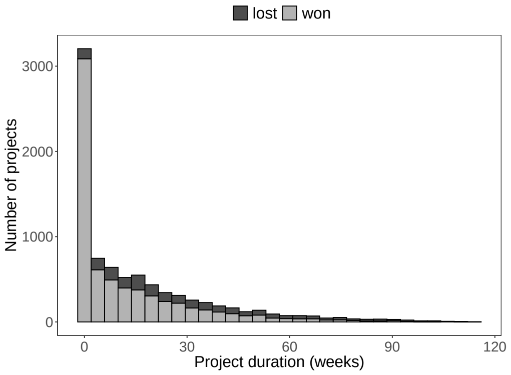


Figure 5.3: Time in the system before a decision is made.

In order to determine the project-specific uplift values Δp_k from the data, we first need estimates for the probability $\hat{p}(\xi_k)$ of winning project k where k is specified by some feature vector ξ_k. Clearly, this probability depends on many factors, some of which we can influence (e.g., the number of sales representatives' visits) and some of which we cannot (e.g., the type of project, involved partners or the type of project assignment). Figure 5.4 visualizes an exemplary timeline of such a decision process. Additional services are typically requested by the partners and hence are exogeneous whereas the company decides how much face-to-face sales effort to put into a lead, i.e., how often a sales representative would visit the partner company in person.

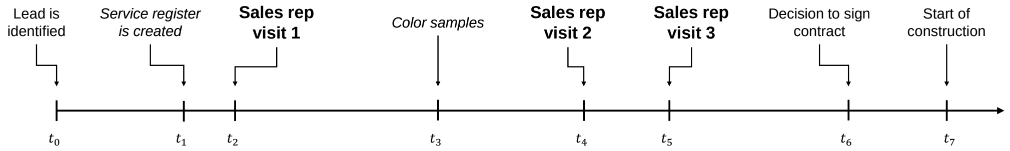


Figure 5.4: Illustration of a projects's timeline.

We use supervised machine learning to obtain a model that yields predictions $\hat{p}(\xi_k)$ given a feature vector ξ_k. Supervised machine learning means that we train such a model on a large set of historical data consisting of pairs of a feature vector and the information whether the project was won or lost in order to discover a relational structure between these information in the data. In the following subsection, we illustrate how we obtain the features given the raw data and subsequently train and evaluate different machine learning approaches.

5.4.2 Feature Engineering

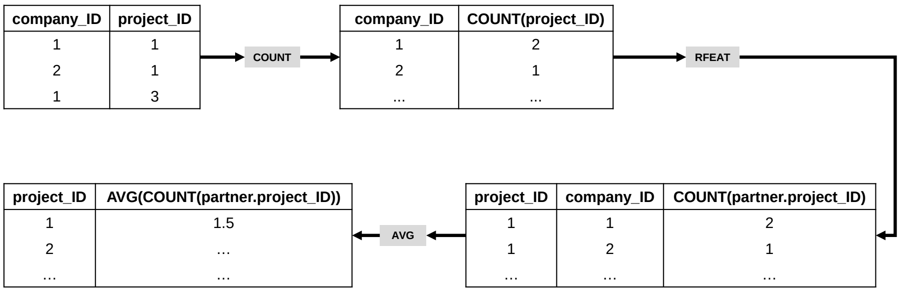
Relating back to the data structure described in 5.4.1, it becomes obvious that some of the data attributes can be directly used as features (e.g., the static building and project-specific information). However, information encoded in other entities (e.g., partners, services or visits) have to be exploited by means of suitable data transformations. We considered the following direct features:

- **building type** – e.g., apartment, office, church, hotel, ...
- **project type** – e.g., reconstruction, renovation, ...

- **work package** – e.g., interior / exterior painting works, heat insulation, ...
- **lead source** – e.g., planner, building owner, general contractor, ...
- **selection process** – e.g., direct, public tender, regional tender,...

These basic features capture static, isolated properties of a project. To also include dynamic relationships as well as inter-dependencies we performed extensive feature engineering activities described below.

Features from Entity-Relation Summaries In order to generate further meaningful features, we applied simple data transformations following the general approach presented in Kanter and Veeramachaneni (2015). First, we calculated the number of involved partner companies via a count operation on the partner table. Second, for each partner company, the number of collaborations on past projects was determined. Then, we calculated the average of this number over all partner companies involved in a particular project. Figure 5.5 visualizes the process of determining this feature.



company_ID	project_ID
1	1
2	1
1	3

company_ID	COUNT(project_ID)
1	2
2	1
...	...

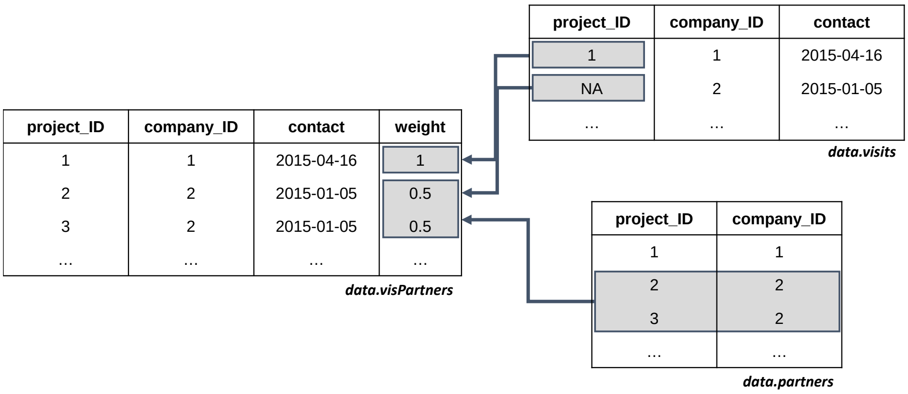
project_ID	company_ID	COUNT(partner.project_ID)
1	1	2
1	2	1
...	...	...

project_ID	AVG(COUNT(partner.project_ID))
1	1.5
2	...
...	...

Figure 5.5: Exemplary calculation of the average number of historical projects with involved partner companies for a particular project.

Furthermore, we aggregated the value of additional services per project and considered it as collaboration-specific feature. Finally, we added binary features describing whether a service register was created and whether additional services were performed.

Recency of project-related customer visits A particularly interesting set of features is the number of visits at the involved partner companies within the last t weeks before the project decision is made. To determine this number we first have to assign a particular visit to a project. For project-specific visits, the *data.visits* table reports a unique project identifier which, however, is not mandatory. In the case such a project identifier is stored, the visit can be entirely attributed to this project (weight $w = 1$). On the other hand, for visits without a unique project identifier we implicitly assume that all collaborations with partner i were discussed and we weight the visit for each single project with $w = 1/|K_i|$. Figure 5.6 visualizes this procedure.



project_ID	company_ID	contact
1	1	2015-04-16
NA	2	2015-01-05
...	...	...

data.visits

project_ID	company_ID	contact	weight
1	1	2015-04-16	1
2	2	2015-01-05	0.5
3	2	2015-01-05	0.5
...	...	...	...

data.visPartners

project_ID	company_ID
1	1
2	2
3	2
...	...

data.partners

Figure 5.6: Feature engineering to calculate the weighted number of visits

Besides the raw number of visits, we posit that the timing of these visits is informative with respect to the success probability. In order to capture this temporal effect, we aggregate the number of past visits over distinct lookback horizons spanning from one week to two years. These aggregates are then included as individual features. Naturally, these features will give rise to a certain degree of multicollinearity. While this is problematic in explanatory modeling where one wants to interpret coefficient values, we are only interested in high predictive power. In particular, machine learning algorithms are considered as robust with respect to multicollinearity (Mayr et al. 2014).

5.4.3 Models and Training

We train and evaluate several prediction models to estimate the success probability of winning a project. In particular, we use both white-box methods, i.e., methods where the model structure and its fitted parameters are interpretable by a human decision-maker, and black-box models which typically achieve better predictive accuracy at the cost of limited interpretability.

As a baseline white-box classifier, we rely on logistic regression (Cox 1958). Common problems in applied logistic regression arise from the abovementioned multicollinearity of covariates and from separation, i.e., when a linear combination of features is highly predictive of the outcome. For this reason, we use a Bayesian version of logistic regression (cf. Gelman et al. 2008). While being highly interpretable, logistic regression typically performs inferior to more elaborate black-box models in many practice-relevant scenarios due to its implicit assumption of a constant, linear relationship between covariates and the outcome. For this reason, we compare its results with those of three black-box models. The first of these models is a support vector machine which is based on the theory developed in Vapnik (1996). Grounded in statistical learning theory, support vector machines fit hyperplanes into the feature space in order to optimally separate the output classes of interest. A major benefit of support vector machines is their high robustness towards overfitting (Smola and Schölkopf 2004). Additionally, we implement an artificial neural network (ANN) model. ANNs use non-linear functions applied to linear combinations of features in order to make predictions. They are a powerful and flexible learning method and applicable in many fields (Hastie, Tibshirani, and Friedman 2013). Finally, we train a random forest classification model (Breiman 2001a) that is based on an ensemble version of decision trees and has proven to perform well over a wide range of applications (Caruana, Karampatziakis, and Yessenalina 2008).

We use an 80% subsample of the data to train the models. The remaining 20% of the data are split into a test sample to evaluate the respective classification performance (15%) and an evaluation sample ($\sim$ 5%) where we perform the evaluation of our prescriptive scheduling model in Section 5.7. Within the training data, we apply 10-fold cross-validation to tune the models (Hastie, Tibshirani, and Friedman 2013).

5.4.4 Model Evaluation and Selection

In order to compare the performance of the four examined models, we report the *receiver operating characteristic (ROC)*, the according *area under the curve (AUC)*, the *F1 score* and the *phi coefficient (ϕ)*. The (ROC) is a graphical illustration of the predictive performance of a binary prediction model. To capture the information from ROC curves in a single numeric metric, one typically calculates the area under the curve (AUC) for comparisons between classifiers. The *F1 score* is a measure for the accuracy of a test that considers both - precision and recall, where precision is the ratio of true positives from all positive predictions and recall is the ratio of true positive predictions from all positive samples.

Besides the AUC and F1 statistic, the phi coefficient ϕ, also known as the Matthews correlation coefficient, is often regarded as one of the best single number measures of classification performance (Powers 2011). This is in particular because of its robustness towards class imbalances. For binary classification problems the ϕ coefficient coincides with the Pearson correlation coefficient measuring the correlation between true and predicted outcome of binary classification yielding values between -1 and +1. When used as a metric for machine learning model evaluation only positive ϕ values are of interest: A value of +1 indicates a perfect prediction while a value of 0 is a random prediction.

Figure 5.7 visualizes the ROC curves of the four examined models. We see that the Random Forest classifier performs considerably better than its competitors over a large range of the considered spectrum while the neural network works best in a specific area of the considered configurations.

The numerical performance metrics (Table 5.2) confirm this assessment. The Random Forest approach achieves AUC, F1 and ϕ scores that are higher than those of the other approaches, which is why we choose this model as a base for our subsequent uplift approximation procedure.

5.5 Uplift Approximation

A crucial input to optimize the scheduling problem presented in Section 5.3 are the specific values of a visit at a particular customer's location. Figure 5.8

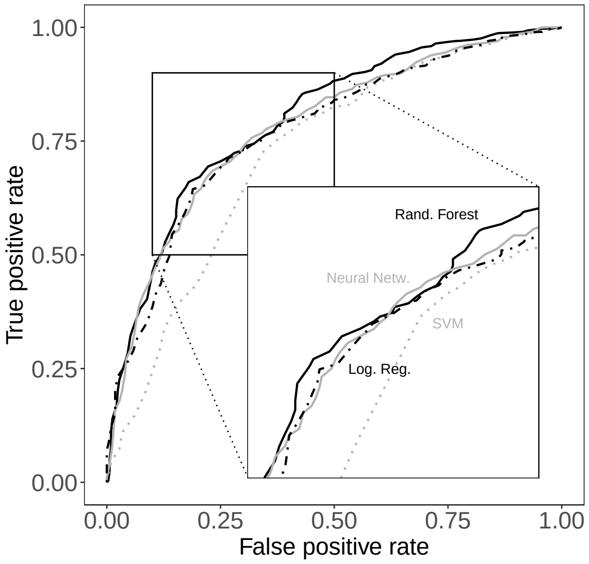
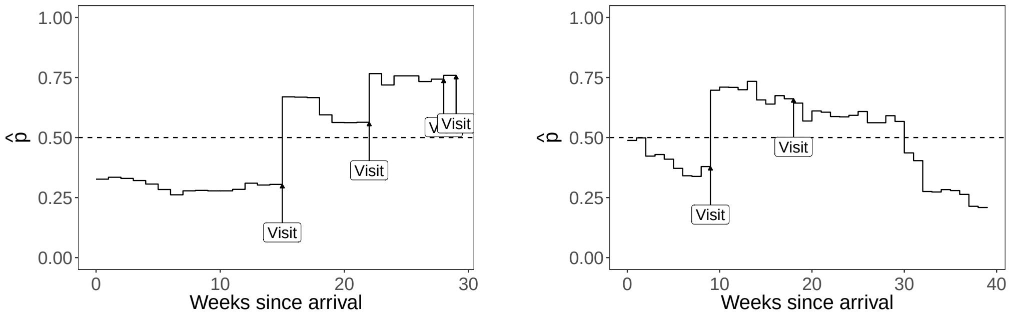


Figure 5.7: Comparison of the models' receiving-operating characteristics

	AUC	**F1**	ϕ
Support Vector Machine	0.72	0.82	0.30
Logistic Regression	0.77	0.82	0.34
Neural Network	0.78	0.82	0.37
Random Forest	**0.80**	**0.83**	**0.42**

Table 5.2: Comparison of classifier performance

visualizes exemplary trajectories of the estimated probability $\hat{p}$ as given by the Random Forest model from Section 5.4. Evidently, the information about the success probability by itself does not allow the planner to optimize the schedule of her sales agents: Without knowing the actual “uplift” of an additional visit (compare, for example, the value, i.e., the uplift of the first visit to the one of the second and third visits in figure 5.8a) one cannot trade-off driving further distances for a large increase in a single project's success probability versus increasing the probabilities of multiple projects in the surroundings by a smaller margin.



(a) Won Project

(b) Lost project

Figure 5.8: Exemplary trajectories of success likelihood

Hence, instead of the estimated probability $\hat{p}$, we are interested in the uplift Δp_k for each potential visit. Determining such uplift values is a particularly challenging task as the actual uplift cannot be observed. In the following, we describe our approach to approximate such uplifts given the prediction model from Section 5.4.

Modeling uplift values typically requires data from two distinct groups, the treatment group and the control group. However, in our setting, the action variable, i.e., the number of customer visits, is numerical and for this reason we cannot divide our data into disjoint subsets. Instead, we generate a "synthetic" treatment data set by adding fictitious customer visits and employ the predictive model to recalculate success probabilities. Then, we can determine the uplift as the difference between the probabilities of the synthetic data set and the original data set. This logic draws from the literature, particularly from Foster, Taylor, and Ruberg (2011) and van de Geer, Wang, and Bhulai (2018). Our feature vector ξ can be refined by interpreting it as the union of a vector of endogenous "action features" $\mathbf{a}$, i.e., the number of customer visits, and the vector of exogenous features $\mathbf{z}$. Hence, our vector of predictions becomes $\hat{p}(\xi) = \hat{p}(\mathbf{a}, \mathbf{z})$. Note that we use the same prediction models as introduced in the previous section. Given the feature data set as well as the trained prediction models described in Subsection 5.4.4, we first generate the synthetic "treatment" data set in which we increment $\mathbf{a}_k$ by one. We then calculate $\hat{p}(\mathbf{a}_k + 1, \mathbf{z}_k)$ where all other features $\mathbf{z}_k$ are retained unchanged. Finally, we calculate the uplift as $\Delta p_k = \hat{p}(\mathbf{a_k} + 1, \mathbf{z}_k) - \hat{p}(\mathbf{a}_k, \mathbf{z}_k)$.

Figure 5.9 visualizes the distribution of the calculated uplift values depending on the number of prior visits $\mathbf{a}_k$. Not surprisingly, we observe that on average visits have a positive effect on the chance of winning a project ($\overline{\Delta\hat{p}}$). Additionally, we see that the median value of the first visit is significantly higher compared to additional visits. Partly, this effect can be explained by the fact that the prior probabilities $\hat{p}(\mathbf{a}_k, \mathbf{z}_k)$ are increasing with the number of visits resulting in a lower overall uplift potential. However, the uplift of a visit can not be completely explained by the number of prior visits. This finding shows, that the other features $\mathbf{z}$ capture a relevant part of the information and provide valuable information to identify persuadable customers.

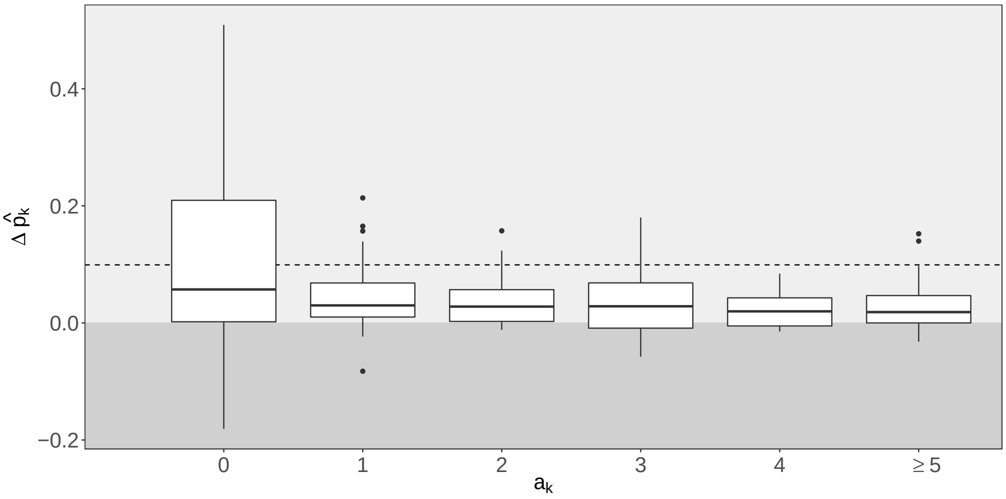


Figure 5.9: Uplift values Δp depending on the number of prior visits $\mathbf{a}$.

To better understand the robustness of our approach, we examine the impact on distributional properties of synthetically changing the original data set. In particular, we compare the distribution of predicted probabilities for the original data with the distribution for the synthetically modified number of visits. Figure 5.10 shows that the empirical CDFs of the predicted probabilities behave very similarly for the original and the synthetic data set. Therefore, the synthetic treatment approach does not systematically distort the distributional properties of success likelihood in a local neighborhood. In turn, we surmise that this approach is capable of generating reliable uplift predictions.

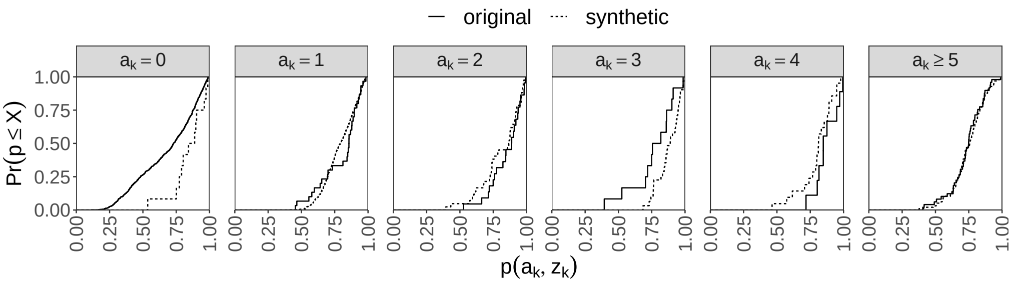


Figure 5.10: Comparison of the distribution of predicted probabilities. Individual panels correspond to number of prior visits **a**.

In many settings this uplift information by itself already provides important insights and can be used, for example, to prioritize projects where additional effort seems beneficial (cf. van de Geer, Wang, and Bhulai 2018). In particular, this applies for the case of sales activities in finance or insurance where sales actions involve outbound calls to potential customers. However, in our setting, the effort, i.e., the total time required to visit a particular customer is largely affected by the travel time of the sales representative and, hence, strongly depends on the location of the customer as well as the locations of other customers. Therefore, we use a sales force scheduling model that accounts for the trade-off between higher uplifts and the time required for visiting a customer.

5.6 Prescriptive Sales-Force Scheduling With Forecast Uncertainty

In our prescriptive sales force scheduling approach we leverage the uplift predictions $\Delta\hat{p}_k$ and at the same time account for the quality of these predictions—that is, the fact that these predictions are uncertain and may entail a larger or smaller margin of error. If we assume that we have a perfect prediction model that yields accurate uplift values we can simply solve the optimization problem stated in Section 5.3 using the predictions $\Delta\hat{p}_k$ as inputs. Without information about the uplift, the optimal strategy would be to maximize the sum of the potential profits, which is equivalent to solving the optimization problem stated in Section 5.3 with a constant uplift $\Delta\hat{p}_k \equiv \alpha$ across all projects. The objective function of the optimization problem then simplifies to

$$\max v(\pi) = \sum_{c\in C}\sum_{t\in T}\sum_{k\in K_c} \alpha\chi_k y_{ckt} \tag{5.8}$$

In reality, we will face situations that lie between these two corner cases: Even with sophisticated predictive machine learning models and extensive data, a margin of error will remain and ignoring this uncertainty may lead us to schedule detours to visit customers with overestimated uplifts while, at the same time, not scheduling visits to customers with (too) low uplift predictions.

Clearly, the worse the prediction quality, the less we should rely on the uplift predictions and the more we should prioritize according to the profitability of the projects. The Hodges-Lehmann criterion (Hodges, Lehmann, et al. 1952) from decision theory offers a simple way of formalizing the tradeoff between the two corner cases. It suggests that with unreliable probability assessments a decision-maker has to compromise between a conservative worst-case policy and the optimistic expected value maximizing alternative. In our setting the worst-case corresponds to discarding the information from the uplift prediction model while the expected value maximization puts full confidence in the uplift predictions. These two objectives are linearly weighted by a parameter λ which has to be specified by the decision-maker (e.g., based on experience). Based on this logic, the objective function can be stated as

$$\max v(\pi_\lambda) = (1-\lambda)\sum_{c\in C}\sum_{t\in T}\sum_{k\in K_c} \overline{\Delta\hat{p}}\chi_k y_{ckt} + \lambda\sum_{c\in C}\sum_{t\in T}\sum_{k\in K_c} \Delta\hat{p}_k\chi_k y_{ckt}, \tag{5.9}$$

where

$$\overline{\Delta\hat{p}} = \frac{1}{|K|}\sum_{k\in K}\Delta\hat{p}_k. \tag{5.10}$$

For $\lambda = 0$ we obtain the conservative worst-case policy and for $\lambda = 1$ the policy that maximizes the expected value by putting full confidence in the uplift predictions. To normalize the scale we set α to the average predicted uplift of all projects in the test data $\overline{\Delta\hat{p}}$ (Equation 5.10).

Naturally, the decision maker faces the problem of setting λ. We argue that a natural candidate for this parameter is the model quality ϕ. This can be rationalized by drawing an analogy with a simple linear regression model: We interpret the uplift predictions $\mathbf{\Delta\hat{p}}$ as independent variables and the unknown true uplifts $\mathbf{\Delta p}$ as dependent variables. The regression parameters α and β can be estimated as

$$\hat{\alpha} = \overline{\mathbf{\Delta p}} - \hat{\beta}\overline{\mathbf{\Delta}\hat{p}} \qquad \text{and} \qquad \hat{\beta} = \rho_{\mathbf{\Delta\hat{p}}\mathbf{\Delta p}} \frac{s_{\mathbf{\Delta p}}}{s_{\mathbf{\Delta\hat{p}}}}. \tag{5.11}$$

Typically, rich machine learning models—such as gradient boosting—exhibit very limited bias (Friedman, Hastie, Tibshirani, et al. 2000), so $\overline{\mathbf{\Delta p}} \approx \overline{\mathbf{\Delta}\hat{p}}$. For the estimate of β, we can leverage the fact that ϕ, our measure of the prediction model's quality, corresponds to the Pearson correlation coefficient ρ between predictions and true values. Furthermore, we do not need to scale the coefficient of correlation by the ratio of standard deviations, because there are no differences in the underlying real units. Therefore, the estimates in Equation (5.11) can be written as

$$\hat{\alpha} = (1 - \hat{\beta})\overline{\Delta\hat{p}} \qquad \text{and} \qquad \hat{\beta} = \phi. \tag{5.12}$$

Based on these estimates the objective function of the sales force scheduling model can be expressed as

$$\max v(\pi_\lambda) = \sum_{c \in C} \sum_{t \in T} \sum_{k \in K_c} \chi_k \left[(1 - \phi)\, \overline{\Delta\hat{p}} + \phi \Delta\hat{p}_k \right] y_{ckt}. \tag{5.13}$$

The logic underlying the quality-corrected predictions can be interpreted as follows: A visit at a location i to pitch project k is assumed to increase the probability of winning the project by the average predicted uplift of all projects in the test data $\overline{\Delta\hat{p}}$ (Equation 5.10). We then correct this baseline towards the predictions depending on the predictive model's quality ϕ.

5.7 Numerical Evaluation

This section presents the result of extensive numerical analyses that were carried out to evaluate the performance of our prescriptive scheduling approach relative to relevant benchmark policies, to assess the robustness of our approach, and to develop further structural and managerial insights.

For these analyses we utilize DAW's data as described in Section 5.4. Section 5.7.1 first describes our evaluation process. In Section 5.7.2 we then assess the value of our prescriptive policy relative to a predictive policy (that ignores the prediction model's quality and assumes perfect uplift information) and a zero-information policy (that does not account for uplift information) for a base case scenario. In the subsequent sections, we evaluate how the value of our prescriptive policy is affected by the heterogeneity of the projects' profits (Section 5.7.3), the sales force capacity (Section 5.7.4), and the quality of the prediction model (Section 5.7.5).

5.7.1 Evaluation process

In our numerical study we determine and evaluate the sales force schedules for individual sales forces s located at $S = 10$ different home bases in Germany. Each sales force consists of $|T_s|$ sales reps and serves an exclusive sales territory. Figure 5.11 illustrates our evaluation process.

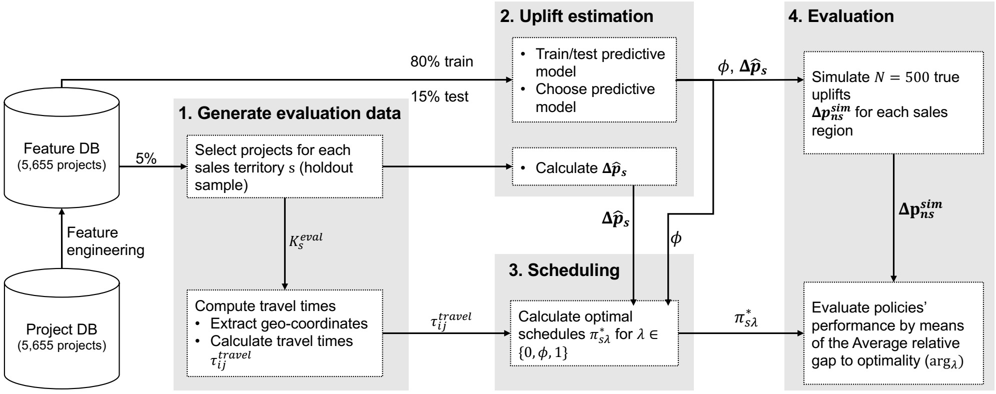

2. Uplift estimation
4. Evaluation
80% train
15% test
Train/test predictive model
Choose predictive model
ϕ, $\Delta\hat{p}_s$
Simulate $N = 500$ true uplifts Δp_{ns}^{sim} for each sales region
Feature DB (5,655 projects)
1. Generate evaluation data
5%
Select projects for each sales territory s (holdout sample)
Calculate $\Delta\hat{p}_s$
$\Delta\hat{p}_s$
ϕ
Δp_{ns}^{sim}
Feature engineering
K_s^{eval}
3. Scheduling
Project DB (5,655 projects)
Compute travel times
Extract geo-coordinates
Calculate travel times τ_{ij}^{travel}
τ_{ij}^{travel}
Calculate optimal schedules $\pi_{s\lambda}^*$ for $\lambda \in \{0, \phi, 1\}$
$\pi_{s\lambda}^*$
Evaluate policies' performance by means of the Average relative gap to optimality ($\arg_\lambda$)

Figure 5.11: Evaluation process

In a first step we generate the input data for our analysis. The data for our evaluation includes the features (described in Section 5.4) of a holdout sample of 500 projects from DAW's project data base that were neither used for training nor for testing of the predictive models. Prior to training and testing the predictive models, we assigned each project in the holdout sample to an evaluation set of a sales territory. Therefore, we assign the 50 projects closest to each home base s to the set K_s^{eval}, such that $|K_s^{eval}| = 50$.

We used Google Maps API[16] to retrieve the geocoordinates of the home bases 0_s and all customer locations of the 500 projects. Thereafter, we used the HERE API[17] to obtain the driving durations τ_{ij}^{travel} between any two locations i, j in sales territories $s = 1, ..., 10$.

In the second step, we estimated the uplifts $\mathbf{\Delta\hat{p}_s}$ for the projects of each sales territory s based on their features. For this, we used the procedure described in Section 5.5 based on the random forest model that led to the highest predictive performance on the test data (see Section 5.4).

In the third step we used the estimated uplifts $\mathbf{\Delta\hat{p}_s}$ $(s = 1, ...10)$ obtained in the previous step, as well as the model quality ϕ and the travel times τ_{ij}^{travel} to solve the scheduling problem (see Section 5.6) for each sales territory s and $\lambda \in \{0, \phi, 1\}$. $\lambda = \phi$ corresponds to our prescriptive policy as described in the previous section. $\lambda = 1$ implies a "predictive policy" that implicitly assumes perfect uplift predictions and maximizes the expected additional profits. In contrast, for $\lambda = 0$ uplift information is neglected; this "zero-information policy" determines schedules that maximize the sum of the projects' profits. The outcome of this step are the optimal schedules $\pi_{s\lambda}^*$ $(\lambda \in \{0, \phi, 1\})$ for each sales territory s.

In the final step we evaluate the performance of the three policies. Because we do not know the true uplifts for the 500 projects, we cannot directly compare the performance of the three policies. To overcome this problem and to ensure an objective performance comparison, we proceed as follows: We use simulation to generate $N = 500$ vectors of ("true") uplift realizations $\mathbf{\Delta p_{sn}^{sim}}$ for each sales territory s. We apply the Cholesky decomposition of the covariance matrix (e.g., Gentle (2009)) to ensure that the correlation between the estimated uplifts $\mathbf{\Delta\hat{p}_s}$ and the simulated true uplifts $\mathbf{\Delta p_{sn}^{sim}}$ is equal to ϕ. We then determine $v_{sn}(\pi_{s\lambda}^*)$ for $n = 1, .., 500$, which is the additional expected profit achieved when the optimal schedule $\pi_{s\lambda}^*$ of policy λ, determined based on the uplift predictions $\mathbf{\Delta\hat{p}_s}$, is executed, but the true uplifts are $\mathbf{\Delta p_{sn}^{sim}}$. Based on $v_{sn}(\pi_{s\lambda}^*)$ we define, as performance measure, the average relative optimality gap (arg) of policy λ:

$$\mathrm{arg}_\lambda = \frac{1}{S}\frac{1}{N}\sum_s\sum_n\left(1 - \frac{v_{sn}(\pi_{s\lambda}^*)}{\hat{v}_{sn}}\right), \tag{5.14}$$

[16] `https://cloud.google.com/maps-platform/?hl=de`

[17] `https://developer.here.com`

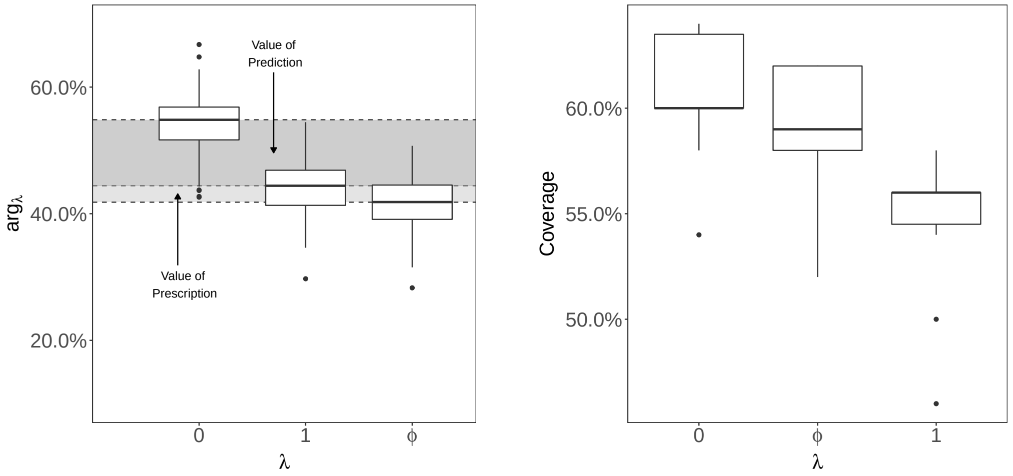
where $\hat{v}_{sn} = \sum_{k \in K_s^{eval}} \chi_k \max(0, \Delta p_{snk}^{sim})$ represents an upper bound on the expected additional profits in sales territory s for realization n of the true uplifts $\boldsymbol{\Delta p_{sn}^{sim}}$ if the company has perfect uplift predictions and sufficient capacity to visit all customers.

5.7.2 Value of the prescriptive policy – Base case

In our first analysis we evaluate the performance of the predictive, the prescriptive and the zero information policy for a base case scenario. We assume that each sales territory has a sales force of size $|T| = 5$ and we use the results of our best predictive model achieving a quality ϕ of 0.42 (see Section 5.4). Unfortunately, our partner DAW was not able to provide us with information on the profitability χ_k of the individual projects. For the purpose of this first analysis, we assume that projects are homogeneous in terms of their profits, i.e. $\chi_k = 1$ for all $k \in K_s^{Eval}$. In the next section we study, how profit heterogeneity impacts the policies' performance.

Figure 5.12 (a) displays the arg_λ for each policy λ and its distribution across the simulation runs. We clearly see that the predictive and prescriptive policies lead to substantially lower values of the arg than the zero-information benchmark ($\lambda = 0$). Also, the prescriptive policy that accounts for the model quality ($\lambda = \phi$) leads to a slightly higher performance than the predictive policy ($\lambda = 1$). In this base case, the major part of the performance increase can be attributed to the availability of a strong predictive model ("value of prediction"). In contrast, only a comparatively small additional increase is achieved by the prescriptive model—we term this increase "value of prescription".

Figure 5.12 (b) displays the relative coverage of the policies, i.e. the relative share of customers visited by each policy. Not surprisingly, the zero-information benchmark leads to the highest coverage because it maximizes the number of visits when all projects have the same profitability. The predictive policy, which relies only on the uplift predictions, leads to the lowest number of visits and hence the lowest coverage. Because the prescriptive policy ($\lambda = \phi$) trades off the number of visits and the projects' uplifts based on the quality of the predictive model ($\phi = 0.42$)—that is, how reliable the estimate of the uplifts are—it leads to a coverage that lies between the zero-information policy and the predictive policy. In comparison to the zero-information policy, the prescriptive policy sacrifices some visits and "invests" more travel time into projects with higher uplift predictions, but it does so in a more conservative manner than the predictive policy to account for the fact that the uplift predictions are uncertain. Since all projects have the same profitability, these results are rather intuitive. We do, however, observe that the results are plausible, and that the different policies lead to different sales force schedules that result in performance differences in terms of the *arg*. In the next section, we explore, how these results change when projects exhibit varying profitabilities.



(a) Average relative gap to optimality (*arg*)

(b) Coverage

Figure 5.12: Homogeneous profits ($|T| = 5$, $\phi = 0.42$)

5.7.3 Effect of the profit heterogeneity

This section studies the effect of heterogeneous project profitabilities on the relative performance of the three policies. For each project $k \in K_s^{Eval}$ in each sales territory s we draw a profit χ_k from a (symmetric) triangular distribution with mode $c = 1$, support $[a, b]$, and width $w = b - a$. $w = 0$ corresponds to the case of homogeneous profits discussed in the previous section. We vary the support of the distribution to obtain two levels of heterogeneity $w = 0.5$ and $w = 1$. We draw 20 profit realizations for each heterogeneity level and sales region to ensure the robustness of our analysis.

Figure 5.13 displays the policies' arg for different levels of heterogeneity. We observe that the predictive and prescriptive policies consistently outperform the zero-information benchmark ($\lambda = 0$) and that their arg decreases in the profit heterogeneity. Clearly, they are better able to prioritize projects with high profits and high uplifts.

The performance of the zero-information policy also increases at heterogeneity levels of $w = 0.5$ and $w = 1$, because this policy now prioritizes projects with higher profits. It does not, however, consider the uplift predictions and may therefore schedule visits for projects that have a high profitability, but low uplift. As a consequence, its arg is higher than those of the predictive and prescriptive policies. The advantage of the prescriptive policy increases as profit heterogeneity increases—this policy is less prone to schedule visits to high profit projects with too high (and incorrect) uplift predictions. In the initial setting ($w = 0$), the ability of the predictive and the prescriptive policies to leverage uplift forecasts accounts for a large part of the outperformance. However, with increasing profit heterogeneity the value of prescription increases and it becomes more important to account for the quality of the predictive model.

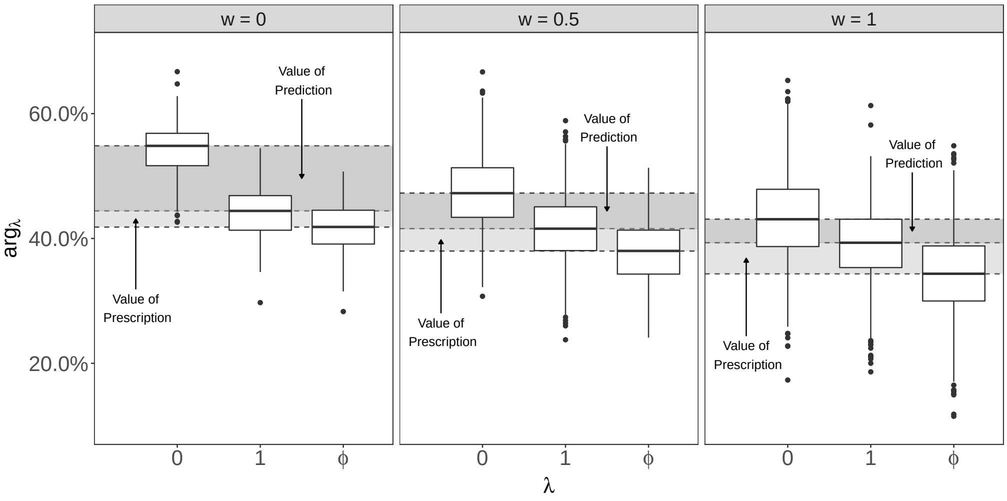


Figure 5.13: Average relative gap to optimality for varying levels of profit heterogeneity ($|T| = 5$, $\phi = 0.42$)

5.7.4 Effect of the sales force capacity

This section addresses the impact of the sales force capacity on the performance of the three policies. The size of the sales force corresponds to the possible coverage and serves as a measure for the scarcity of visits. We vary the size of the sales force in each region between 1 and 7 and report the performance of the three policies at a model quality of $\phi = 0.42$ and for different levels of profit heterogeneity. Figure 5.14 displays each policy's arg depending on the size of the sales force and the level of profit heterogeneity.

Compared to the zero-information policy, the predictive and prescriptive policies are able to leverage the sales force capacity more efficiently when capacity is scarce. Because the zero-information policy deploys sales representatives in an uninformed way, the performance improvement associated with an additional sales rep is almost constant. In contrast, the predictive and prescriptive policies exhibit decreasing marginal values of additional sales representatives. As capacity increases, more and more customers with lower expected additional profits are visited, which explains why their performances converge with that of the benchmark at high levels of sales force capacity.

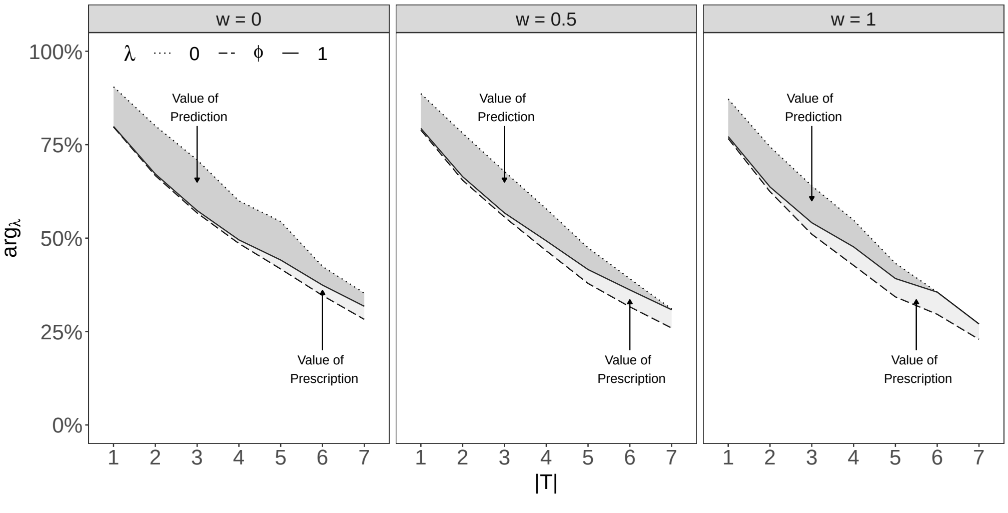


Figure 5.14: Average relative gap to optimality for varying levels of profit heterogeneity and sales force capacity ($\phi = 0.42$)

The prescriptive policy and the predictive policy lead to (almost) identical performances at low capacity levels. At medium levels of capacity the prescriptive policy leads to a higher performance than the predictive policy, and this difference increases as more capacity becomes available. These performance differences can be explained by the set of customers each policy chooses to visit. Figure 5.15 displays the Jaccard coefficient of similarity for all combinations of policies at different levels of capacity and for varying profit heterogeneity. Simply speaking, the Jaccard coefficient captures the number of identical customers both policies choose to visit relative to the total number of visits of both policies. At very low levels of capacity the predictive and the prescriptive policy schedule visits to a very similar set of customers—that is, customers with high predicted uplifts and high profit margins (for $w = 0.5$ and $w = 1$). In contrast, the zero-information policy chooses a different set of customers because it ignores the uplifts and focuses only on the trade-off between profit margins and travel times. The predictive and the prescriptive policies' choices diverge as more capacity becomes available: While the predictive policy continues to prioritize customers with with (a slightly) higher predicted uplifts or profits, the prescriptive policy hedges against prediction errors by balancing the expected additional profits with the sales effort, which is reflected by the time required to visit a customer. At medium and high levels of capacity, the prescriptive policy becomes more similar to the zero-information policy than the predictive policy and this effect is more pronounced when the projects' profit heterogeneity is large. This behavior of the prescriptive policy explains why "the value of prescription" as displayed in Figure 5.14 increases both in the sales force capacity and the profit heterogeneity.

Therefore, we conclude that the prescriptive policy should always be preferred over the predictive policy and that its benefits are particularly pronounced when profit heterogeneity is high and the sales force capacity is not severely constrained.

5.7.5 Effect of the model quality

Depending on the availability of data, its predictiveness, and the choice and configuration of a predictive model, companies will face varying qualities of predictions, which we capture with parameter ϕ. This section explores how the quality of the predictive model that is used for estimating the uplifts,

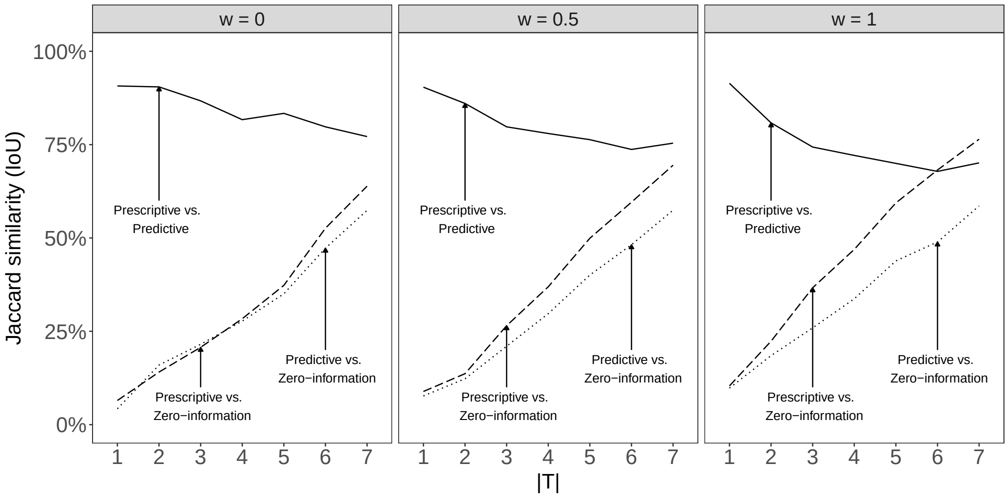


Figure 5.15: Similarity between the policies for varying levels of profit heterogeneity and sales force capacity ($\phi = 0.42$)

impacts the performance of the predictive and the prescriptive policies. Most importantly, we intend to understand, if a poor model quality renders our prescriptive policy ineffective in practice. To this end we simulate "true" uplifts for values of $\phi \in \{0, 0.05, \ldots, 0.6\}$ and evaluate the different policies as described in Section 5.7.1. Figure 5.16 plots the policies' arg for homogeneous customers, i.e. for $w = 0$, a sales force capacity of $|T| = 5$ and varying model qualities ϕ.

By definition, the prescriptive policy has a lower performance bound at $\phi = 0$ where its arg corresponds to that of the zero-information policy; its performance increases (almost linearly) in the model quality and consistently outperforms the predictive policy, which may, at very low quality levels, lead to a lower performance than the zero-information policy. The robust behavior of the prescriptive policy is particularly attractive, as it enables a company to leverage predictive models, even though they may exhibit a relatively low predictive quality. To illustrate this fact we highlight in Figure 5.16 the performance that would have been achieved with the various predictive models of Section 5.4 (see vertical dashed lines in Figure 5.16). We observe, for instance, that the prescriptive policy would lead to significant performance gains over the zero-information policy even if we used models based on support vector machines or logistic regressions—which lead to the lowest quality ϕ—to obtain uplift predictions.

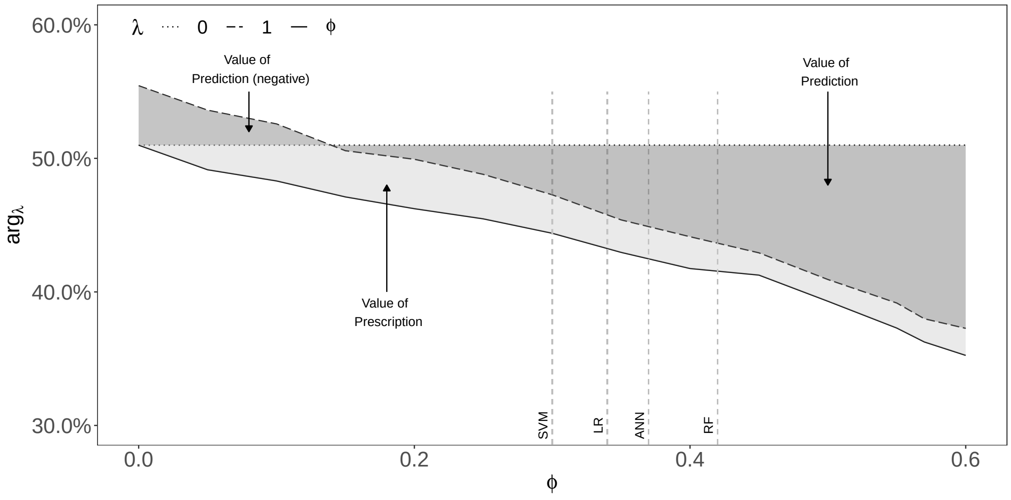


Figure 5.16: Average relative gap to optimality for varying model qualities ($|T| = 5$, $w = 0$)

The results of this analysis reinforce our previous findings and conjectures: Our prescriptive scheduling approach consistently outperforms the predictive and the zero-information policy. Its performance benefit increases in profit heterogeneity, it is particularly pronounced when capacity is not severely constrained *and* it dominates its contenders independent of the prediction model's quality.

5.8 Conclusion

Our research addresses a problem which is very common in many companies across different industries. Considering a large area with potential customers, we propose a novel data-driven approach to optimize the scheduling of sales representatives. To this end, we formalize the sales force scheduling problem at hand in the sense of a team orienteering problem. The uplifts associated with additional customer visits for each project are a crucial but often unknown input for this model. To estimate the uplifts, we propose a two-step approach. First, we train classification models to predict the probabilities of winning a project based on a rich feature set including the number of visits at a potential customer. Based on this prediction model, we estimate the success probability of a given project. Subsequently, we artificially increase the number of visits to estimate the uplift as the difference between the success probabilities prior and posterior the additional visit.

Building on the uplift predictions, we propose two different policies to solve the scheduling problem at hand. Solving the team orienteering problem based on the predicted uplifts posts a predictive policy to the sales force scheduling problem. Going further, we present a prescriptive policy that explicitly controls for the remaining uncertainty of our uplift predictions based on the performance of the predictive models.

We evaluate our approach using a real-world data set from a large European manufacturer of building paint and coating solutions. Performing an extensive numerical evaluation, we compare both policies to a benchmark approach that cannot leverage uplift predictions. We show that the predictive policy outperforms the benchmark in the problem setting at hand. However, its performance can be worse if the models predicting the uplift have a low performance. In contrast, the prescriptive policy shows to be the best policy in all evaluated settings. Its out-performance over the predictive policy is driven by customers heterogeneity, high sales force capacity, and low prediction quality.

6 Conclusion and Future Research Opportunities

Companies seek to establish data-driven decision cultures to leverage competitive advantages in terms of efficiency and effectiveness. However, the transformation towards data-rich environments will require significant changes in decision making processes across all industries. This dissertation set out to address these changes and to investigate potential opportunities to leverage prescriptive analytics in different planning problems. Ultimately, the goal of this work was to answer the guiding research question:

> *How can information systems combine state-of-the-art machine learning techniques and operations management modeling to provide prescriptive analytics models that are robust to prediction errors?*

To answer this question and to structure the thesis, four subordinate research questions (**RQ1 - RQ4**) have been defined in Chapter 1. This chapter summarizes and concludes the thesis and provides an outlook on future research opportunities. It is structured alongside the analytics stack and the subordinate research questions.

6.1 Summary

The first two articles (Chapters 2 and 3) focus on predictive models that serve as a mandatory backbone for powerful prescriptive models. To this end, we answer **RQ1** by identifying appropriate machine learning setups for two different prediction tasks. Additionally, Chapter 3 answeres **RQ2** by putting forward a data science toolbox for industrial analytics applications.

The third and fourth article focus on prescriptive analytics. First, Chapters 4 answers **RQ3** by introducing a data-driven solution to the searcher path problem. Subsequently, Chapters 5 answers **RQ4** by proposing a novel prescriptive approach to data-driven sales force scheduling that explicitly accounts for the underlying model uncertainty.

6.1.1 Predictive Analytics

Predictive models are an essential prerequisite for prescriptive analytics. Hence, Chapters 2 and 3 of this thesis aim at answering two research questions focusing on predictive analytics. The first goal is to identify appropriate machine learning setups for different predictions tasks (**RQ1**). The second objective is to derive and evaluate guidelines and best practices for the development of predictive analytics models (**RQ2**).

Building on state-of-the-art machine learning techniques, Chapter 2 showcases the development of a predictive model in the context of capacity planning and staffing at an IT consulting company. The presented approach improves the planning accuracy by more than 30% compared to traditional forecasting approaches and enables the company to ensure both high capacity utilization and service levels.

Chapter 3 focuses on predictive analytics applications in the manufacturing sector. More specifically, it presents a data science toolbox providing guidelines and best practices for modeling, feature engineering, and model interpretation to manufacturing decision-makers. Showcasing the application of this toolbox on a large data set from a manufacturing company yields twofold results. On the one hand, it becomes evident that plunging vast amounts of data into powerful new algorithms is not the silver bullet often promised by software companies. Rather, constant improvement, feature engineering, and consolidation are needed to complementary create value-generating predictive analytics applications.

6.1.2 Prescriptive Analytics

Simply using the improved forecasts provided by powerful predictive models enables decision-makers to generate additional business value in some situations. However, many complex tasks require elaborate operational planning procedures. Here, transforming additional information into valuable actions requires new planning algorithms. Therefore, Chapters 4 and 5 set out to answer two research questions focusing on prescriptive analytics. The first goal is to analyze how prescriptive analytics can be leveraged to provide decision support for manual processes in manufacturing settings (**RQ3**). Subsequently, it is analyzed how prescriptive models can account for the uncertainty of the underlying predictive models (**RQ4**).

The development of a prescriptive decision support system supporting individual workers is illustrated in Chapter 4. In particular, a leak detection process in a high-tech composite manufacturing environment is analyzed. Combining highly sensitive sensors with a scalable and generic feature extraction approach allows us to predict leak locations via statistical learning. Using advanced machine learning techniques improves the forecasts by almost 90% compared to a linear regression. Embedding the forecasts into the underlying searcher path problem enables us to develop predictive as well as prescriptive search policies. Comparing these policies against simple benchmarks shows that the prescriptive policy can dramatically reduce the search time as well as its variability. Consequently, it leads to faster and more stable processes.

While rapid advances in artificial intelligence research boost the predictive power of machine learning models, there remains a model uncertainty in most settings. Chapter 5 proposes a prescriptive approach that accounts for the fact that predictions are imperfect and that the arising uncertainty needs to be considered. More specifically, it presents a data-driven approach to sales-force scheduling. Based on a large data set, a model to predictive the benefit of additional sales effort—the “uplift”—is trained. Subsequently, the uplift predictions are embedded into the underlying team orienteering problem to determine optimized schedules. To this end, we proposed a predictive policy (that puts complete trust in the uplift estimations) as well as prescriptive policy (that accounts for the model quality) and compare them with a zero-information policy (that does not account for uplift forecasts).

6.2 Future Research Opportunities

As mentioned in Chapter 1, advanced analytics is an active and promising field of research offering a variety of future research opportunities across multiple fields. While the last years showed a tremendous development regarding new methods and applications, the potential for future research continues to be large.

6.2.1 Predictive Analytics

Even though an increasing number of companies starts leveraging predictive analytics for business applications, it is still an active area of research. Driven by advances in the machine learning community, a variety of new opportunities for predictive analytics arises.

Unstructured Data Continuously improving machine learning algorithms are pushing the limits of computer vision (LeCun, Bengio, and Hinton 2015; Szegedy et al. 2016), speech processing (Maas et al. 2015; Saon and Picheny 2017) and natural language understanding (Hirschberg and Manning 2015). While the applications of predictive analytics presented in this thesis rely mainly on structured and semi-structured data, tapping into the improvements mentioned above enables future work to leverage large pools of unstructured data. Thereby, existing predictive analytics applications can be improved, and new—even more powerful—potentials can be unlocked (Wang et al. 2018).

Interpretable Machine Learning Most machine learning models are currently evaluated and selected based on their predictive power (e.g., accuracy). While a model's predictive power is a necessary condition for predictive analytics applications, it is often not a sufficient one. As soon as safety-relevant decisions are made by machine learning techniques (Varshney and Alemzadeh 2017), not only a prediction but also an explanation for the model's decision is required. In the European Union, people significantly affected by automated decisions even have a right to an explanation (Goodman and Flaxman 2017). Against this backdrop, the research on machine learning interpretability and explanatory artificial intelligence is rapidly growing (Doshi-Velez and Kim 2017). Future work in the field of predictive analytics has to analyze, evaluate, and quantify the trade-off between predictive power and interpretability.

6.2.2 Prescriptive Analytics

Following Wedel and Kannan (2016), establishing data-driven decision cultures already provides companies with competitive advantages and has a significant impact on financial performance. To this end, prescriptive analytics are a key component for companies that want to leverage the full potential of the arising data ubiquity. Therefore, prescriptive analytics is an active and promising field of research offering a variety of future research opportunities.

Joint Estimation & Optimization Currently, most existing prescriptive analytics applications use predictive modeling to estimate parameters and to subsequently solve mathematical planning models to determine optimal policies. However, the idea of jointly solving the prediction as well as the planning problem has gained increasing attention in recent years (Bertsimas and Kallus 2014). Despite some promising first results (Bertsimas, Kallus, and Hussain 2016; Ban and Rudin 2018; Notz and Pibernik 2019; Meller, Taigel, and Pibernik 2018; Taigel and Meller 2018), this development is still in its infancy. Future research is required to quantify the advantages of this integrated approach in comparison to traditional sequential planning approaches.

Explainable Artificial Intelligence Explanatory artificial intelligence is a promising research stream not only for predictive analytics but also in the field of prescriptive analytics. On the one hand, interpretable prediction models provide human-readable decision rules that can be used to transform predictions into actions (Stefani and Zschech 2018). On the other hand, Bertsimas and Stellato (2018) propose to use machine learning models as "Voice of Optimization." To this end, interpretable models are used to solve continuous and mixed-integer optimization problems. Thereby, the logic behind the optimal solution can be extracted from the interpretable model.

Even though this new approach yields promising first results regarding interpretability (Bertsimas and Stellato 2018), as well as solution speed (Bertsimas and Stellato 2019), more research, is required to evaluate its performance on larger problem sets.

Deep Reinforcement Learning All approaches presented above rely on models of the underlying problem to find optimal policies for specific cases. However, these models are limited in more complex settings with large state spaces and a high degree of uncertainty. Deep reinforcement learning is a promising branch of machine learning research to solve problems in such settings. By interacting with the environment and learning from historical actions, this group of algorithms can identify prescriptive policies based on experience. Deep reinforcement learning has proven its ability to play a variety of games such as Atari games (Mnih et al. 2013; Mnih et al. 2015), Go (Silver et al. 2016), Starcraft (Vinyals et al. 2017), and Poker (Brown and Sandholm 2017, 2018, 2019), at a super-human level in recent years. Additionally, deep reinforcement learning has recently shown promising first results for operations management tasks such as vehicle routing (Nazari et al. 2018) and inventory replenishment (Gijsbrechts et al. 2018; Oroojlooyjadid, Snyder, and Takác 2016). However, those techniques require huge amounts of training data to learn optimal policies. While this requirement is easy to meet in controlled game settings, which can be simulated, it becomes a major challenge for real-world applications. Future research has to identify new ways to efficiently leverage the available data for the training of prescriptive reinforcement learning models. A promising avenue is to generate large synthetic data sets replicating the properties of the limited available data to train the agents. To this end, we plan to leverage recent advantages to unsupervised learning of complicated distributions. In particular, we plan to leverage generative adversarial networks (Goodfellow et al. 2014) and variational autoencoders (Kingma and Welling 2013). Both methods have shown promising first results in the generation of image (Mirza and Osindero 2014; Chen et al. 2016; Arjovsky, Chintala, and Bottou 2017; Hou et al. 2017), speech (Hsu, Zhang, and Glass 2017; Hsu et al. 2017), text (Semeniuta, Severyn, and Barth 2017; Yu et al. 2017), and tabular data (Park et al. 2018; Xu et al. 2019; Xu and

Veeramachaneni 2018).

6.3 Practical Implications

The continuing spread of sensor technology, connectivity, and declining costs of data storage lead the transformation towards data-rich environments. The works within this thesis provide four examples of how advanced analytics can improve operational planning and foster data-driven decision-making. The results of Chapters 2 and 5 show that firms from different sectors can leverage existing data via advanced analytics to improve their processes using existing data. In addition to existing data, an increasing number of sensors are embedded into various devices and machines in light of the IoT paradigm (Chen, Mao, and Liu 2014). Companies can use the newly available sensor data by means of predictive (Chapter 3) and prescriptive (Chapter 4) analytics. Apart from data availability, however, companies have to rethink existing processes and establish data-driven decision cultures to improve processes and gain a competitive advantage.

List of Figures

List of Tables

Bibliography

Aalst, Wil MP van der, Hajo A Reijers, Anton JMM Weijters, Boudewijn F van Dongen, AK Alves De Medeiros, Minseok Song, and HMW Verbeek. 2007. "Business Process Mining: An Industrial Application". *Information Systems* 32 (5): 713–732.

Aburto, Luis, and Richard Weber. 2007. "Improved Supply Chain Management Based on Hybrid Demand Forecasts". *Applied Soft Computing* 7 (1): 136–144.

Aggarwal, Charu C. 2015. *Data Mining.* Springer International Publishing.

Albers, Sönke, Kalyan Raman, and Nick Lee. 2015. "Trends in Optimization Models of Sales Force Management". *Journal of Personal Selling & Sales Management* 35 (4): 275–291.

Aldor-Noiman, Sivan, Paul D Feigin, and Avishai Mandelbaum. 2009. "Workload Forecasting for a Call Center: Methodology and a Case Study". *The Annals of Applied Statistics*: 1403–1447.

Alpaydin, Ethem. 2010. *Introduction to Machine Learning.* 2nd. The MIT Press.

Angalakudati, Mallik, Siddharth Balwani, Jorge Calzada, Bikram Chatterjee, Georgia Perakis, Nicolas Raad, and Joline Uichanco. 2014. "Business Analytics for Flexible Resource Allocation Under Random Emergencies". *Management Science* 60 (6): 1552–1573.

Archetti, Claudia, Francesco Carrabs, and Raffaele Cerulli. 2018. "The Set Orienteering Problem". *European Journal of Operational Research* 267 (1): 264–272.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou. 2017. "Wasserstein Generative Adversarial Networks". In *Proceedings of the International Conference on Machine Learning.*

Armstrong, J Scott. 1978. *Long-Range Forecasting: From Crystal Ball to Computer.* John Wiley & Sons Canada, Limited.

Arymurthy, Aniati Murni, and Darmatasia. 2016. "Predicting the Status of Water Pumps Using Data Mining Approach". In *Proceedings of the International Workshop on Big Data and Information Security (IWBIS).*

Aviv, Yossi. 2007. "On the Benefits of Collaborative Forecasting Partnerships Between Retailers and Manufacturers". *Management Science* 53 (5): 777–794.

Ban, Gah-Yi, and Cynthia Rudin. 2018. "The Big Data Newsvendor: Practical Insights From Machine Learning". *Operations Research* 67 (1): 90–108.

Barton, Dominic, and D Court. 2012. "Making Advanced Analytics Work for You". *Harvard business review* 90 (10): 78–83.

Basili, Victor R. 1996. "The Role of Experimentation in Software Engineering: Past, current, and future". In *Proceedings of the 18th International Conference on Software Engineering.*

Bergstra, James S, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. "Algorithms for Hyper-Parameter Optimization". In *Proceedings of Advances in Neural Information Processing Systems (NIPS).*

Bergstra, James, and Yoshua Bengio. 2012. "Random Search for Hyper-Parameter Optimization". *Journal of Machine Learning Research* 13:281–305.

Bertsimas, Dimitris, and Nathan Kallus. 2014. "From Predictive to Prescriptive Analytics". *arXiv preprint arXiv:1402.5481.*

Bertsimas, Dimitris, Nathan Kallus, and Amjad Hussain. 2016. "Inventory Management in the Era of Big Data". *Production and Operations Management* 25 (12): 2006–2009.

Bertsimas, Dimitris, and Bartolomeo Stellato. 2019. "Online Mixed-Integer Optimization in Milliseconds". *arXiv preprint arXiv:1907.02206.*

— . 2018. "The Voice of Optimization". *arXiv preprint arXiv:1812.09991.*

Beutel, Anna Lena, and Stefan Minner. 2012. "Safety Stock Planning Under Causal Demand Forecasting". *International Journal of Production Economics* 140 (2): 637–645.

Bleakie, Alexander, and Dragan Djurdjanovic. 2013. "Analytical Approach to Similarity-Based Prediction of Manufacturing System Performance". *Computers in Industry* 64 (6): 625–633.

Bontempi, Gianluca, Souhaib Ben Taieb, and Yann-Aël Le Borgne. 2012. "Machine Learning Strategies for Time Series Forecasting". In *Proceedings of the European Business Intelligence Summer School (eBISS).*

Breiman, Leo. 2001a. "Random Forests". *Machine learning* 45 (1): 5–32.

— . 2001b. "Statistical Modeling: The Two Cultures". *Statistical Science* 16 (3): 199–215.

Breuker, Dominic, Martin Matzner, Patrick Delfmann, and Jörg Becker. 2016. "Comprehensible Predictive Models for Business Processes". *MIS Quarterly* 40 (4).

Brown, Noam, and Tuomas Sandholm. 2017. "Libratus: The Superhuman AI for No-Limit Poker". In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI).*

— . 2018. "Superhuman AI for Heads-Up No-Limit Poker: Libratus Beats Top Professionals". *Science* 359 (6374): 418–424.

— . 2019. "Superhuman AI for Multiplayer Poker". *Science.*

Burrell, Jenna. 2016. "How the Machine 'Thinks': Understanding Opacity in Machine Learning Algorithms". *Big Data & Society* 3 (1).

Campbell, Ann M., Michel Gendreau, and Barrett W. Thomas. 2011. "The Orienteering Problem With Stochastic Travel and Service Times". *Annals of Operations Research* 186 (1).

Cankurt, Selcuk. 2016. "Tourism Demand Forecasting Using Ensembles of Regression Trees". In *Proceedings of the International Conference on Intelligent Systems (IS).*

Carbonneau, Real, Kevin Laframboise, and Rustam Vahidov. 2008. “Application of Machine Learning Techniques for Supply Chain Demand Forecasting”. *European Journal of Operational Research* 184 (3): 1140–1154.

Caruana, Rich, Nikos Karampatziakis, and Ainur Yessenalina. 2008. “An Empirical Evaluation of Supervised Learning in High Dimensions”. In *Proceedings of the 25th International Conference on Machine Learning (ICML).*

Castelli, Mauro, Leonardo Vanneschi, Luca Manzoni, and Aleš Popovič. 2016. “Semantic Genetic Programming for Fast and Accurate Data Knowledge Discovery”. *Swarm and Evolutionary Computation* 26:1–7.

Cater-Steel, Aileen. 2009. “IT Service Departments Struggle to Adopt a Service-Oriented Philosophy”. *International Journal of Information Systems in the Service Sector (IJISSS)* 1 (2): 69–77.

Cater-Steel, Aileen, and Neil McBride. 2007. “IT Service Management Improvement - Actor Network Perspective”. In *Proceedings of the 15th European Conference on Information Systems (ECIS).*

Cater-Steel, Aileen, Wui-Gee Tan, and Mark Toleman. 2006. “Challenge of Adopting Multiple Process Improvement Frameworks”. In *Proceedings of 14th European Conference on Information Systems (ECIS).*

Chae, Bonsug (Kevin), and David L. Olson. 2013. “Business Analytics for Supply Chain: A Dynamic-Capabilities Framework”. *International Journal of Information Technology & Decision Making* 12 (01): 9–26.

Chakraborty, Kanad, Kishan Mehrotra, Chilukuri K Mohan, and Sanjay Ranka. 1992. “Forecasting the Behavior of Multivariate Time Series Using Neural Networks”. *Neural networks* 5 (6): 961–970.

Chapman, Pete, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rudiger Wirth. 2000. *CRISP-DM 1.0 Step-By-Step Data Mining Guide.* Tech. rep. The CRISP-DM consortium.

Chen, Mu-Chen, Cheng-Lung Huang, Kai-Ying Chen, and Hsiao-Pin Wu. 2005. “Aggregation of Orders in Distribution Centers Using Data Mining”. *Expert Systems with Applications* 28 (3): 453–460.

Chen, Hsinchun, Roger HL Chiang, and Veda C Storey. 2012. "Business Intelligence and Analytics: From Big Data to Big Impact." *MIS quarterly* 36 (4): 1165–1188.

Chen, Min, Shiwen Mao, and Yunhao Liu. 2014. "Big Data: A Survey". *Mobile networks and applications* 19 (2): 171–209.

Chen, Tianqi, and Carlos Guestrin. 2016. "XGBoost: A Scalable Tree Boosting System". In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Chen, Tianqi, and Tong He. 2015. "Higgs Boson Discovery With Boosted Trees". In *Proceedings of the NIPS 2014 Workshop on High-Energy Physics and Machine Learning.*

Chen, Xi, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. "Infogan: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets". In *Proceedings of Advances in Neural Information Processing Systems (NIPS.*

Choudhary, A. K., J. A. Harding, and M. K. Tiwari. 2008. "Data Mining in Manufacturing: A Review Based on the Kind of Knowledge". *Journal of Intelligent Manufacturing* 20 (5): 501–521.

Coussement, Kristof, Stefan Lessmann, and Geert Verstraeten. 2017. "A Comparative Analysis of Data Preparation Algorithms for Customer Churn Prediction: A Case Study in the Telecommunication Industry". *Decision Support Systems* 95:27–36.

Cox, David R. 1958. "The Regression Analysis of Binary Sequences". *Journal of the Royal Statistical Society* 20 (2): 215–242.

Cui, Ruomeng, Gad Allon, Achal Bassamboo, and Jan A. Van Mieghem. 2015. "Information Sharing in Supply Chains: An Empirical and Theoretical Valuation". *Management Science* 61 (11): 2803–2824.

Davenport, Thomas H. 2006. "Competing on Analytics". *Harvard Business Review* 84 (1): 98.

Davis, Jason V, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. 2007. "Information-Theoretic Metric Learning". In *Proceedings of the 24th International Conference on Machine Learning.*

Dhar, Vasant. 2013. “Data Science and Prediction”. *Communications of the ACM* 56 (12): 64–73.

Dhawan, Rajat, Bernd Hei, and Kevin Laczkowski. 2018. *Disruptive Forces in the Industrial Sectors.* Tech. rep. McKinsey & Company.

Diakopoulos, Nicholas. 2014. “Algorithmic Accountability Reporting: On the Investigation of Black Boxes”. *Tow Center for Digital Journalism.*

Diao, Yixin, Hani Jamjoom, and David Loewenstern. 2009. “Rule-Based Problem Classification in It Service Management”. In *Proceedings of the IEEE International Conference on Cloud Computing.*

Dieulle, L, C Bérenguer, a Grall, and M Roussignol. 2003. “Sequential Condition-Based Maintenance Scheduling for a Deteriorating System”. *European Journal of Operational Research* 150, no. 2 (): 451–461.

Disterer, Georg. 2012. “Why Firms Seek ISO 20000 Certification-A Study of ISO 20000 Adoption.” In *Proceedings of the 20th European Conference on Information Systems (ECIS).*

Domingos, Pedro. 2012. “A Few Useful Things to Know About Machine Learning”. *Communications of the ACM* 55 (10): 78–87.

Doshi-Velez, Finale, and Been Kim. 2017. “Towards a Rigorous Science of Interpretable Machine Learning”. *arXiv preprint arXiv:1702.08608.*

Dresner Advisory Services. 2017. *2017 Big Data Analytics Market Study.* Tech. rep. Dresner Advisory Services, LLC.

Eickemeyer, Steffen C., Felix Herde, Pratik Irudayaraj, and Peter Nyhuis. 2014. “Decision Models for Capacity Planning in a Regeneration Environment”. *International Journal of Production Research* 52 (23): 7007–7026.

Elmagarmid, Ahmed K, Panagiotis G Ipeirotis, and Vassilios S Verykios. 2007. “Duplicate Record Detection: A Survey”. *IEEE Transactions on knowledge and data engineering* 19 (1): 1–16.

Evans, James R, and Carl H Lindner. 2012. “Business Analytics: The Next Frontier for Decision Sciences”. *Decision Line* 43 (2): 4–6.

Evers, Lanah, Kristiaan Glorie, Suzanne van der Ster, Ana Isabel Barros, and Herman Monsuur. 2014. “A Two-Stage Approach to the Orienteering Problem With Stochastic Weights”. *Computers & Operations Research* 43:248–260.

Fayyad, Usama, Gregory Piatetsky-Shapiro, and Padhraic Smyth. 1996. “From Data Mining to Knowledge Discovery in Databases”. *AI magazine* 17 (3): 37.

Feillet, Dominique, Pierre Dejax, and Michel Gendreau. 2005. “Traveling Salesman Problems With Profits”. *Transportation Science* 39 (2).

Flach, Peter A. 2003. “The Geometry of ROC Space: Understanding Machine Learning Metrics Through ROC Isometrics”. In *Proceedings of the 20th International Conference on Machine Learning.*

Flath, Christoph M, and Nikolai Stein. 2018. “Towards a Data Science Toolbox for Industrial Analytics Applications”. *Computers in Industry* 94:16–25.

Foresee, F Dan, and Martin T Hagan. 1997. “Gauss-Newton Approximation to Bayesian Learning”. In *Proceedings of the International Conference on Neural Networks.*

Foster, George. 1977. “Quarterly Accounting Data: Time-Series Properties and Predictive-Ability Results”. *Accounting Review*: 1–21.

Foster, Jared C., Jeremy M G Taylor, and Stephen J. Ruberg. 2011. “Subgroup Identification From Randomized Clinical Trial Data”. *Statistics in Medicine* 30 (24).

Friedman, Jerome H. 2002. “Stochastic Gradient Boosting”. *Computational Statistics & Data Analysis* 38 (4): 367–378.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2001. *The Elements of Statistical Learning.* Vol. 1. Springer series in statistics Springer, Berlin.

Friedman, Jerome, Trevor Hastie, Robert Tibshirani, et al. 2000. “Additive Logistic Regression: A Statistical View of Boosting”. *The annals of statistics* 28 (2): 337–407.

Galton, Francis. 1886. “Regression Towards Mediocrity in Hereditary Stature.” *The Journal of the Anthropological Institute of Great Britain and Ireland* 15:246–263.

Gelman, Andrew, Aleks Jakulin, Maria Grazia Pittau, and Yu-Sung Su. 2008. “A weakly informative default prior distribution for logistic and other regression models”. *The Annals of Applied Statistics* 2 (4): 1360–1383.

General Electric, Accenture. 2015. *Industrial Internet Insights Report.* Tech. rep. General Electric, Accenture.

Gentle, James E. 2009. *Computational Statistics.* Vol. 308. Springer.

Geurts, Pierre. 2002. “Contributions to Decision Tree Induction: Bias/Variance Tradeoff and Time Series Classification”. PhD thesis, University of Liège Belgium.

Giesecke, Kay, Gui Liberali, Hamid Nazerzadeh, J George Shanthikumar, and Chung Piaw Teo. 2018. “Call for Papers—Management Science—Special Issue on Data-Driven Prescriptive Analytics”. *Management Science* 64 (6): 2972–2972.

Gijsbrechts, Joren, Robert N Boute, Jan A Van Mieghem, and Dennis Zhang. 2018. “Can Deep Reinforcement Learning Improve Inventory Management? Performance and Implementation of Dual Sourcing-Mode Problems”. *Performance and Implementation of Dual Sourcing-Mode Problems (December 17, 2018).*

Goby, Niklas, Tobias Brandt, Stefan Feuerriegel, and Dirk Neumann. 2016. “Business Intelligence for Business Processes: The Case of IT Incident Management.” In *Proceedings of the 24th European Conference on Information Systems (ECIS).*

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. “Generative Adversarial Nets”. In *Proceedings of Advances in Neural Information Processing Systems (NIPS).*

Goodman, Bryce, and Seth Flaxman. 2017. “European Union Regulations on Algorithmic Decision-Making and a “right to Explanation””. *AI Magazine* 38 (3): 50–57.

Grabocka, Josif, Martin Wistuba, and Lars Schmidt-Thieme. 2015. "Scalable Classification of Repetitive Time Series Through Frequencies of Local Polynomials". *IEEE Transactions on Knowledge and Data Engineering* 27 (6): 1683–1695.

Grall, A., C. Bérenguer, and L. Dieulle. 2002. "A Condition-Based Maintenance Policy for Stochastically Deteriorating Systems". *Reliability Engineering & System Safety* 76, no. 2 (): 167–180.

Griebel, Matthias, Alexander Dürr, and Nikolai Stein. 2019. "Applied Image Recognition: Guidelines for Using Deep Learning Models in Practice". In *Proceedings of the 14th International Conference on Wirtschaftsinformatik (WI).*

Gualtieri, Mike, Stephen Powers, and Vivian Brown. 2015. *The Forrester Wave™: Big Data Predictive Analytics Solutions.* Tech. rep.

Guelman, Leo, Montserrat Guillén, and Ana M. Pérez-Marín. 2015. "Uplift Random Forests". *Cybernetics and Systems* 46 (3-4).

Gunawan, Aldy, Hoong Chuin Lau, and Pieter Vansteenwegen. 2016. "Orienteering Problem: A Survey of Recent Variants, solution approaches and applications". *European Journal of Operational Research* 255 (2): 315–332.

Guosheng, Hu, and Zhang Guohong. 2008. "Comparison on Neural Networks and Support Vector Machines in Suppliers' Selection". *Journal of Systems Engineering and Electronics* 19 (2): 316–320.

Gupta, Saurav, Nitin Anand Shrivastava, Abbas Khosravi, and Bijaya Ketan Panigrahi. 2016. "Wind Ramp Event Prediction With Parallelized Gradient Boosted Regression Trees". In *Proceedings of the International Joint Conference on Neural Networks (IJCNN).*

Gurobi Optimization, Inc. 2016. *Gurobi Optimizer Reference Manual.*

El-Hajj, Racha, Duc-Cuong Dang, and Aziz Moukrim. 2016. "Solving the Team Orienteering Problem With Cutting Planes". *Computers & Operations Research* 74:21–30.

Halevy, Alon, Peter Norvig, and Fernando Pereira. 2009. "The Unreasonable Effectiveness of Data". *IEEE Intelligent Systems* 24 (2): 8–12.

Hansotia, Behram, and Brad Rukstales. 2002. “Incremental Value Modeling”. *Journal of Interactive Marketing* 16 (3): 35–46.

Hastie, Trevor. 2004. *Stanford University Presentation on Boosting.*

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2013. *The Elements of Statistical Learning.* 745. Springer New York.

Hauser, Matthias, Daniel Zügner, Christoph M Flath, and Frederic Thiesse. 2015. “Pushing the Limits of RFID: Empowering RFID-based Electronic Article Surveillance With Data Analytics Techniques”. In *Proceedings of the 36th International Conference on Information Systems (ICIS).*

Heimbach, Irina, Daniel S. Kostyra, and Oliver Hinz. 2015. “Marketing Automation”. *Business & Information Systems Engineering* 57 (2): 129.

Hevner, Alan R, Salvatore T March, Jinsoo Park, and Sudha Ram. 2004. “Design Science in Information Systems Research”. *MIS Quarterly: Management Information Systems* 28 (1): 75–105.

Hipp, D Richard, and D Kennedy. 2016. *SQLite.*

Hirschberg, Julia, and Christopher D Manning. 2015. “Advances in Natural Language Processing”. *Science* 349 (6245): 261–266.

Ho, Jyh-Wen, and Chih-Chiang Fang. 2013. “Production Capacity Planning for Multiple Products Under Uncertain Demand Conditions”. *International Journal of Production Economics* 141 (2): 593–604.

Ho, SL, M Xie, and TN Goh. 2002. “A Comparative Study of Neural Network and Box-Jenkins ARIMA Modeling in Time Series Prediction”. *Computers & Industrial Engineering* 42 (2): 371–375.

Hochstein, Axel, Gerrit Tamm, and Walter Brenner. 2005. “Service Oriented IT Management: Benefit, cost and success factors”. In *Proceedings of 13th European Conference on Information Systems (ECIS).*

Hodges, Joseph L, Erich L Lehmann, et al. 1952. “The Use of Previous Experience in Reaching Statistical Decisions”. *The Annals of Mathematical Statistics* 23 (3): 396–407.

Holsapple, Clyde, Anita Lee-Post, and Ram Pakath. 2014. “A Unified Foundation for Business Analytics”. *Decision Support Systems* 64:130–141.

Hotelling, Harold. 1933. "Analysis of a Complex of Statistical Variables Into Principal Components." *Journal of educational psychology* 24 (6): 417.

Hothorn, Torsten, Kurt Hornik, and Achim Zeileis. 2006. "Unbiased Recursive Partitioning: A Conditional Inference Framework". *Journal of Computational and Graphical Statistics* 15 (3): 651–674.

Hou, Xianxu, Linlin Shen, Ke Sun, and Guoping Qiu. 2017. "Deep Feature Consistent Variational Autoencoder". In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV).*

Hsu, Chin-Cheng, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang. 2017. "Voice Conversion From Unaligned Corpora Using Variational Autoencoding Wasserstein Generative Adversarial Networks". *arXiv preprint arXiv:1704.00849.*

Hsu, Wei-Ning, Yu Zhang, and James Glass. 2017. "Learning Latent Representations for Speech Generation and Transformation". *arXiv preprint arXiv:1704.04222.*

Hu, Ke, Ashfaqur Rahman, Hari Bhrugubanda, and Vijay Sivaraman. 2017. "HazeEst: Machine Learning Based Metropolitan Air Pollution Estimation From Fixed and Mobile Sensors". *IEEE Sensors Journal* 17 (11): 3517–3525.

Huang, Tingliang, and Jan A. Van Mieghem. 2014. "Clickstream Data and Inventory Management: Model and Empirical Analysis". *Production and Operations Management* 23 (3): 333–347.

Huang, Wei, Yoshiteru Nakamori, and Shou-Yang Wang. 2005. "Forecasting Stock Market Movement Direction With Support Vector Machine". *Computers & Operations Research* 32 (10): 2513–2522.

Hull, Giles Hindle, Martin Kunc, Michael Mortensen, Asil Oztekin, and Richard Vidgen. 2018. "Call for Papers—Business Analytics: Defining the Field and Identifying a Research Agenda". *European Journal of Operations Research.*

Hyndman, Rob J, and George Athanasopoulos. 2014. *Forecasting: Principles and Practice.* OTexts.

Hyndman, Rob J, and Anne B Koehler. 2006. "Another Look at Measures of Forecast Accuracy". *International journal of forecasting* 22 (4): 679–688.

Iden, Jon, and Tom Roar Eikebrokk. 2013. “Implementing IT Service Management: A Systematic Literature Review”. *International Journal of Information Management* 33 (3): 512–523.

Iden, Jon, and Lars Langeland. 2010. “Setting the Stage for a Successful ITIL Adoption: A Delphi Study of IT Experts in the Norwegian Armed Forces”. *Information systems management* 27 (2): 103–112.

Ilhan, Taylan, Seyed M. R. Iravani, and Mark S. Daskin. 2008. “The Orienteering Problem With Stochastic Profits”. *IIE Transactions* 40 (4): 406–421.

Imgrund, Florian, Marcus Fischer, Christian Janiesch, and Axel Winkelmann. 2017. “Managing the Long Tail of Business Processes”. In *Proceedings of the 25th European Conference on Information Systems (ECIS).*

IoT Analytics. 2016. *Industrial Analytics 2016/2017: The Current State of Data Analytics Usage in Industrial Companies.* Tech. rep. IoT Analytics.

Jane, Chin-Chia, and Yih-Wenn Laih. 2005. “A Clustering Algorithm for Item Assignment in a Synchronized Zone Order Picking System”. *European Journal of Operational Research* 166 (2): 489–496.

Kaggle.com. 2016. *Bosch Production Line Performance.*

Kanter, James Max, and Kalyan Veeramachaneni. 2015. “Deep Feature Synthesis: Towards Automating Data Science Endeavors”. In *Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA).*

Karabuk, Suleyman, and David S. Wu. 2003. “Coordinating Strategic Capacity Planning in the Semiconductor Industry”. *Operations Research* 51 (6): 839–849.

Kaufman, Leonard, and Peter J Rousseeuw. 2009. *Finding Groups in Data: An Introduction to Cluster Analysis.* Vol. 344. John Wiley & Sons.

Ke, Liangjun, Zongben Xu, Zuren Feng, Ke Shang, and Xueming Qian. 2013. “Proportion-Based Robust Optimization and Team Orienteering Problem With Interval Data”. *European Journal of Operational Research* 226 (1): 19–31.

Khashei, Mehdi, and Mehdi Bijari. 2011. “A Novel Hybridization of Artificial Neural Networks and ARIMA Models for Time Series Forecasting”. *Applied Soft Computing* 11 (2): 2664–2675.

Kim, Taehoon, Dongeun Lee, Jaesik Choi, Anna Spurlock, Alex Sim, Annika Todd, and Kesheng Wu. 2015. “Extracting Baseline Electricity Usage With Gradient Tree Boosting”.

Kingma, Diederik P, and Max Welling. 2013. “Auto-Encoding Variational Bayes”. *arXiv preprint arXiv:1312.6114*.

Kohavi, Ron, Neal J Rothleder, and Evangelos Simoudis. 2002. “Emerging Trends in Business Analytics”. *Communications of the ACM* 45 (8): 45–48.

Krenzer, Adrian, Nikolai Stein, Matthias Griebel, and Christoph M Flath. 2019. “Augmented Intelligence for Quality Control of Manual Assembly Processes Using Industrial Wearable Systems”. In *Proceedings of the 27th International Conference on Information Systems (ICIS).*

Krollner, Bjoern, Bruce Vanstone, and Gavin Finnie. 2010. “Financial Time Series Forecasting With Machine Learning Techniques: A Survey”. In *Proceedings of the European Symposium on Artificial Neural Networks: Computational and Machine Learning.*

LaValle, Steve, Eric Lesser, Rebecca Shockley, Michael S Hopkins, and Nina Kruschwitz. 2011. “Big Data, analytics and the path from insights to value”. *MIT sloan management review* 52 (2): 21.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. “Deep Learning”. *Nature* 521 (7553): 436.

Lee, Jay, Edzel Lapira, Behrad Bagheri, and Hung-an Kao. 2013. “Recent Advances and Trends in Predictive Manufacturing Systems in Big Data Environment”. *Manufacturing Letters* 1 (1): 38–41.

Létourneau, Sylvain, Fazel Famili, and Stan Matwin. 1999. “Data Mining to Predict Aircraft Component Replacement”. *IEEE Intelligent Systems* 14, no. 6 (): 59–66.

Li, Xiaonan, and Sigurdur Olafsson. 2005. “Discovering Dispatching Rules Using Data Mining”. *Journal of Scheduling* 8 (6): 515–527.

Lustig, Irv, Brenda Dietrich, Christer Johnson, and Christopher Dziekan. 2010. “The Analytics Journey”. *Analytics Magazine*, no. 6: 11–13.

Maas, Andrew, Ziang Xie, Dan Jurafsky, and Andrew Ng. 2015. “Lexicon-Free Conversational Speech Recognition With Neural Networks”. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

Maaten, Laurens van der, and Geoffrey Hinton. 2008. “Visualizing Data Using T-Sne”. *Journal of Machine Learning Research* 9 (Nov): 2579–2605.

MacQueen, James, et al. 1967. “Some Methods for Classification and Analysis of Multivariate Observations”. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability.*

Maksai, Andrii, Jasmina Bogojeska, and Dorothea Wiesmann. 2014. “Hierarchical Incident Ticket Classification With Minimal Supervision”. In *Proceedings of the IEEE International Conference on Data Mining (ICDM).*

Manchanda, Puneet, and Pradeep K Chintagunta. 2004. “Responsiveness of Physician Prescription Behavior to Salesforce Effort: An Individual Level Analysis”. *Marketing Letters* 15 (2-3): 129–145.

Mangal, Ankita, and Nishant Kumar. 2016. “Using Big Data to Enhance the Bosch Production Line Performance: A Kaggle Challenge”. *arXiv preprint arXiv:1701.00705.*

Manyika, James, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela H Byers. 2011. *Big Data: The Next Frontier for Innovation, Competition, and Productivity.* Tech. rep. McKinsey Global Institute.

Marrone, Mauricio, and Lutz M Kolbe. 2011. “Impact of IT Service Management Frameworks on the IT Organization”. *Business & Information Systems Engineering* 3 (1): 5–18.

Martens, David, and Foster Provost. 2014. “Explaining Data-Driven Document Classifications”. *MIS Quarterly* 38 (1): 73–99.

Maurya, Abhinav. 2016. “Bayesian Optimization for Predicting Rare Internal Failures in Manufacturing Processes”. In *Proceedings of the IEEE International Conference on Big Data (Big Data).*

Mayr, Andreas, Harald Binder, Olaf Gefeller, and Matthias Schmid. 2014. “The Evolution of Boosting Algorithms”. *Methods of Information in Medicine* 53 (06): 419–427.

Mayrink, Victor, and Henrique S Hippert. 2016. “A Hybrid Method Using Exponential Smoothing and Gradient Boosting for Electrical Short-Term Load Forecasting”. In *Proceedings of the IEEE Latin American Conference on Computational Intelligence (LA-CCI).*

McAfee, Andrew, Erik Brynjolfsson, Thomas H Davenport, DJ Patil, and Dominic Barton. 2012. “Big Data: The Management Revolution”. *Harvard Business Review* 90 (10): 61–67.

McBride, Neil. 2009. “Exploring Service Issues Within the IT Organisation: Four Mini-Case Studies”. *International journal of information management* 29 (3): 237–243.

Meller, Jan, Fabian Taigel, and Richard Pibernik. 2018. “Prescriptive Analytics for Inventory Management: A Comparison of New Approaches”. *Available at SSRN 3229105.*

Mirza, Mehdi, and Simon Osindero. 2014. “Conditional Generative Adversarial Nets”. *arXiv preprint arXiv:1411.1784.*

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. “Playing Atari With Deep Reinforcement Learning”. *arXiv preprint arXiv:1312.5602.*

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. “Human-Level Control Through Deep Reinforcement Learning”. *Nature* 518 (7540): 529.

Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar. 2012. *Foundations of Machine Learning.* MIT press.

Moro, Sérgio, Paulo Cortez, and Paulo Rita. 2014. “A Data-Driven Approach to Predict the Success of Bank Telemarketing”. *Decision Support Systems* 62:22–31.

Mortenson, Michael J, Neil F Doherty, and Stewart Robinson. 2015. “Operational Research From Taylorism to Terabytes: A Research Agenda for the Analytics Age”. *European Journal of Operational Research* 241 (3): 583–595.

Mosallam, Ahmed, Kamal Medjaher, and Noureddine Zerhouni. 2016. “Data-Driven Prognostic Method Based on Bayesian Approaches for Direct Remaining Useful Life Prediction”. *Journal of Intelligent Manufacturing* 27 (5): 1037–1048.

Müller, Oliver, Iris Junglas, Jan vom Brocke, and Stefan Debortoli. 2016. “Utilizing Big Data Analytics for Information Systems Research: Challenges, Promises and Guidelines”. *European Journal of Information Systems* 25 (4).

Murtagh, Fionn, and Pedro Contreras. 2012. “Algorithms for Hierarchical Clustering: An Overview”. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2 (1): 86–97.

Nassif, Ali Bou. 2016. “Short Term Power Demand Prediction Using Stochastic Gradient Boosting”. In *Proceedings of the 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA).*

Nazari, Mohammadreza, Afshin Oroojlooy, Lawrence Snyder, and Martin Takác. 2018. “Reinforcement Learning for Solving the Vehicle Routing Problem”. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 9839–9849.

Notz, Pascal, and Richard Pibernik. 2019. “Prescriptive Analytics for Flexible Capacity Management”. *Available at SSRN 3387866.*

Oroojlooyjadid, Afshin, Lawrence Snyder, and Martin Takác. 2016. “Applying Deep Learning to the Newsvendor Problem”. *arXiv preprint arXiv:1607.02177.*

Pai, Ping-Feng, and Chih-Sheng Lin. 2005. “A Hybrid ARIMA and Support Vector Machines Model in Stock Price Forecasting”. *Omega* 33 (6): 497–505.

Papapanagiotou, V., R. Montemanni, and L. M. Gambardella. 2014. “Objective Function Evaluation Methods for the Orienteering Problem With Stochastic Travel and Service Times”. *Journal of Applied Operational Research* 6 (1): 16–29.

Park, Noseong, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. 2018. “Data Synthesis Based on Generative Adversarial Networks”.

Pavlyshenko, B. 2016. “Machine Learning, Linear and Bayesian Models for Logistic Regression in Failure Detection Problems”. *arXiv.*

Pierdzioch, Christian, Marian Risse, and Sebastian Rohloff. 2016. “A Boosting Approach to Forecasting the Volatility of Gold-Price Fluctuations Under Flexible Loss”. *Resources Policy* 47:95–107.

Popovič, Aleš, Tomaž Turk, and Jurij Jaklič. 2010. “Conceptual Model of Business Value of Business Intelligence Systems”. *Management: Journal of Contemporary Management Issues* 15 (1): 5–30.

Powers, David Martin. 2011. “Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation”. *Journal of Machine Learning Technologies.*

Provost, Foster, and Tom Fawcett. 2013. “Data Science and Its Relationship to Big Data and Data-Driven Decision Making.” *Big Data* 1 (1): 51–59.

Quinlan, J. Ross. 1986. “Induction of Decision Trees”. *Machine learning* 1 (1): 81–106.

Raheja, D., J. Llinas, R. Nagi, and C. Romanowski. 2006. “Data Fusion/-Data Mining-Based Architecture for Condition-Based Maintenance”. *International Journal of Production Research* 44 (14): 2869–2887.

Rappel, Niklas, Nikolai Stein, and Christoph M Flath. 2016. “Dynamic Intrusion Detection in Database Systems: A Machine-Learning Approach”. In *Proceedings of the 37th International Conference on Information Systems (ICIS).*

Rasmussen, Carl Edward. 2006. *Gaussian Processes for Machine Learning.* MIT Press.

Reddy, Aala Santhosh. 2016. “Why IoT Analytics Are a Manufacturer’s Most Important Tool”. *Harvard Business Review.*

Reif, Rupert, and Willibald A Günthner. 2009. “Pick-By-Vision: Augmented Reality Supported Order Picking”. *The Visual Computer* 25 (5-7): 461–467.

Reutterer, Thomas, Kurt Hornik, Nicolas March, and Kathrin Gruber. 2016. “A Data Mining Framework for Targeted Category Promotions”. *Journal of Business Economics*: 1–22.

Rotella, Perry. 2012. “Is Data the New Oil?” *Forbes.*

Russom, Philip. 2011. *TDWI Best Practices Report: Big Data Analytics.* Tech. rep. TDWI.

Rzepakowski, Piotr, and Szymon Jaroszewicz. 2012. “Decision Trees for Uplift Modeling With Single and Multiple Treatments”. *Knowledge and Information Systems* 32 (2).

Sanders, Nada R., and Ram Ganeshan. 2015. “Special Issue of Production and Operations Management on “Big Data in Supply Chain Management””. *Production and Operations Management* 24 (3): 519–520.

Saon, George, and Michael Picheny. 2017. “Recent Advances in Conversational Speech Recognition Using Convolutional and Recurrent Neural Networks”. *IBM Journal of Research and Development* 61 (4/5): 1–1.

Schwegmann, Bernd, Martin Matzner, and Christian Janiesch. 2013. “A Method and Tool for Predictive Event-Driven Process Analytics”. In *Proceedings of the 11th International Conference on Wirtschaftsinformatik (WI 2019).*

Schwerdtfeger, Björn, Rupert Reif, Willibald A Günthner, and Gudrun Klinker. 2011. “Pick-By-Vision: There Is Something to Pick at the End of the Augmented Tunnel”. *Virtual reality* 15 (2-3): 213–223.

Segebarth, Dennis, Matthias Griebel, Alexander Duerr, Cora R von Collenberg, Corinna Martin, Dominik Fiedler, Lucas B Comeras, Anupam Sah, Nikolai Stein, Rohini Gupta, et al. 2018. “DeepFLaSh, a deep learning pipeline for segmentation of fluorescent labels in microscopy images”. *bioRxiv*: 473199.

Semeniuta, Stanislau, Aliaksei Severyn, and Erhardt Barth. 2017. “A Hybrid Convolutional Variational Autoencoder for Text Generation”. *arXiv preprint arXiv:1702.02390.*

Sharma, Rajeev, Sunil Mithas, and Atreyi Kankanhalli. 2014. “Transforming Decision-Making Processes: A Research Agenda for Understanding the Impact of Business Analytics on Organisations”. *European Journal of Information Systems* 23 (4): 433–441.

Shen, Haipeng, and Jianhua Z Huang. 2008. “Interday Forecasting and Intraday Updating of Call Center Arrivals”. *Manufacturing & Service Operations Management* 10 (3): 391–410.

Shmueli, Galit, et al. 2010. “To Explain or to Predict?” *Statistical Science* 25 (3): 289–310.

Shmueli, Galit, and Otto R Koppius. 2011. “Predictive Analytics in Information Systems Research”. *MIS Quarterly* 35 (3).

Silver, David, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. “Mastering the Game of Go With Deep Neural Networks and Tree Search”. *Nature* 529 (7587): 484.

Sipos, Ruben, Dmitriy Fradkin, Fabian Moerchen, and Zhuang Wang. 2014. “Log-Based Predictive Maintenance”. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Smola, Alex J, and Bernhard Schölkopf. 2004. “A Tutorial on Support Vector Regression”. *Statistics and computing* 14 (3): 199–222.

Song, Chen, Xiaohong Guan, Qianchuan Zhao, and Yu-Chi Ho. 2005. “Machine Learning Approach for Determining Feasible Plans of a Remanufacturing System”. *IEEE Transactions on Automation Science and Engineering* 2 (3): 262–275.

Souza, Gilvan C. 2014. “Supply Chain Analytics”. *Business Horizons* 57 (5): 595–605.

Stefani, Karolin, and Patrick Zschech. 2018. “Constituent Elements for Prescriptive Analytics Systems.” In *Proceedings of the 26th European Conference on Information Systems (ECIS).*

Stein, Nikolai, and Christoph M Flath. 2017. “Applying Data Science for Shop-Floor Performance Prediction”. In *Proceedings of the 25th European Conference on Information Systems (ECIS).*

— . 2016. “Decision Support for Group-Based Electricity Prices in Smart Grids”. In *Proceedings of the 22nd Americas Conference on Information Systems (AMCIS).*

Stein, Nikolai, Christoph M Flath, and Carsten Boehm. 2018. “Predictive Analytics for Application Management Services”. In *Proceedings of the 26th European Conference on Information Systems (ECIS).*

Stein, Nikolai, Jan Meller, and Christoph M Flath. 2018. “Big Data on the Shop-Floor: Sensor-Based Decision-Support for Manual Processes”. *Journal of Business Economics*: 1–24.

Stein, Nikolai, Jan Meller, Christoph M Flath, and Richard Pibernik. 2019. “Data-Driven Sales Forcescheduling”. Working Paper.

Stein, Nikolai, Benedikt Walter, and Christoph M Flath. 2019. “Towards Open Production: Designing a Marketplace for 3d-Printing Capacities”. In *Proceedings of the 27th International Conference on Information Systems (ICIS).*

Stevenson, William J, Mehran Hojati, and James Cao. 2007. *Operations Management.* Vol. 8. McGraw-Hill/Irwin Boston.

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. “Rethinking the Inception Architecture for Computer Vision”. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*

Taieb, Souhaib Ben, and Rob J Hyndman. 2014. “A Gradient Boosting Approach to the Kaggle Load Forecasting Competition”. *International journal of forecasting* 30 (2): 382–394.

Taigel, Fabian, and Jan Meller. 2018. “Data-Driven Inventory Management: Integrated Estimation and Optimization”. *Available at SSRN 3256643.*

Tan, Baris, et al. 1998. “Effects of Variability on the Due-Time Performance of a Continuous Materials Flow Production System in Series”. *International Journal of production economics* 54 (1): 87–100.

Tay, Francis EH, and Lijuan Cao. 2001. “Application of Support Vector Machines in Financial Time Series Forecasting”. *Omega* 29 (4): 309–317.

Taylor, James W. 2008. “A Comparison of Univariate Time Series Methods for Forecasting Intraday Arrivals at a Call Center”. *Management Science* 54 (2): 253–265.

— . 2010. “Triple Seasonal Methods for Short-Term Electricity Demand Forecasting”. *European Journal of Operational Research* 204 (1): 139–152.

Tobler, Waldo R. 1970. “A Computer Movie Simulating Urban Growth in the Detroit Region”. *Economic geography* 46 (sup1): 234–240.

Tofallis, Chris. 2015. “A Better Measure of Relative Prediction Accuracy for Model Selection and Model Estimation”. *Journal of the Operational Research Society* 66 (8): 1352–1362.

Trummel, KE, and JR Weisinger. 1986. “Technical Note — The Complexity of the Optimal Searcher Path Problem”. *Operations Research* 34 (2): 324–327.

Tseng, T.-L., C.-C. Huang, F. Jiang, and J. C. Ho. 2006. “Applying a Hybrid Data-Mining Approach to Prediction Problems: A Case of Preferred Suppliers Prediction”. *International Journal of Production Research* 44 (14): 2935–2954.

Tukey, John W. 1977. *Exploratory Data Analysis.* Pearson.

Tulabandhula, Theja, and Cynthia Rudin. 2014. “On Combining Machine Learning With Decision Making”. *Machine Learning* 97 (1-2): 33–64.

Uspensky, James Victor. 1937. “Introduction to Mathematical Probability”.

van de Geer, Ruben, Qingchen Wang, and Sandjai Bhulai. 2018. “Data-Driven Consumer Debt Collection via Machine Learning and Approximate Dynamic Programming”. *SSRN Electronic Journal.*

Van Der Aalst, Wil, Arya Adriansyah, Ana Karla Alves De Medeiros, Franco Arcieri, Thomas Baier, Tobias Blickle, Jagadeesh Chandra Bose, Peter van den Brand, Ronald Brandtjen, Joos Buijs, et al. 2011. "Process Mining Manifesto". In *Proceedings of the International Conference on Business Process Management.*

Vansteenwegen, Pieter, Wouter Souffriau, and Dirk Van Oudheusden. 2011. "The Orienteering Problem: A Survey". *European Journal of Operational Research* 209 (1): 1–10.

Vapnik, Vladimir. 1996. *The Nature of Statistical Learning Theory.* Springer.

Varshney, Kush R, and Homa Alemzadeh. 2017. "On the Safety of Machine Learning: Cyber-Physical Systems, decision sciences, and data products". *Big data* 5 (3): 246–255.

Vinyals, Oriol, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, et al. 2017. "Starcraft Ii: A New Challenge for Reinforcement Learning". *arXiv preprint arXiv:1708.04782.*

Wager, Stefan, and Susan Athey. 2018. "Estimation and Inference of Heterogeneous Treatment Effects Using Random Forests". *Journal of the American Statistical Association*: 1–15.

Waller, Matthew A, and Stanley E Fawcett. 2013. "Data Science, predictive analytics, and big data: a revolution that will transform supply chain design and management". *Journal of Business Logistics* 34 (2): 77–84.

Wang, Jinjiang, Yulin Ma, Laibin Zhang, Robert X Gao, and Dazhong Wu. 2018. "Deep Learning for Smart Manufacturing: Methods and Applications". *Journal of Manufacturing Systems* 48:144–156.

Wang, K-J, JC Chen, and Y-S Lin. 2005. "A Hybrid Knowledge Discovery Model Using Decision Tree and Neural Network for Selecting Dispatching Rules of a Semiconductor Final Testing Factory". *Production planning & control* 16 (7): 665–680.

Wang, Qingchen. 2019. "Machine learning applications in operations management and digital marketing". PhD thesis, Amsterdam Business School Research Institute.

Ward Jr, Joe H. 1963. “Hierarchical Grouping to Optimize an Objective Function”. *Journal of the American statistical association* 58 (301): 236–244.

Wattenberg, Martin, Fernanda Viégas, and Ian Johnson. 2016. “How to Use T-Sne Effectively”. *Distill.*

Wedel, Michel, and PK Kannan. 2016. “Marketing Analytics for Data-Rich Environments”. *Journal of Marketing* 80 (6): 97–121.

Williams, Christopher, John Summerscales, and Stephen Grove. 1996. “Resin Infusion Under Flexible Tooling (RIFT): A Review”. *Composites Part A: Applied Science and Manufacturing* 27 (7): 517–524.

Wu, Desheng. 2009. “Supplier Selection: A Hybrid Model Using DEA, decision tree and neural network”. *Expert Systems with Applications* 36 (5): 9105–9112.

Wu, Shaomin, and Artur Akbarov. 2011. “Support Vector Regression for Warranty Claim Forecasting”. *European Journal of Operational Research* 213 (1): 196–204.

Wuest, Thorsten, Daniel Weimer, Christopher Irgens, and Klaus-Dieter Thoben. 2016. “Machine Learning in Manufacturing: Advantages, challenges, and applications”. *Production & Manufacturing Research* 4 (1): 23–45.

Xu, Lei, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. “Modeling Tabular Data Using Conditional GAN”. *arXiv preprint arXiv:1907.00503.*

Xu, Lei, and Kalyan Veeramachaneni. 2018. “Synthesizing Tabular Data Using Generative Adversarial Networks”. *arXiv preprint arXiv:1811.11264.*

Yan, Xiaowei, Chengqi Zhang, and Shichao Zhang. 2003. “Toward Databases Mining: Pre-Processing Collected Data”. *Applied Artificial Intelligence* 17 (5-6): 545–561.

Yu, Lantao, Weinan Zhang, Jun Wang, and Yong Yu. 2017. “Seqgan: Sequence Generative Adversarial Nets With Policy Gradient”. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence.*

Yu, Lean, Shouyang Wang, and Kin Keung Lai. 2006. "An Integrated Data Preparation Scheme for Neural Network Data Analysis". *IEEE Transactions on Knowledge and Data Engineering* 18 (2): 217–230.

Zhang, G Peter. 2003. "Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model". *Neurocomputing* 50:159–175.

Zhang, Shichao, Chengqi Zhang, and Qiang Yang. 2003. "Data Preparation for Data Mining". *Applied Artificial Intelligence* 17 (5-6): 375–381.

Zhang, Shu, Jeffrey W Ohlmann, and Barrett W Thomas. 2014. "A Priori Orienteering With Time Windows and Stochastic Wait Times at Customers". *European Journal of Operational Research* 239 (1): 70–79.

Zhao, Yan, Xiao Fang, and David Simchi-Levi. 2017. "Uplift modeling with multiple treatments and general response types". In *Proceedings of the 2017 SIAM International Conference on Data Mining*, 588–596. Philadelphia, PA: Society for Industrial / Applied Mathematics.

Appendix

A List of Publications

Flath, Christoph M, and Nikolai Stein. 2018. “Towards a Data Science Toolbox for Industrial Analytics Applications”. *Computers in Industry* 94:16–25.

Griebel, Matthias, Alexander Dürr, and Nikolai Stein. 2019. “Applied Image Recognition: Guidelines for Using Deep Learning Models in Practice”. In *Proceedings of the 14th International Conference on Wirtschaftsinformatik (WI).*

Krenzer, Adrian, Nikolai Stein, Matthias Griebel, and Christoph M Flath. 2019. “Augmented Intelligence for Quality Control of Manual Assembly Processes Using Industrial Wearable Systems”. In *Proceedings of the 27th International Conference on Information Systems (ICIS).*

Rappel, Niklas, Nikolai Stein, and Christoph M Flath. 2016. “Dynamic Intrusion Detection in Database Systems: A Machine-Learning Approach”. In *Proceedings of the 37th International Conference on Information Systems (ICIS).*

Segebarth, Dennis, Matthias Griebel, Alexander Duerr, Cora R von Collenberg, Corinna Martin, Dominik Fiedler, Lucas B Comeras, Anupam Sah, Nikolai Stein, Rohini Gupta, et al. 2018. “DeepFLaSh, a deep learning pipeline for segmentation of fluorescent labels in microscopy images”. *bioRxiv*: 473199.

Stein, Nikolai, and Christoph M Flath. 2017. “Applying Data Science for Shop-Floor Performance Prediction”. In *Proceedings of the 25th European Conference on Information Systems (ECIS).*

— . 2016. “Decision Support for Group-Based Electricity Prices in Smart Grids”. In *Proceedings of the 22nd Americas Conference on Information Systems (AMCIS).*

Stein, Nikolai, Christoph M Flath, and Carsten Boehm. 2018. "Predictive Analytics for Application Management Services". In *Proceedings of the 26th European Conference on Information Systems (ECIS).*

Stein, Nikolai, Jan Meller, and Christoph M Flath. 2018. "Big Data on the Shop-Floor: Sensor-Based Decision-Support for Manual Processes". *Journal of Business Economics*: 1–24.

Stein, Nikolai, Jan Meller, Christoph M Flath, and Richard Pibernik. 2019. "Data-Driven Sales Forcescheduling". Working Paper.

Stein, Nikolai, Benedikt Walter, and Christoph M Flath. 2019. "Towards Open Production: Designing a Marketplace for 3d-Printing Capacities". In *Proceedings of the 27th International Conference on Information Systems (ICIS).*

B Numerical Evaluation

In Chapter 5, we estimate success probabilities of winning a project to calculate the uplift predictions and determine optimized visiting policies. To this end, we leverage several binary probabilistic classification models to estimate the true success probabilities and the expected uplifts. However, the true success probabilities prior to the final decision cannot be observed to evaluate the models. To overcome this issue, we use the ϕ coefficient to evaluate the performance of the models based on the binary realizations while accounting for the observed class imbalances.

Even though the ϕ coefficient measures the correlation between two binary variables, we use it as a proxy for the Pearson correlation ρ between the predicted uplifts $\Delta\hat{p}$ and the true (but unknown) uplifts Δp to find a suitable value for the weighting parameter λ.

In this appendix, we analyze the relationship between the ϕ coefficient and the Pearson correlation ρ through a brief numeric study. To this end, we use a controlled environment in which success probabilities for the binary outcomes are known at all times.

In particular, we simulate rolls of n fair dice. Each die has s sides with values ranging from 1 to s. Let x_i describe the outcome of die i and $y_m = \sum_{i=1}^{m-1} x_i$ the sum of all rolls prior to die m. To create a binary target variable z, a game is considered "won" ($z = 1$) if the sum of the dice rolls is greater or equal than a given target t and "lost" ($z = 0$) otherwise:

$$z = \begin{cases} 1 & \text{if } y_{n+1} \geqslant t \\ 0 & \text{otherwise} \end{cases} \tag{B.1}$$

Before each roll m, the success probability p is defined as

$$p = 1 - P\left(y_{n+1} - y_m \geqslant t\right). \tag{B.2}$$

Following Uspensky (1937), the probability of rolling a sum of q points with n s sided dice is

$$P\left(\sum_{i=1}^{n} x_i = q\right) = \frac{1}{s^n} \sum_{k=0}^{\left\lfloor \frac{q-n}{s} \right\rfloor} (-1)^k \binom{n}{k} \binom{q-sk-1}{n-1}. \tag{B.3}$$

Hence, the best possible prediction for the success probability prior to roll m is

$$\hat{p}_m = 1 - \sum_{q=t-y_m}^{(n-m+1)s} P\left(\sum_{i=1}^{n} x_i = q\right). \tag{B.4}$$

The best possible prediction for the binary target variable z prior to roll m is

$$\hat{z}_m = \begin{cases} 1 & \text{if } \hat{p}_m \geqslant 0.5 \\ 0 & \text{otherwise} \end{cases}. \tag{B.5}$$

We compare ρ and ϕ by simulating r rounds of the game described above. To this end, we want to account for different model qualities as well as for the fact that we cannot observe the true underlying success probabilities in the setting described in Chapter 5. We account for the model quality by defining predictors of different qualities based on the number of "known" realizations prior to roll i ($\hat{p}_i$ and $\hat{z}_i$). Additionally, we account for the unknown true success probabilities by defining "true" values for $\hat{p}_j$ and $\hat{z}_j$ for all combinations of $j \geqslant i$.

Figure B.1 compares $\rho(\hat{p}_i, \hat{p}_j)$ and $\phi(\hat{z}_i, \hat{z}_j)$ for varying model qualities ($i \in 1, \ldots, 11$) and "true" values ($j \in 1, \ldots, 11$). Here, the facets show different quality levels of the estimations for the "true" values while the dots visualize the quality of the different prediction models. Additionally, we report the r^2 of a linear regression between ϕ and ρ without an intercept. We see that the ϕ coefficient measuring the correlation between the binary predictions and the binary outcome is a very good proxy for the correlation ρ between the predicted and the true success probabilities in the evaluated setting.

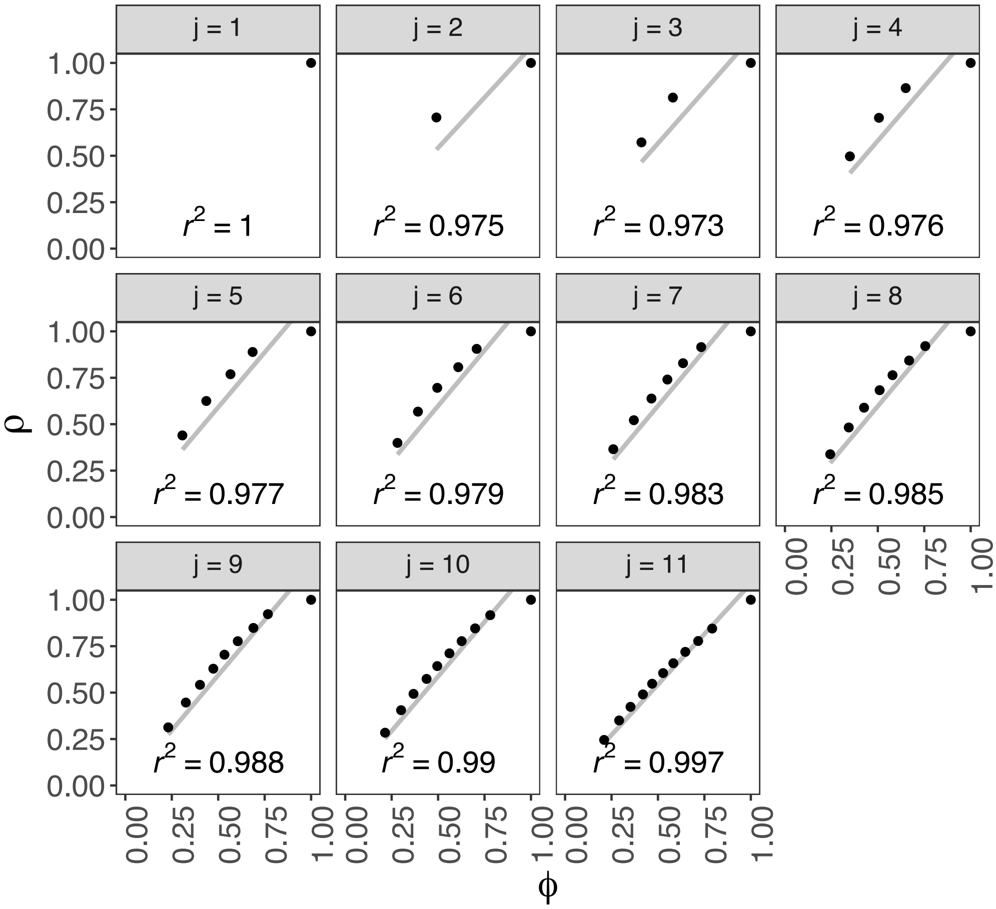


Figure B.1: Comparison of ϕ coefficient and ρ for different model quality levels.

While, this comparison between the ϕ coefficient and ρ yields promising first results for a linear relationship between the dependent and independent variables. Future work should analyze the relationship for more complex settings.

Eidesstattliche Erklärung

Hiermit erkläre ich gemäß § 6 Abs. 2 Nr. 2 der Promotionsordnung der wirtschaftswissenschaftlichen Fakultät der Universität Würzburg, dass ich diese Dissertation eigenständig, d.h. insbesondere selbständig und ohne Hilfe eines kommerziellen Promotionsberaters angefertigt habe. Ausgenommen davon sind jene Abschnitte, bei deren Erstellung ein Koautor mitgewirkt hat. Diese Abschnitte sind entsprechend gekennzeichnet und die Namen der Koautoren sind vollständig und wahrheitsgemäß aufgeführt. Bei der Erstellung der Abschnitte, bei denen ein Koautor mitgewirkt hat, habe ich einen signifikanten Beitrag geleistet, der meine eigene Koautorschaft rechtfertigt.

Außerdem erkläre ich, dass ich außer den im Schrifttumsverzeichnis angegebenen Hilfsmitteln keine weiteren benutzt habe und alle Stellen, die aus dem Schrifttum ganz oder annähernd entnommen sind, als solche kenntlich gemacht und einzeln nach ihrer Herkunft nachgewiesen habe.

Würzburg, den 11. Dezember 2019

Nikolai Werner Stein