MASTER

Forecasting the demand for resources through a predictive modeling approach

Eikelenboom, D.

Award date:
2016

Link to publication
Forecasting the Demand for Resources through a Predictive Modeling Approach

by

D. Eikelenboom

...
Note: Information and drawings embodied in this work are considered strictly confidential and are supplied on the understanding that they will be held confidentially and not disclosed to third parties without the prior written consent of the author and Microsoft B.V.

A public and electronic version of this thesis is available at http://repository.tue.nl/.
Executive Summary

This dissertation is an integral part of the graduation track within the European double degree master program of the European Institute of Technology called ‘EIT Digital Master School’. Besides a major in the field of Service Design and Engineering, formed as a combination of the degrees in Computer Science and Business Information Systems from respectively the University of Trento and the Technical University of Eindhoven, a large minor in Innovation and Entrepreneurship characterizes the program. The work described in this document, is written based on a project and internship performed at Microsoft, Schiphol, The Netherlands.

In this work, we propose a machine learning framework to forecast the demand for resources in the environment of consultancy organizations, and in specific for Microsoft Consultancy Services. Specific characteristics for such an organization are the presence of a sales pipeline, a work backlog, a pool of resources, and a planning and a delivery phase.

Through domain research we found the three dimensions of expected workload, resource mix and timing as important aspects contributing to the resource need. Accordingly, we proposed a decomposition of the modeling approach into three dimensions, thereby reducing the complexity of the modeling effort, and introducing multiple predictive models to predict demand for each dimension. Among these predictive models are models for sales pipeline classification, workload estimation (leveraging on earlier work by the ESBI \(^1\) team), early resource mix determination, project length and identification of project start date slipping. Together, a view on future work can be constructed on a 6/12-months horizon.

The extent to which the demand for resources is predictable is defined by the predictive power of each of the individual models and problems. Sales win determination showed to be a problem that can be predicted through a classification approach. Workload in hours is found to be generally well-predictable, where the workload is mainly determined by the value of a project. Varying project value estimations during the services life cycle however, make the project value subjective to estimations of stakeholders, which are shown to be lowly correlated with the eventual workload. Generally, the further a project is situated in the services life-cycle, the better the accuracy of the predictions. A clustering approach has been used to identify a taxonomy of projects out of a-priori unknown groups of similar project resource mixes. For each cluster found, staffing templates could subsequently be derived that are characteristic for projects in a certain cluster. A classification approach showed that the found taxonomy of resource mix distributions is strongly related to service offerings and is therefore already identifiable in an early sales stage. The predictability of project duration is found to be reasonable and influenced by specific service-offerings, project value, planning and the availability of resources. Models are found to outperform the human forecast on the project length. Moreover, through a classification approach, the slipping of a project’s start date is found to be predictable with good precision.

Through interactive dashboards, insights are provided into the workforce planning process, providing views on capacity planning, resource planning and past delivery. Collaboration with leading resource and capacity planners led to the understanding of which data is important, and gave direction to dashboard design to maximize the value delivered to end-users. At this moment, a prototype is available for the Western European-area and we are looking for upscaling to EMEA \(^2\).

\(^1\) Enterprise Services Business Intelligence
\(^2\) Europe, the Middle East and Africa
Acknowledgments

These are exciting times: data science initiatives are rising and tend to initiate a new industrial revolution that is powered by machine learning and insights achieved through data analysis. I am very happy that I have got the chance to perform my thesis work at a company that is working at the forefront of this revolution and have been able to contribute my little part to it.

Especially I wish to thank my supervisor during this period, Mykola Pechenizkiy, for the provided feedback at least once a week, and his swift response times to emails even in the middle of the night. At least as much I want to thank Ingeborg Struijk, Wim van Ginkel and Julio Peironcely for their feedback, sharing of experiences and knowledge, and for their support on a regular basis. I want to thank Anna Wilbik for supervising my minor thesis besides this work, Rui Jorge Almeida and Yannis Velegrakis for taking place in the assessment committee for this thesis, and my family for their support along the way.

Furthermore, I want to thank Judith Cuppen, Services Lead in the Netherlands, and Mariusz Matuszewski, Global Capacity Management Lead Data Insights, as important customers in this project, for their attention and feedback for the direction of this project. As well, I want to thank the ESBI team in Redmond for sharing their knowledge and earlier results, and further anyone else who has contributed to this project in any form but has not been mentioned above.

Sincere thanks may also go to the European Institute of Technology, that has provided me with the chance and the resources to study and obtain a degree at two leading universities in Europe, and meet people and cultures from all over the world. This experience has been truly unforgettable.

Finally, I want to thank Boudewijn van der Zwan and Microsoft for offering me a position in the EMEA Data Insights team. I can’t wait to get started with this new opportunity!

Dennis.
## Contents

**Executive Summary** iii  
**Acknowledgments** v  

### 1 Introduction
1.1 Workforce Planning 1  
1.1.1 Research Objectives 2  
1.1.2 Research Questions from the Data Science Perspective 2  
1.2 Research Problem from the Business Perspective 3  
1.2.1 Business Objectives 3  
1.2.2 Research Questions from the Business Perspective 4  
1.3 Scope of Research 4  
1.4 Outline of the Thesis 5  

### 2 Background
2.1 Workforce Planning 7  
2.1.1 Definitions 8  
2.1.2 Aspects of Workforce Planning 8  
2.1.3 Managing the Skill Pool 8  
2.1.4 Limitations 9  
2.2 Customer Engagement Life-cycle 9  
2.2.1 Sales Process 9  
2.2.2 Delivery Process 10  
2.2.3 Difficulties 10  
2.3 Data Science Perspective on Workforce Planning 10  
2.3.1 Data Wrangling 11  
2.3.2 Predictive Modeling 11  
2.4 Related Work 12  
2.4.1 Related Work in Academia 12  
2.4.2 Related Work at Microsoft 13  

### 3 Methodology
3.1 Research Overview 15  
3.2 Research Design 15  
3.2.1 The Data Science Work Flow as a Methodology for Predictive Modeling 16  
3.3 Data Collection 17  
3.4 Evaluation Framework 18  
3.4.1 Criteria of Success 18  
3.4.2 Evaluation Metrics 18  

### 4 Domain Analysis 21  

### 5 Data Preparation
5.1 Systems and Available Data Sources 23  
5.2 Constructing a Bird’s Eye View on the Process 24  
5.3 Process Progress Dependent Views 24  
5.4 Feature Engineering 25  
5.5 Feature Selection 25  
5.6 Issues and Limitations 25  
5.6.1 Identifiers and Loosely Coupled Systems 26  
5.6.2 Granularity of data 26  
5.6.3 Low Data Quality 26
5.6.4 Changing Environment and Systems .......................... 27
5.6.5 Process Hacking .............................................. 27

6 A Machine Learning Framework to Forecast the Resource Need 29
6.1 Introduction ....................................................... 29
6.2 Objectives ......................................................... 29
6.3 Problem Decomposition ......................................... 30
6.4 Sub-models and Definitions .................................... 31
6.5 Modeling Risks .................................................... 32
   6.5.1 Data Quality ................................................. 32
   6.5.2 Propagation of Errors ...................................... 33
6.6 Evaluation ......................................................... 33

7 Sales Win/Loss Classification 35
7.1 Problem Formulation ............................................. 35
7.2 Discussion of Earlier Work at Microsoft ......................... 36
   7.2.1 Feature Importance ......................................... 36
   7.2.2 Model Performance ......................................... 37
7.3 Evaluation and Integration of Work .............................. 37

8 Workload Prediction 39
8.1 Problem Formulation ............................................. 39
8.2 Feature Selection and Motivation ............................... 40
8.3 Model Training ................................................... 41
8.4 Performance Evaluation ....................................... 42
8.5 Findings .......................................................... 43

9 Predicting Project Resource Mix 47
9.1 Problem Formulation ............................................. 47
9.2 A Clustering Approach to Project Taxonomy Identification .. 48
9.3 Classifying Early Available Features to Taxonomies .......... 49

10 Project Timing Models 55
10.1 Definitions ....................................................... 55
10.2 Seasonal Influence .............................................. 56
   10.2.1 Observations ................................................ 56
   10.2.2 Time Series Analysis ..................................... 56
   10.2.3 Time Series Forecast ...................................... 57
10.3 Start Date Slipping ............................................. 58
   10.3.1 Problem Formulation ...................................... 58
   10.3.2 Feature Selection and Engineering ....................... 59
   10.3.3 A Regression Approach .................................. 59
   10.3.4 A Binary Classification Approach ....................... 60
   10.3.5 A Multi-Class Classification Approach ............... 61
   10.3.6 Modeling methodology .................................. 62
   10.3.7 Evaluation ................................................... 62
10.4 Estimating Actual Project Length ............................. 62
   10.4.1 Problem Formulation ...................................... 62
   10.4.2 Feature Selection and Considerations ................. 63
   10.4.3 Modeling Methodology .................................. 64
   10.4.4 Evaluation ................................................. 64

11 Aggregated Forecasts 67
11.1 A Periodical View on Future Resource Demand ............. 67
   11.1.1 Confidence Intervals ..................................... 68
   11.1.2 Determination of Role-specific Workload ............. 68
11.2 A Three-Dimensional Aggregated Forecast .................. 68
11.3 Forecast Limitations .......................................... 69

12 Interactive Visual Analytics 71
13 Conclusion 73
14 Recommendations for Future Work 77
  14.1 Short Term ......................................................... 77
  14.2 Mid Term .......................................................... 78
  14.3 Long Term ......................................................... 78
Bibliography 81
Introduction

Demand forecasting is the art of predicting the future demand for a certain product or service. This means determining the probability of demand for a specific product or service in future time periods, given activities and events that occurred in the past [10]. Demand is of large importance in the management of a business. It helps an organization to reduce risks involved in business activities, supports business decisions, and provides direction into organization’s future investments and expansion directions [2]. Examples of common forecasting problems are predicting future sales [40], estimating the amount of work required to deliver a certain service [35], or forecasting if a customer will return to buy a product [34]. Also the prediction of future energy demand based on historical consumption, trends and weather data, is an example of demand forecasting [51].

In this work we focus on predicting workforce demand, which can be regarded as a product of sales in a consulting organization, where we face this problem through a predictive modeling approach. We research the feasibility to give insights into, and predict the demand for resources in the context of a consultancy organization. In specific, we propose a machine learning framework to forecast the demand for resources for such an organization based on an analysis of sales opportunities and projects yielding from sales. Moreover, we decompose the resource demand forecasting problem into a number of dimensions, and leverage predictive analytics with interactive visualizations to give insights into future demand. The resulting insights are supportive to the capacity planning, resource planning and sales processes.

This introduction is structured as follows. First, section 1.1 provides an introduction to the research problem of workforce planning as the main subject in this work. Then, section 1.2 provides more context by describing the workforce planning problem from a business perspective, involving a case study at a Fortune 500 organization. In section 1.1 we state the main research questions as researched in this work. Lastly section 1.4 gives an outline of the structure of the remainder of this work.

1.1. Workforce Planning

Capacity management and resource planning are crucial topics in every organization that has to manage a group of professionals, together also referred to as workforce planning. It consists of the planning of the workforce size and the allocation of resources to fulfill customer demands. Demands that often require various mixes of skills from people, and that is fluctuating over time, which makes resource planning not a trivial task. Capacity management makes sure that there are not too many, neither too less, people employed in an organization. Forecasting fluctuations in the demand for resources is crucial, but complex, as the demand is dependent on many different internal and external factors that influence the quantity of work and specialisms in skills that will be required.

Scarcity of resources and the laboriousness of hiring are complicating factors in the capacity management process. Limitations on the resources’ availability make a well-forecasted capacity need desirable to pursue an effective capacity management strategy. If peaks in resource demand can be foreseen, time to market can
be decreased and people can be hired in time so that the additional demand can be handled by the organization. Early insights in the demand for resources could support business decisions and simplify operations, as the workforce size and the resource mix can be adapted upon.

The current state-of-the-art in demand forecasting mainly focuses on predicting results from sales by analyzing historical sales results of products and fluctuation of trends [12, 27, 40, 49, 50]. Some work has been performed regarding workforce demand forecasting, but focuses on a high economical level [55], or on other sectors [23, 31]. Developments in the upcoming areas of data analytics and machine learning recently, have enabled us to leverage an increasing amount of data to discover insights in numerous fields of study []. These developments open up new possibilities to also face capacity management challenges through a data-driven and predictive modeling approach. Earlier work regarding workforce planning generally does not yet broadly adopt machine learning as a methodology [53]. IBM developed a scheduling tool called SWOPS [25] (Shift Work Optimized Planning and Scheduling) that addresses complex scheduling issues and that uses statistical techniques to create its forecast. SWOPS is capable of forecasting the workload, computing the resource requirements and building a work schedule. The tool however focuses on resource scheduling rather than capacity planning and lacks the bird-eye view on upcoming demand.

1.1.1. Research Objectives

The primary objective of this research is to provide a proof-of-concept that the demand for resources can be predicted through a machine learning approach. More in specific, we aim to provide a machine learning framework that can serve as the baseline solution for future predictive modeling work on the topic of workforce demand forecasting, and in the context of a service-delivering organization.

Through application of predictive modeling concepts and advanced analytical techniques, we research how the process of workforce planning can be made more transparent, insightful and predictable. Especially we want to put attention to the notion of risk in demand forecasts by finding a way to express prediction confidence while communicating model outcomes.

Moreover, we wish to research to what extend the demand for resources is predictable. Given the multiple problem dimensions, we aim to find out which parts of the problem are more predictable, and provide evidence of the problem’s predictability through a case study at a Fortune 500 company.

1.1.2. Research Questions from the Data Science Perspective

Given the research problem of workforce planning, and especially its aspects of capacity planning and resource planning, we define the following research question as the central problem as discussed in this work: ‘Can we propose a machine learning framework to predict the demand for resources in a consulting environment?’

More in detail, regarding the objectives from a research perspective as described in section 1.1.1, we consider the following research questions.

• What are the key dimensions in resource demand forecasting?
• How can these dimensions be combined to form a future view on the demand for resources?
• To what extend is the demand for resources predictable?
• How can the services life-cycle be made more insightful?
• How can data insights support the delivery process?
• How can data insights support the capacity management process?
• To what extend can we predict sales pipeline wins?
• To what extend can we predict a project’s workload?
• To what extend can we predict a project’s resource mix?
• How predictable is the length of the delivery phase of a project?
1.2. Research Problem from the Business Perspective

For our study, we have access to data from the consulting services department of Microsoft, which has made data available from over the last decade for the Western-European area. Consulting Services is the business unit engaged with delivering consultancy services to its enterprise and government clients, and has a flexible workforce of consultants, architects, project managers and engineers, also referred to as resources, that possess different skills and operate over multiple regions.

Because of a fluctuating demand in resource need, planning of resources need to be scheduled accordingly. Moreover, capacity management needs to make sure that there are not too many, neither too less, people employed. Forecasting these fluctuations in resource need is hard, as the process is dependent on many different factors that influence the quantity and type of projects that result in the work backlog. Looking at the past, we have knowledge about what the demand has been for historical projects and what kind of resources were required. For current work, resource planners base their planning mainly on their intuition and experience. However regarding future demand the knowledge is rather limited. At this point, the main questions for this research project are raised: can we construct a view on future work by systematically looking at what kind of work is coming from the sales pipeline, and can we estimate the expected work effort for consultancy projects with the knowledge about delivery that we gained from the past? In the following subsections 1.2.2, 1.2.1 and 1.3 we describe business questions, objectives and scope in more detail.

1.2.1. Business Objectives

At the first place this research is a feasibility study to analyze if the demand for resources can be predicted through a predictive modeling approach, and with the data available at Microsoft. A successful result, could potentially make a large impact on the way that capacity management is performed at Microsoft and at other service-delivering organizations.

From a company perspective, there is a strong need to get better insights in the resource need for the Western European region. Due to changes in the resource planning methodology, the resource need has now to be determined on a region-wide level, stretching multiple countries, where this has been country-specific in the past. Achieving region-wide insights into the resource need will contribute to make capacity management and resource planning more predictable. In the ideal case the forecast can serve as decision-supportive to the process, aiming for a forecasting horizon of six up to twelve months.

Moreover, achieving insights into the trajectory from sales to delivery for consultancy projects has never been done before. For this reason, the various systems do not have many commonalities and thus a first objective is to create a bird-eye view on all the systems that are involved to get a broad insight into the full services life-cycle. Moreover this broad view is important since in order to create predictive models on early stage projects, we need to have a proper view of completed projects in their early stages to serve as training data during the modeling phase.

Lastly, an important objective is the communication and visualization of model outcomes. An effective dashboard, displaying the forecasting outcomes, would allow the organization to derive insights and incorporate the forecasts as a decision-supportive tool. An important factor in the visualization is the representation of confidence intervals in the forecast.
1.2.2. Research Questions from the Business Perspective

The hypothesis is raised that the resource need can be predicted by using a machine learning approach. By leveraging data from the sales and resource planning process, and historical project delivery, the assumption is that a predictive model can be derived that gives a view on the resource need for the upcoming period. To assess these hypotheses, a number of research questions can be formulated.

- What characterizes the Services Life-cycle at Microsoft, who is involved and which systems are used?
- Can we reconstruct the process from initial sales lead, to the delivery, making a bird-eye view on the trajectory of a project?
- Can we achieve insights into the workforce demand for future time periods?
- Can we predict if sales opportunities will result in a sales win or loss?
- Can we predict the workload for a consultancy project given its characteristics?
- Can we estimate the project length based on a project its characteristics?
- Can we forecast delay in project delivery?
- Is the demand for resources influenced by trends and seasonality?
- Can we predict the resource mix for a specific type of project?
- How can a forecast on future work be visualized in such a way that differentiation can be made between the different levels of certainty?
- Can we outperform the human forecast on project length?
- Can we outperform the human forecast on delivered work?

1.3. Scope of Research

There are a number of reasons why a predictive modeling approach to workforce planning is only recently discussed in literature and at Microsoft. The reason can be seen as multi-fold and is driven by an evolution that is more and more seen as the fourth industrial revolution.

Half a century ago, information technology and electronics started the third industrial revolution, which led to automation of many production processes. Now, the start of a new revolution has announced itself, following on many developments around data: the rise of machine learning for the mass, developments in artificial intelligence and the out-roll of connected devices and robotics at scale seem to be the start of a fourth revolution. Building on the infrastructure of the third industrial revolution, through information technology and automated production lines, data can be collected and analysed at large scale. Large collections of data hide many information, and new technologies allow us to reveal these insights and leverage them into our processes. These technological developments have led to industrial changes at such a scale and velocity, that they can likely not be accounted under earlier industrial revolutions according to Klaus Schwab, the founder of the World Economic Forum [48].

To allow the new revolution to take place, massive amounts of data storage, analytical capabilities, and increased investments in related research are needed. This is why technology giants like Microsoft are currently investing heavily in cloud technology, data centers and research. Microsoft’s CEO Satya Nadella puts high bets on data and regularly mentions that data will be the new currency. This mind set and new direction of the company motivates initiatives such as this project and the large investments in similar initiatives and research.

For the scope of this project within Microsoft we focus on the consultancy branch of the services organization. Regarding the type of services that are delivered at Microsoft we can differentiate between two types. First there are Premier Support Services, which are in general annual contracts with a fixed number of support hours. Secondly, there are the Consultancy Services. Since both types are significantly different and the latter is less defined and less predictable we limit ourselves in this work to the Consultancy Services (MCS).
Moreover, the geographical scope of this work is limited to the Western European region. Chapter ?? later discusses the domain concepts and restrictions in more detail.

1.4. Outline of the Thesis

In this chapter we introduced the problem domain of this study, the research and business problems, and the scope, assumptions and limitations of this research project. The remainder of this work is structured as follows.

In the next chapter, background information is provided discussing the key concepts related to the problem domain, such as workforce planning and predictive modeling. As well related work from academia and existing work at Microsoft is discussed. Chapter 3 describes the methodology used in this study, which is particularly organized in an iterative way. Chapter ?? gives an elaborate analysis on the domain, describing involved processes and limitations to the current workforce planning processes. Then, chapter 5 goes into more technical detail by describing available data and systems, issues with the data and processes, and how data transformations have been applied to increase the data quality and to achieve a better fit to the domain of this study.

Chapter 6, 7, 8, 9, 10 and 11 form the core part of this work, describing the problem decomposition, machine learning framework, and the predictive modeling work. Chapter 6 proposes a decomposition of the model into three dimensions, thereby reducing the model complexity into a number of cohesive concepts. Discussed is how the separate concepts, can together form a machine learning framework to forecast the future demand for resources, through chaining multiple predictive models together. To forecast the future backlog of work, a first predictive model is introduced in chapter 7 which applies classification on sales opportunities to identify the opportunities that will eventually lead to a signed deal and which do not. Then, chapter 8 describes a predictive model to forecast the amount of work required to deliver a certain project. Chapter 9 describes an approach to find the resource mix, the roles, required to deliver a certain project. Chapter 10 covers the aspects of timing, discussing predictive models that predict delay in start date and estimate actual project length. Chapter 11 describes the combination of the individual predictive models into an aggregated forecast on the demand for resources.

Chapter ?? describes interactive analytics and dashboards, and how they allow for extensive and accessible risk analysis possibilities in contrast to traditional visualization techniques. Finally, in chapter 13 we conclude on the research outcomes and give recommendations for future work in chapter 14.
2

Background

In this chapter we discuss key concepts for the problem domain and discuss the current state-of-the-art in literature. First we discuss the profession of workforce planning and its processes. Then, we describe the so-called Customer Engagement Life-cycle, which covers the process from the initial sale to delivery of a project for consultancy organizations. Section 2.3 discusses the field of data science. In section 2.4 we discuss earlier work related to the problem domain as described in literature, as well as at Microsoft.

2.1. Workforce Planning

The planning of a workforce is one of the most complex challenges that businesses face. The problem includes the decisioning about how many employees should be hired or dismissed and when these employees must be assigned to projects. This makes the workforce planning problem a combination of a staffing and a scheduling problem, and moreover a problem that is increasing in complexity as the size of an organization grows [22]. Moreover, Capelli [16] describes workforce planning as the foundation to the talent management process. Labor demand forecasting is crucial for workforce planning processes. Having a good view on future resource needs are essential for driving profitability in a service-oriented business. At the first place you do not want a shortage of employees or a surplus of resources who are not fully deployed. Secondly, gaps in the resource pool are undesirable since they result in reduced performance, productivity and profitability [9].

Prediction of future demand in general is a research topic that is applied in numerous industries [12, 23, 27, 31, 40, 49, 50]. It is also a topic for which literature goes back till 1992 when Grinó [20] already described research towards neural network time series prediction on water demand. Literature in relation to capacity planning management mostly describes research on the topic of demand prediction for data centers, electricity consumption or networking traffic, but conceptually research shows a lot of overlap independent of the problem domain. Literature on demand prediction for data center applications [26], for example, describes that the accuracy of capacity planning predictions depends on the ability to recognize trends for expected changes in future demands, and to reflect those unexpected changes in future demand predictions. This insight can be applied for any domain that faces the capacity planning problem.

One of the distinguishing factors that makes the workforce planning process explicitly different than other scheduling problems is the fact that people are a very heterogeneous set of resources since each resource possesses different skills, capabilities and levels of experience that cannot be neglected.

In the following subsections we discuss the workforce planning problem in more detail referring to existing literature around the topic including influential factors, determinants of skills, the treatment of differences in skills between resources, and optimization methodologies. First however, we state more clear the differences between resource planning and capacity planning as part of the workforce planning problem.
2. Background

2.1. Definitions

Workforce planning is a two-fold problem that can be seen as a combination of a staffing problem and a scheduling problem. Capacity planning is the staffing problem that deals with the management decision of how many people to hire or layoff, with which skills and for which period. Moreover, it is a process that is stretching over relatively long periods, for which the result of decisions can often only be seen after a long time. Hence, hiring processes are time-consuming and it may take more than six months before a new resource can be on-boarded within the organization. From a capacity planning perspective, and for a period in time, it is desirable to know 1) the amount of work in hours that can be expected and 2) the roles and skills required from the resources.

As a counter-process, we can define the resource planning process on an operational level to deal with the scheduling of resources to work. Demand for work, performed in projects, and thus the demand for resources, is directly influenced by the outcome of the sales process. And thus by the demand from the market and the capabilities and objectives of the sales workforce. Resource planning is a process that covers relatively short time periods, and that has as the main goal to find the best fit between required skills for projects and the available workforce. If a project is not executed by resources with the appropriate skills, or when resources are underutilized, this will lead to a loss of profit to the business [43]. For each project it is desirable to identify 1) the workload in hours to deliver the project 2) the amount of work for each role type that will be required for each role over time. 3) the moment of project delivery in time.

2.1.2. Aspects of Workforce Planning

In order to create predictive analytics as support for the workforce planning it is important to understand the potential influencing factors that may impact the capacity and resource planning. Since resources are very heterogeneous in terms of skills, we must find a way on how to deal with these differences. In this section we describe the various methodologies described in former work.

By defining classes and types for skills, differences between resource skills can be better analyzed. In earlier work [11, 13, 22], skills get categorized in terms of hierarchy classes and categorical classes. When skills are distinguished hierarchically, then resources with a higher skill can also perform the work for resources with a lower skill, which make their work substitutional. For example a senior consultant, may execute the work of a junior consultant, but not the other way around. A consultancy role hierarchy level can for example be described by Principal Consultant ⊆ Senior Consultant ⊆ Junior Consultant ⊆ Consultant. On the other hand, when skills are organized into categories (e.g. technical domains), people with different skills can do different tasks, but only limited to their core tasks. One can overcome this limitation of the categorical model, by training people for multiple tasks, also known as cross-training [22].

In more detail, skills of a resource can be determined by taking into account multiple aspects. Earlier work [14, 19, 29, 41] describes determinants of age, seniority levels, capability, experience, qualifications, job titles, grades and licenses, but also personality types or gender might be taken into account. When categorizing these determinants into either categorical or hierarchical classes, seniority is a typical example for a hierarchical class, whether qualification is an example for a categorical class.

2.1.3. Managing the Skill Pool

Because of a changing demand, changes in the skill pool are regularly needed, requiring the composition of the skill pool to change. In earlier work, multiple ways are described to adapt the skill pool. In general, two approaches to manage the skill pool can be observed, either through staffing measures or through training, which we both discuss shortly.

A popular way to adapt the skill pool, is the hiring or dismissal of temporal workers. This is a regularly used instrument as it provides a large flexibility in the size of the workforce. As well it is an ideal way to face fluctuating demands because of seasonality. If changes are needed for the longer term, the organization can also decide to hire or dismiss full time employees. Due to regulations however this is often a more difficult procedure than for temporal employees. Neither it is desirable from an organizational perspective as dismissing full time employees often means a drain of knowledge and expertise from the organization.
Another way to adapt the skill pool is to provide training for the existing workforce. Through training, new skills can be obtained by employees, making them eligible to perform a broader set of tasks. Advantages of this approach are a greater flexibility in the workforce. [45] recognizes a lower degree of efficiency as a disadvantage for a work force with resources that are broadly skilled. Also, when higher educated resources must do work that is normally performed by lower-educated employees, demotivation under resources might occur.

2.1.4. Limitations

Earlier work describes operational aspects and technical aspects of workforce planning. However, to make general descriptions, many simplifications and assumptions have been made to the workforce planning process which makes it hard to apply proposed mathematical models in practice.

Most work describes on how the workforce can be adapted in skills and size to face changes in the resource demand, but only few describe ways to actually forecast this demand. Capelli [16] even suggests to not construct elaborate forecasting models on the resource demand, because this is under influence of too many uncertain factors and rests on too many assumptions. Instead, he claims in his work that one can better look at the uncertainty of prior estimates and the mismatch of cost of being wrong about demand to adjust our forecasts to the risk of uncertainty.

In this research we attempt to mitigate these limitations in the workforce planning process by predicting the resource demand through a machine learning approach. Our assumption is that we can make a reliable forecast on demand by combining multiple influential factors into a set of predictive models. This is in contrast with Capelli's suggestions to not create elaborate forecasting models, but we assume that this is possible when using modern machine learning methodologies that do take into account historical data, as well as the uncertainty observed in prior estimates.

2.2. Customer Engagement Life-cycle

The Customer Engagement Life-cycle describes the process from sales lead to project delivery for service-delivering consultancy organizations. We can differentiate between a sales phase and a delivery phase as the main phases in the customer engagement life-cycle, which we describe both in more detail in the following sections.

2.2.1. Sales Process

New work for services-oriented organizations is a direct result of sales. Part of the services organization therefore is a sales team that engages with (potential) customers, and is keeping track of the progress of sales opportunities. As part of the customer engagement life-cycle the sales phase is situated as the first phase in the cycle, as represented in figure 2.1.

The sales process is often seen as a funnel, where we move from a large amount of leads in the beginning, to a select number of opportunities that make it to the deal phase and eventually become wins. In terms of the Customer Engagement Life-cycle, Kulkarni [39] described in more detail four consecutive steps in the sales process.

In the initial phase the seller identifies new service opportunities. Many leads are submitted to the sales team from the marketing department, others come directly from the seller or from consulting resources that work closely with the customer. Then, in a second so-called qualification phase the team checks if the there is the time and expertise to actually pursue an opportunity, as well as the check is made if a customer has the resources to fund the proposed service project. In the third step a proposal of the to-be-delivered services is made towards the customer. At this step there is a reasonable clear view on the services that are meant to be delivered, and thus also on the resources that will be required. The fourth stage is about negotiating and closing the deal with the customer. After this point, and if the opportunity has been won, the delivery process starts. If an opportunity gets lost along the way, the reason usually gets documented. An opportunity might also be disengaged from the services organization's side, for example when an opportunity is less likely to succeed in respect to other sales opportunities.
2.2. Delivery Process

Delivery includes all activities that are performed to successfully commit work to a client. It follows up the sales phase in the Customer Engagement Life-cycle, and as a consecutive step the amount of work that has to be delivered is bounded by the outcome of the sales process i.e. signed contracts. The delivery phase can be subdivided into multiple activities, and starts at the moment that a contract is signed. Moreover, there is a change in the terminology that is used. Sales opportunities convert into one or multiple projects, and the total work that is yet to be delivered is referred to as the ‘backlog’.

First, resources are selected based on their availability, skills and seniority level. Then, dependent on the resources’ availability and customer wishes, a plan and agreement are made on how the work is going to be delivered. At the project start, resources will start working on their tasks. On a regular basis, the resources book the hours that they have made. This helps the management to better understand how much work has been delivered already, and it also allows to invoice the customer on a regular basis for the hours that have been delivered so far. During the execution phase, contracts, plans and budgets are regularly checked to make sure that delivery is going as intended.

When all the work has been delivered, the engagement with the customer and the contract end, and the last invoices will be send. The last activity in the delivery phase is to provide customer support and help the customer to maximize the adoption of the delivered services. Successful cases might raise the interest and demand from a customer to adopt more services. This will lead to new sales opportunities, and thus initiating a new cycle.

2.2.3. Difficulties

Due to the inflexibility of the capacity planning process, it is highly desirable to research to what extend the workforce planning process is predictable. The sequenced set of activities in the Customer Engagement Life-cycle show the importance of opportunity forecasting. Hence, inaccurate forecasts will lead to a mismatch between the resources that are available and the work that needs to be delivered. If the demand for resources can be known at an earlier stage, this would have a large impact on the planning process and the speed and quality of the delivery process. Looking at the causal nature of the Customer Engagement Life-cycle however, the only way to make the resource planning process more predictable is to provide early insights on the resource requests that will lead from the sales process.

2.3. Data Science Perspective on Workforce Planning

Recent developments in the industry, have introduced a new phase in the digitalization of our society, also known as the digital transformation. Development in technology, have brought us in the position to be able to process and analyze large amounts of data. With information systems that have been storing large amounts of process data for the last decade we can now defer business insights, through a combination of programming, mathematical models and domain knowledge. These developments enable us to create data products like personalized recommender systems and predictive models, applications that were merely possible in
2.3. Data Science Perspective on Workforce Planning

theory 10-15 years ago. Also in workforce planning, data science can make a large impact through e.g. smart scheduling of resources, the composition of optimal delivery teams of resources, and the construction of predictive models on future demand as described in this work. Two important methods in data science are data wrangling and predictive modeling. We describe them in the following subsections.

2.3.1. Data Wrangling

Preparing data for data science applications often requires the formatting of field into new formats, coping with missing and faulty values, and combining multiple features into new features. These data processing steps taken together are known as ‘data wrangling’. For example if one has two systems with in one system a data column represented as \( yyyyymmdd \) and in another system represented as \( ddmmyy \), then we wish to convert both columns to one and the same type. This will allow us to use them together and combine data sets.

Feature engineering is a rather creative step, in which one leverages domain knowledge to construct features that make more sense to the problem domain and make machine learning algorithms work better. According to [52] features are either projected on the problem domain (number of consultants on project \( p \) for customer \( x \)) or on the solution domain (\( h \) delivered hours by resource \( r \) with role ‘consultant’ on date \( t \)). Feature Engineering therefore is “the disciplined projection of features from the problem domain into the solution domain”. It bridges the gap between different views on the system.

For example from a dataset with daily delivered work \( h \) by some resource on a project \( p \) we can make an aggregation of the hours by creating a mapping \( p \rightarrow h \), representing the total hours made by resource \( r \). Combining the work of multiple resources allows to create a view on the total delivered work on a project \( p \rightarrow h \). Other examples of engineered features are a feature for a certain project \( p \) listing the number of resources that have worked on \( p \), or a feature describing the total number of hours on a project by resources with a specific role.

2.3.2. Predictive Modeling

Predictive modeling is a technique in which we attempt to create a model that estimates the likelihood of a certain outcome. Unless the names suggests, predictive models are not merely used to make predictions about the future. In principle, the technique is used to make predictions about any unknown events, based on historical facts and current observations. It is a field that combines analytical techniques from statistics like regression, with machine learning techniques like neural networks and support-vector machines to construct predictive models for a wide variety of problems.

To construct a predictive model, historical observations are used as example i.e. training data, to fit the model. The algorithm determines the way of learning and aims to find relations in the training data. This makes each model unique in terms of the chosen algorithm, model parameters, and features that were used as model input. In order to test how well a model has been trained, its performance gets tested against a set of unseen test data. In this way, it becomes clear how well the model performs on real-world data.

Moreover in the case of machine learning models for which can set model parameters (such as the number of nodes, the number of trees, etc.), a validation set is used before testing the model’s performance to find the optimal model parameters. This data set is used to compare the performance of the multiple algorithms that we fitted on the training data, after which the best performing model gets selected. Commonly, the set of data is split and sampled into a training, validation and test set with the largest amount of observations assigned to the training set, for example following a 60/20/20 distribution.

Each modeling technique and machine learning algorithm has its own strengths and weaknesses, and makes it suitable for a particular type of problem. Below we group a number of the most common machine learning algorithms and shortly describe their applications.

- **Anomaly Detection** is a type of machine learning models that is used to discover irregular patterns in data. Typically, anomalous observations refer to some kind of abnormality in the problem domain like fraud, system intrusions or any other kind of outliers. In consultancy, anomaly detection may be used to identify high risk projects, projects that face issues during the delivery phase or to identify abnormal expenses.
• **Classification** is used to split the dataset into two or multiple classes. In the case of two classes we speak about binary classification, in other cases we refer to this set of algorithms as multi-class classification. This kind of algorithms build on a set of labeled examples as training data, and classifies new observations into one of the predefined classes based on similarity or distance-based measures. Examples in the problem domain are sales classification, identification of project delay, or the clustering of projects into low and high risk.

• **Clustering** is a technique that can be seen as similar to classification, expect for the large difference that clustering is a form of unsupervised learning. This means that the algorithm itself tries to find and assign observations to clusters (classes) based on similarity measures. Clustering can for example be applied to find clusters of similar projects, similar customers, or similar product and services.

• **Regression** is the type of machine learning algorithms that is used to find the relationship between one or more independent variables on a dependent variable. A model is iteratively refined to minimize the error between the fitted regression line, and the actual data. In the case of one independent variable we speak about simple linear regression. Examples within the regression domain are the estimation of project value based on project characteristics, numbers of required resources or project lengths.

• **Time Series Prediction** leverages a chronologically ordered set of points to infer trends and seasonality, and to predict a future series of points in time. Examples of time series prediction applications are forecasts on the delivered hours per day, or forecasts on sales.

The above model types show a general grouping of the available machine learning algorithms, but variations between the various types are possible, as well as models that are especially designed to fulfill specialized tasks like feature selection. Models that are commonly used in machine learning and combine elements from multiple of the above groupings are decision trees, Bayesian and neural network algorithms, which we discuss shortly below.

• **Decision Trees** are a set of models that build hierarchies of decisions based on attributes in the training data. Decision trees are often used for regression and classification problems, and are considered fast and accurate. An example of a decision tree might be a model to determine project delay.

• **Bayesian Algorithms** use Bayesian logic to construct regression and classification type of models. Bayes’ theorem provides a methodology to determine the probability of a certain hypothesis, given our prior knowledge. Bayesian Algorithms work well for classification type of problems such as sales win classification.

• **Neural Networks** are used for complex problems with a large amount of input parameters. The technique leverages a large set of interconnected nodes, linked by weighted connections, to learn patterns. The technique is inspired by the human brain. Neural networks are complex and require training of the adaptive weights on the links by a predefined learning function. Neural networks are often applied on complex problems such as resource scheduling problems.

### 2.4. Related Work

This section discusses earlier work on applying machine learning techniques in the field of workforce planning and its relation to demand forecasting. First we discuss a selection of related work discussed in academia. Then, we highlight three projects in specific that have been worked on before at Microsoft.

#### 2.4.1. Related Work in Academia

Besides related work on workforce planning as referred to throughout this chapter, there are a number of projects described in literature on the topic of demand forecasting for the purpose of workforce planning. An article [44] on the topic of increasing sales through win/loss classification describes domain concepts and questionnaires for conducting a sales win analysis. The authors mention sales win/loss classification through machine learning as a large data mining opportunity. The larger part of the literature on resource demand forecasting in relation with machine learning is published by IBM Research, describing forecasting work to support the capacity planning for their service-delivering organization. Two papers [56, 57] describe predictive modeling work on sales pipeline analytics. Another paper [21] describes resource demand forecasting
through the building of project taxonomies. Gilat et al. [25] describe an approach to shift work optimized planning and scheduling (SWOPS). It is a workforce management tool designed for shift work in multi-skill environments. The problem is formulated as a scheduling problem and is solved through a tool that leverages integer linear programming to create work schedules. We highlight three papers that are especially of interest to our work.

- Yan et al. [57] describes sales pipeline predictive analytics as a win-propensity prediction problem. They describe that in contrast to otherwise human subjective ratings on sales win-propensity, they evaluate sales wins through a machine learning approach wherein seller’s activities influencing the win outcome are evaluated, together with lead’s personalized profiles. Moreover they found that sellers to focus their efforts on only a few leads during a relatively and short time, and also that opportunities tend to reach the win stage shortly after these interactions.

- Hu et al. [32] researched a systematic, repeatable approach for determining the staffing requirements. They describe a methodology to generate automated staffing plans involving roles, skills sets and experience levels based on a project taxonomy that has been derived through clustering of labor records.

- Continuing work by Hu et al. [21] on [32] describes a data mining technique for discovering clusters of consultancy projects that show a similar resource usage over the project life-cycle. They leveraged the clustering results and domain expertise to create a taxonomy of related projects, and staffing templates that can be linked to project resource requirements. The authors formulate the problem as a sequence clustering problem in which sequences represent weekly distributions of resources usages over a project.

2.4.2. Related Work at Microsoft

Related to this research, a number of similar projects have been done before at Microsoft. Traditional Business Intelligence applications and reports prevail, but increasingly more predictive analytics are leveraged within the company to support business decisions. Within the scope of services two projects have been performed on forecasting sales revenue that is coming from the sales pipeline. One focuses on the forecasting of aggregated sales revenue, the other attempts to face this problem on a project-level. Below, we discuss both projects in more detail.

A first project focused on forecasting the aggregated sales revenue at Consulting Services by using data from the sales pipeline. Objectives of this project were to reduce the amount of time that people are spending on forecasting within the company, as well as to improve the forecast in respect to the human-judged forecast. The team eventually used elastic nets, after having evaluated diverse time series and regression models to create their forecast. Moreover, model performance has been evaluated over time by using a rolling-window based approach. Eventually, in 72% of the cases the team was able to outperform the human forecasts on the expected sales revenue.

The other project by the Enterprise Services Business Intelligence (ESBI) Team is more similar to this research, and focuses on creating a forecast that is on the granularity-level of projects. In their approach, they distinct two clear phases. In a first phase, current opportunities in the sales pipeline get analyzed and classified according to their characteristics to either become a win, or a loss. In a second phase they use regression techniques to calculate the cumulative burn rate of backlog-value to forecast the aggregated revenue per month. Disregarding the large variance between the predictions per region, the team achieved a model performance of 3% bias between the model predictions and the actual revenue.

At the moment of this research, the Data and Decision Sciences group is also working on a similar project, to predict the resource need for the Americas region. The main goal of this project is to provide a proof-of-concept that the resource demand forecasting problem can be solved through machine learning, similar to this project.
In this chapter we describe the overall methodology that we have used during our study to answer the research questions as defined in chapter 1. First, we describe in section 3.1 our overall research methodology. Then in section 3.2.1 we zoom in on the process of data science and on predictive modeling in specific. Lastly, in section 3.4 we describe an evaluation framework to assess the performance of the constructed predictive models and framework.

### 3.1. Research Overview

The current state-of-the-art in demand forecasting mainly focuses on predicting results from sales by analyzing historical sales results, fluctuations and trends as described in chapter 1. Moreover, demand forecasting work has been performed on specific subtopics of workforce planning as described in chapter 2, but as far as literature goes the various components of workforce planning have not yet been combined into a forecast that is made up through the composition of multiple predictive models.

In this work we aim to provide a machine learning framework that can serve as the baseline solution for future predictive modeling work on the topic of workforce demand forecasting, and in the context of service-delivering organizations. As well we want to provide a solution to make the demand for resources more insightful from a business perspective.

### 3.2. Research Design

This research is a quantitative study based on data made available by Microsoft Services. We use advanced mathematical and statistical methods to provide answers to questions from both a research as from a business perspective, as defined in the introduction. Although described in general for a services-delivering organization, elements of this research may be specific for the processes at Microsoft. Therefore, we provide a contextual view and domain analysis in chapter ?? where we discuss processes, scope and limitations specific to the domain.

During our study, we followed generally an iterative data science approach that consists of partly domain research, experimentation, implementation and evaluation phases. Analysis of literature and gained domain knowledge through a case study at Microsoft, as well as input from stakeholders, defined the direction and path of this research along the way, and helped to realign focus to find answers on the defined research questions. An overview of our overall methodology is well-displayed by figure 3.1. The figure shows the position of each step in the methodology and the moments of interaction with the organizational processes. Starting with a basic understanding at the beginning of the project, we enter a iterative research process, in which process steps are regularly repeated. Moreover, the figure shows a feedback loop created by the data science process: new data insights influence decision making and processes.
3. Methodology

Our approach to find answers to the defined research questions around the problem domain was the following. To find the key dimensions involved in resource demand forecasting during the domain research step, we used a combination of literature reviews and interviews with around 30 people within the Services department at Microsoft. The results of this background study are described in chapter 1, 2, discussing literature and processes, and as well in chapter ?? where the utility of visualizations and tools is discussed.

To answer research questions around the topic of the feasibility of predictive models, we followed the data science work flow as described in more detail in section 3.2.1. To evaluate the quality of the made forecasts and to evaluate the model’s ability to outperform a human forecast, as part of the business research questions, we introduce an evaluation framework in section 3.4. To determine the extend to which the actual resource need can be predicted in practice, we combined the proposed predictive models through a machine learning framework and assessed the produced forecasts with actual data. Chapter 11 describes this in more detail.

3.2.1. The Data Science Work Flow as a Methodology for Predictive Modeling

This work is a typical data science project covering a variety of tasks ranging from domain research to the iterative implementation and optimization of machine learning algorithms. As a reference methodology we used the data science process as described by Schutt and O’Neill [47], and custom-tailored this methodology to our specific needs in this project, resulting in the data science subprocess as shown in figure 3.1. Below we describe the data science process step-by-step in detail.

Schutt and O’Neill describe the data science work flow as an iterative process that through raw data processing, cleaning, exploratory analysis and modeling leads to machine learning models that are used to build data products and visualizations. Moreover, they describe a feedback loop from new data products to raw data, where the process starts over again. The major difference between Schutt and O’Neill’s general methodology and our methodology is the large process and data understanding step. As well as the fact that machine learning models are leveraged into interactive visualizations and management dashboards. Dashboards provide new insights, support managerial decisions, which get adopted into processes, information systems and eventually leads to new data and models.

Solving a data science problem starts with defining a well-scoped research question, and problem formulation, similar as in the traditional scientific process. Often the data scientist is limited in his knowledge about
the problem domain, thus requiring either the collaboration with domain experts in the field or acquainting
the essential domain knowledge through studying. Having domain knowledge is important, as it is essential
to understand the meaning of the information represented in the data, to detect correlations, or to identify er-
ors that might be present in the data. The initial project phase is therefore defined by doing domain research.
This includes conducting interviews with different stakeholders to understand the process, the identification
of relevant work, the identification of relevant data sources, and arranging access rights to connect to
these data sources.

The phase that follows consists of analyzing and wrangling the data. By processing the data, it can be shaped
in such a way that it can be consumed as input to a machine learning model. Processing includes the cleaning
of data, the combination of data from multiple data sources and the pre-selection of features. New features
can be generated as well by combining data from the data sources with our domain knowledge, this is also
referred to as 'feature engineering'. For example from a list of resources we can extract the number of con-
sultants and the number of architects. Data cleaning is the process of enhancing the quality of the data by
removing rows that contain missing values, by replacing missing values with replacement values such as av-
erages, and the fixing of errors in the data. Exploratory data analysis helps to get a feeling about the data, to
find correlations, and to identify data problems.

With the data shaped in the right format, a machine learning model can be fitted on the data. Data gets hereby
splitted into multiple sets with at least one set for model training, and another set for testing the model's
performance. Dependent on the problem domain, we decide on which machine learning model type to use
as discussed in section 2.3.2 After training, a model's performance can be validated against the test set. A
usual way for the data scientist to optimize performance, is to evaluate multiple different algorithms on the
dataset, with different selections out of the features in the dataset, and with different model parameters. For
unbiased training, the test set gets splitted multiple times in a test and validation set through k-fold cross
validation. One set is used for model optimization, and one for testing the model's performance on unseen
data. All together, this makes machine learning a highly iterative process.

Lastly, by visualization of data and model outcomes, results can be effectively communicated to stakehold-
ers. Combining a variety of plots and figures into dashboards typically provides the most value for business-
related projects. Another possibility is the integration of model outcomes into data-products such as resource
planning tools.

3.3. Data Collection

For this work we have access to data made available by Microsoft Services. Multiple data warehouses exist that
combine data from various sources and (legacy) systems. The particular data warehouses are relatively large
in size, featuring hundreds of views and tables, requiring a well business understanding to extract relevant
data. Besides the difficulty of size, which makes it hard to see the forest for the trees in terms of data, many
redundant data exist and also views change continuously as the business is changing. Data is available for a
time period of more than 10 years featuring close to 50000 projects. However for the reason of data quality and
a changing business, early analysis shows 11500 projects as relevant for analysis considering only delivered
projects from over the last 5 fiscal years.

As mentioned above, besides the quantitative data available from the data warehouses, information about
business processes has been retrieved through the conduction of interviews. The people interviewed were
diverse, working in various roles, and having different levels of experience. We did not follow a strict interview
format, but rather asked people to describe their roles and difficulties experienced in everyday tasks.

Moreover during domain analysis, contact has been established with multiple teams overseas in the United
States who have worked on related work before. This contact led to the sharing of best practices, experi-
ences, but also to new data including model outcomes, classifications and related propensity scores for sales
pipeline classification.
3. Methodology

3.4. Evaluation Framework

In this section we discuss the evaluation of the obtained results in this research. Regarding the aspect of evaluation of machine learning work, a number of the most relevant evaluation metrics are described in this section. Later in chapter 7, 8, 10 and 9 we describe for each predictive model which error metrics have been considered in specific as the chosen metrics are dependent on the problem type.

3.4.1. Criteria of Success

Given the research and business objectives as defined and introduced in chapter 1, differentiation can be made between quantitative objectives and qualitative objectives. Success on qualitative research objectives’ is analyzed by the following criteria.

1. Could a bird eye-view be constructed on data covering the whole sales and project life-cycle?
2. Have new insights been delivered on the future demand for resources?
3. Do the delivered insights support the resource manager in his task?
4. Do the delivered insights help the capacity manager in his task?
5. Have we been able to construct a notion of risk in forecasts?

Quantitative goals can rather be analyzed by their performance. Machine learning algorithms can be evaluated by their predictive power, and research questions like ‘is demand influenced by trends and seasonality’ can be answered by facts obtained from the data. Moreover we can set the following criteria of success:

1. Can the predictability of a certain model type be expressed?
2. Does a predictive model outperform the human forecast?

The next section introduces a number of evaluation metrics for evaluating the performance of predictive models.

3.4.2. Evaluation Metrics

In this work we describe a number of regression and classification problems. Diverse error metrics can be applied for evaluation, but in general the used error metrics dependent on the problem type. Therefore we describe later for each predictive model which error metrics have been considered in specific.

Equations 3.1, 3.2, 3.3, 3.4 and 3.5 are commonly used error metrics to evaluate the accuracy of regression type of models. The absolute or squared errors are taken so that error values are always non-negative. R-squared (equation 3.5) is commonly used to describe the ratio of the variance that can be explained by the fitted regression model to the total variance. In other words, it describes how much of the variance can be described by the model. Each measure is slightly different, and it depends on the type of problem and the data which evaluation metric is preferred.

\[
\text{Squared Error (SE)} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2
\]  

(3.1)

\[
\text{Root Mean Squared Error (RMSE)} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}
\]  

(3.2)

\[
\text{Mean Absolute Error (MAE)} = \frac{1}{n} \sum_{i} |y_i - \hat{y}_i|
\]  

(3.3)

\[
\text{Relative Absolute Error (RAE)} = \sqrt{\frac{1}{n} \sum_{i} |y_i - \hat{y}_i|}
\]  

(3.4)
3.4. Evaluation Framework

\[ R-Squared = 1 - \frac{\text{Summed Squares Error}}{\text{Summed Squares Total}} = \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \quad (3.5) \]

For time series models, equation 3.6 is a more suitable metric, because of the sensitivity of metrics 3.1, 3.2, 3.3, 3.4 and 3.5 to outliers, and scaling differences [40].

\[ \text{Mean Absolute Percentage Error (MAPE)} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\hat{y}_i}{y_i} \right| \quad (3.6) \]

There are many different ways to consider the performance of classification problems. Precision and recall (equation 3.7) and 3.8 are the most commonly used error metrics in computer science. Typically, the outcome of the classification is compared to a reference classification to identify how well a model is performing [6].

\[ \text{Precision} = \frac{TP}{TP + FP} \quad (3.7) \]
\[ \text{Recall} = \frac{TP}{TP + FN} \quad (3.8) \]

For classification problems, confusion matrices are commonly used to describe the performance of a classifier in terms of true positive, false positive, true negative and false negative cases. Moreover, evaluating the ROC-curve (Receiving Operator Characteristic) shows the performance of a classifier to distinguish classes. It is a useful method to determine a cut-off value to separate between two classes, dependent on if you optimize for precision or for recall. The area under the ROC curve (AUC) is a measure that indicates how well a parameter distinguishes between two classes [46].
Due to the confidentiality of the information represented in this chapter, the content of this chapter has been removed from the public version of this work.
Since the novelty of data science projects that span across the whole consultancy organization of Microsoft, a large amount of work in this research has been devoted to data wrangling work. In this chapter we discuss the data preparation work performed on the different data sets, and how the loosely coupled data sets can be combined to construct a bird-eye view on the trajectory of a project from sales to delivery. Moreover, we discuss observed issues regarding the data quality.

5.1. Systems and Available Data Sources

Corresponding to the different sales and delivery phases as discussed in chapter ??, we identified numerous systems that get used over the phases and for each specific problem-domain at Microsoft. Due to developments of the business over time, business requirements have changed, causing many legacy systems to exist. This results in a large landscape of relatively sparse connected data sets spanning over the full customer engagement life cycle. Below we list the most important data sets and systems, as well as their contribution and position in the workforce planning process.

- **Sales Pipeline (MSX)** - the system used for tracking and managing sales opportunities.
- **Customer Data (CRM)** - the Customer and Relationship Management system for managing interactions and relations with customers, as well as their contact information.
- **Sales Forecasts (FMT)** - Sellers make forecasts on the likeliness of certain opportunities to become a win. These forecasts get registered in the Forecast Management Tool.
- **Early Staffing Requests (Simplify)** - To support the resource planning, an early estimation of the expected resource need is registered in Simplify.
- **Deals and Contracting (Compass)** - Compass is the system for the registration and management of deals and corresponding contracts.
- **Engagement/Project Management (Changepoint)** - In Changepoint, engagements, projects and tasks are set up and managed.
- **Time Tracking (Chronos)** - Built as a layer on top of Changepoint, Chronos allows resources to register their delivered hours.

Figure 5.1 gives an overview of the most relevant systems along the process. As well it provides insights into the moment of availability for each system.

Access to the data underlying the various systems is available through a number of data warehouses. One is the data warehouse for the Enterprise Services Business Intelligence and the other data warehouse contains all sales-pipeline related data.
5.2. Constructing a Bird's Eye View on the Process

In this work, a view is created on the course of an opportunity/project through the full process of the Services organization, which in practice is a combination of data sources that has never been made before. Given this fact, no earlier work is available that could be used as reference material. Neither, the different systems involved in the process do have a lot of commonalities in terms of data logic and identifiers.

For this reason, a first step to take is to create a view on the process from initial sales lead, to planning, to delivery for each delivered project in the past, by combining the different data sources. This will allow to compose views to serve as training data for machine learning models, for different stages in the process. As can be seen in 5.1, more data becomes available along the process. With the more data we get, the certainty increases regarding the resource need. Therefore, it is reasonable to assume that when a unique machine learning model is constructed for each sales stage, the model accuracy would improve.

In order to combine two or more data sources, common logic and identifiers need to be identified by which the data sources can be merged with each other. At Microsoft, each sales opportunity can be identified by its unique ‘opportunity-id’. This identifier is used throughout the further sales and delivery process to refer back to the originating sales opportunity. From the point that an opportunity moves to the delivery phase, a one-to-many relationship is adopted in which an opportunity might be split into one or multiple engagements. Moreover, an engagement is split up into one or multiple projects, and a project in multiple tasks. As mentioned before, in this research we focus on the granularity of projects. From a data combination perspective, this requires a number of complex data manipulation steps in order to construct a combined view.

5.3. Process Progress Dependent Views

Dependent on the stage of an opportunity in the sales phase, certainty increases as more data sources become available. This creates the opportunity to train multiple machine learning models for the different quantities of data for each stage, to enhance accuracy. A number of different views on the data therefore are constructed to create a unique training set for each sales stage. Hereby, distinction can be made between the following five combinations of training data.

- **Early Opportunity** - Sales Pipeline, Product, Customer + Actual Delivery Data
- **60% Opportunity** - Sales Pipeline, Product, Customer, Early Staffing + Actual Delivery Data
- **80% Opportunity** - Product, Customer, Contracting, Risk + Actual Delivery Data
- **Backlog Projects** - Product, Customer, Contracting, Project Planning, Risk + Actual Delivery Data
- **Active Projects** - Product, Customer, Contracting, Project Planning, Risk, Delivery + Actual Delivery Data
In addition to each of the combined data sets as listed above, actual delivery data is added. Actuals are included in the training phase to fit the machine learning models, and is the type of information that we aim to predict for in future projects.

Data that becomes available later in the system can be regarded as more certain. Since contracting information includes more detailed and accurate information about products and project values, pipeline data is not longer needed and disregarded for later stage projects.

5.4. Feature Engineering

Often, data is not available in the most optimal representation to serve as input for a certain type of problem. Or multiple features might be interdependent, leading to more or less redundant features. In these cases it might be worth to construct conjunctive features as a product of multiple features, or to represent data in another or aggregated form. Interdependent features can be identified by computing the linear correlation between features in a data set.

These kind of data transformations have been applied multiple times on our data sets. For example, a project might have multiple resources assigned, subdivided into multiple tasks with corresponding hours, roles and further details. Since we would like to know the number of resources assigned to a project of a specific role, the field of role had to be spread into the binary representations of consultant, architect, etcetera, on a task level. This representation subsequently allows for a second transformation that counts up all the consultants or architects from a task level to a project level, resulting in the features of resources.no or resources.no.consultants that we were looking for.

5.5. Feature Selection

Selection of data features, to use as an input to a machine learning model, has a multi-fold objective as described by [28]: to improve the predictive power of the predictors, to find more efficient predictors, to better understand the process that is underlying the data, to reduce storage requirements or to reduce the time required to train a predictive model. Having knowledge about which features are relevant, gives direction to the search for other features, and allows to derive new insights.

In this research, we perform feature selection techniques primarily with the objective to construct subsets of features that are contributing to good predictors. [15, 28, 37] are papers that discuss the difference between usefulness and relevance. The authors claim that having multiple redundant features is suboptimal for the performance of a predictor. Therefore the authors recommended in their work to find a subset of useful features that exclude redundant, but relevant, features.

Multiple feature selection methods moreover exist. An initial selection of the features can be done by leveraging domain knowledge. This form of ‘pre-selection’ allows for the selection of features which we know or assume to be relevant, and to disregard features that are known as irrelevant or containing too many missing values to be useful in a model. After pre-selection on the data sets, multiple further selection approaches can be taken. One can continue to select features based on domain knowledge. Another technique is filter-based feature selection [28], which is used to identify the features with the greatest predictive power by ranking them with correlation efficiencies [28]. Moreover, one can start by selecting all features, and improve a model’s performance by reducing the size of the feature set. Or the other way around, by starting with zero features and including additional features until the model performance does not longer increase.

In this work, we used a combination of the above techniques. Since each model is influenced by its own set of features, we discuss feature selection results for each predictive model in detail in the next chapters.

5.6. Issues and Limitations

Combination of different data sources is not a trivial practice. Missing key values, incorrect references and different business logic over the systems cause a lot of redundancy, noise, and overhead columns, which are issues and limitations that must be overcome. In this section we discuss the multiple categories of issues encountered in this work, and the approach taken to enhance the data quality and to reduce limitations.
5.6.1. Identifiers and Loosely Coupled Systems

Over the Services data warehouses, each system serves its own purpose, yielding a minimal amount of shared data logic and referencing values between the systems. This loosely coupling between the system implies a large challenge to connect the different data sources. Regularly, identifying values were lacking at all, making combination of data sets unreliable or rather impossible. Moreover, categorical values use different encoding and values to represent the same concepts, requiring many manual mappings in order to combine the systems.

One of the most important examples of identifiers is the ‘opportunity-id’, and can be seen as the main ‘connector’ among the systems. This column can in practice however been observed in two different representations that are both used in order to reference to, and identify unique opportunities. Since a system either uses form A or form B, opportunity-id’s need to be mapped to a single representation. Some data sets do contain by default a reference to a corresponding opportunity, other data sets require multiple joins on other data sets to identify the corresponding opportunity-id’s.

5.6.2. Granularity of data

On several areas, different levels of granularity between data from different sources made this work more complex. Because of the differences in granulation between the project and task level information, task data has been aggregated to a project-level. This aggregation includes for example the sum of planned, baseline and delivered hours grouped by each project. In other cases, data at an engagement-level of detail is combined with data on a project-level to create the most informative view while preserving as much information as possible.

A separation of the opportunity and engagement into projects is made in both the deal phase as in the delivery phase. During the deal phase, contracting information is available, as well as the results of the risk assessment survey. During the delivery phase, planned, baseline and actual delivered hours are available. Ideally, a comparison can be made between the initial planned hours in the deal phase, and the eventually made hours in the delivery phase. In practice however, projects can not be mapped to each other between the different systems. What happens is that data gets transferred manually (by typing over text) from the one system to the other, loosing all referencing columns expect of opportunity-id’s. Due to this fact, the only way to combine project data in the deal phase, like risk information, with project data in the delivery phase, is to aggregate it up to an engagement level and then combine it again with data from the delivery phase. With this transformation, detailed information however gets lost. To preserve as much information as possible regarding risks, we adopt the highest risk value observed among multiple projects in the aggregated representation.

5.6.3. Low Data Quality

Low data quality moreover is an issue that is widely-spread over the systems at Services. Besides missing key values, missing values in general are present at many places. As most machine learning models are not able to handle missing values, missing values must either be replaced by a replacement value or the row or column must be completely discarded. In practice, for each feature that contains missing values another approach must be taken to enhance the data quality. Another kind of risk-full data problem is the presence of large categorical features. When using decision-tree like algorithms, it is easy to over-fit the model on categorical features that have many different levels. Variable reduction techniques can hereby help to reduce the number of levels. For example, for the ‘Primary Product’ feature, the levels of ‘Office 2010’, ‘Office 2013’ and ‘Office 2016’, can be reduced to ‘Office’. In the case of missing categorical values, a common practice is to replace the missing values by an ‘unknown’ category. In this case the machine learning algorithms will be able to handle the missing values.

Often, data quality issues emerge from the source of the system. Some systems are as such outdated that forms do not provide the possibility to select the right input values. Also, regularly fields provide to less boundaries because of data that can be entered through free text fields. This leads to opportunity-id’s that get entered wrongly, and other forms of noise in the data.
5.6.4. Changing Environment and Systems

As the organization is changing, services offerings, products and processes change, and so do the information systems. Often data changes are not limited to migrations of existing systems to new versions, but also includes the introduction of completely new systems and the depreciation of old systems. Changes affect the representation of data and available columns in the data warehouses, requiring adaptation of SQL-queries.

The art is to find both features that are robust over time, as well as to reduce the impact of environmental changes on the data. One measure is to apply reduction techniques on categorical variables as described above.

5.6.5. Process Hacking

‘Clever people know to manipulate clever processes’ is a sentence that could summarize the whole data collection process at Services. Due to a target-driven culture, people are not dedicated to fill data into the systems in a complete or right way. Measurement of people’s performance through the data in the systems is due to this, and people find ways to manipulate the data in their own advantage.

For example, in the sales department there are metrics on the number of deals that get won or lost by certain sellers. Normally, if an opportunity gets lost because a customer decides to go for a product of a competitor this will be registered into the system. However, some sellers found a way to trick this process, by resetting the due date to the moment that the competitor’s license expires and the customer might choose for Microsoft again. Then the opportunity gets marked down as slipped rather than lost, but the data is in fact corrupt. This examples shows how subjective the data is to the underlying process and involved people, and how culture causes noise in the data. In this particular example, the data quality could be improved by marking down an opportunity as a loss after a certain period of inactivity.

Other ways how people misuse the systems are amongst others the on purpose underestimation of deal size, on-purpose delaying the commit to a next sales phase, the late registering of the largest deals, registering only the minimal amount of deal-related information and booking hours under a higher-valued booking category.
A Machine Learning Framework to Forecast the Resource Need

6.1. Introduction

As influenced by numerous internal and external factors, the resource need is a complex figure to predict. In the context of a service-delivering organization, the demand for resources is not a figure that is completely random defined, but is largely a direct result of sales, which bounds the demand for resources until a certain height. Hence, zero result from sales will not require any new demand for resources.

When looking at a certain future period, we can differentiate between multiple demand generators. First, there are already running projects requiring the resources that are already active on their tasks. Secondly, there are projects that rely in the backlog and have not yet started. Lastly, there is work that is yet to be sold i.e. work originating from sales. Together they form the expected backlog of work.

Overall, forecasting the resource need requires looking at three dimensions. That is workload, how much of work is likely to come from the sales pipeline and backlog? The resource mix, which roles will be needed? And timing, when can the work be expected? The multiple dimensions decompose the complexity of the resource need into a less complex form.

In this chapter we discuss the decomposition of the problem into a number of dimensions (section 6.3) and corresponding definitions of subproblems (section 6.4). We discuss multiple predictive models and introduce a framework that allows for the combinations of the output of the predictive models, to make a forecast on the future demand for resources. In the next section, we discuss our objectives while constructing this framework.

6.2. Objectives

Creating a forecast on the demand for resources requires a deep understanding of the quantity and type of work that is currently being delivered and of work that is about to result from the sales pipeline. As well the dimension of time i.e. project length, project start, and hour utilization, largely defines the total demand that is spread over a certain time period.

Suppose we have a set of completed projects \( P_1, P_2, ..., P_n \) that have been fully delivered. For these projects an extensive analysis can be made on the full life-cycle from sales to delivery. For each project there is a team of resources \( R_1, R_2, ..., R_n \), where each resource is labeled according to a technical domain \( D \) and a specific job position \( J \) defined as \( R \rightarrow (J, D) \). Analysis of daily registered work in hours \( H \), (datetime, \( R \)) \( H \), allows for the reconstruction of the resource mix, the actual project length \( l_{\text{actual}} \), the workload over time and allows for analysis of requirements for certain skills in specific phases of the project delivery.

Moreover, analysis of the evolution of project value and characteristics over time allow for more precise predictions on future work. Standardized service descriptions are available along the whole process, but get
further refined when an opportunity $O$ gets split up into one or more engagements $E_1, E_2, \ldots E_n$ and projects.

Regarding project value, figures are available along the whole life cycle: a pipeline value $V_{\text{pipe}}$ is recognized to a sales opportunity from the start; during the deal phase a contract's value $V_{\text{contracted}}$ is available; and eventually the full value $V_{\text{full}}$ of work following on one sales opportunity is known. Hence, contract amendments are possible, leading to new contracts, and thus to an increased value amount. The same can be noted for hours, where starting from an early staffing request $H_{\text{early}}$, to contracts $H_{\text{contracted}}$ to project planning $H_{\text{planned}}$ and delivery $H_{\text{delivered}}$ different values are possible and likely. Differences in estimations and values among the different life cycle stages imply the construction of distinct predictive models for each process stage, in order to maximize the model's predictive power.

The different problem dimensions, data quantities and distinct predictive models for each sales stages, require the introduction of a machine learning framework that allows for the combination of data from multiple sales stages and model outputs, in order to create a complete and unified forecast on the future resource need. In particular, we seek a framework that allows for capturing the following problem characteristics.

1. Introducing a structural approach for resource demand forecasting.
2. Providing a unified view on future work i.e. expected workload, expected resource mix, expected start date and expected project length; irrespective of the sales stage originating the forecast.
3. The construction of the future backlog through combination of current backlog projects and expected backlog projects.
4. Providing a forecast on future workload build and combined on distinct predictive models for each sales stage i.e. data quantity.
5. The indication of risks associated to the forecast quality of each sales stage and the confidence about specific predictions, as well as the propagation of risks through the framework.

6.3. Problem Decomposition

As mentioned above, the resource demand forecasting problem can be regarded as a problem that can be seen from multiple dimensions. As a result, it can be regarded as too complex, and rather infeasible, to create a forecast for all dimensions based on a single predictive model that predicts a project's workload, time and its likelihood to succeed. Hence, decomposition of the problem is required for which we introduce the following separation of concepts:

1. Likelihood of sales success; expected workload
2. Resource Mix
3. Timing

For each concept or problem dimension, a set of predictive models can be defined. Through the subsequent chaining of model sets, through inter-model data pipelines, a forecast can be made that covers each concept. The combination of data processing pipelines and predictive models can be referred to as a machine learning framework. Figure 6.1 sketches the introduced machine learning framework graphically, and shows the position of each set of predictive models within the model chain.

Central in the model is the backlog of work, containing work that is currently being delivered and work that is expected to be delivered. The former part is already known, the latter has to be forecasted from the sales pipeline.

The sales pipeline is a representation of all the possible deals that are currently pursued by the business. Corresponding systems provide insights into sales statuses and corresponding service characteristics like business scenarios, products, and estimated deal sizes. From a demand forecasting perspective, we would like to evaluate the likelihood that an opportunity that is now situated in the sales pipeline will eventually lead to a win, and thus will lead to new work. Therefore a predictive component on the sales pipeline win/loss classification is introduced (figure 6.1) to forecast the likelihood of sales wins based on opportunity characteristics.
For each project in the work backlog, we would like to estimate how much effort it will require from the resources to deliver the work. In its simplest form, this workload can be measured in terms of total hours. More in specific, a distinction can be made between the different roles and technical skills required from the resources, also referred to as the 'resource mix'.

Lastly, a set of models on timing can provide insights into the moment at which backlog's projects will take place. A first objective is to forecast delay in project delivery with respect to the planned project start, secondly to estimate actual project duration, and lastly we want to understand how the distribution of work hours will be given the project duration and the project characteristics.

Section 6.4 introduces formality to the framework of predictive models, by defining for each predictive model its goal, input and output. As well intermediate processing steps are described.

6.4. Sub-models and Definitions

As described in section 6.3 the problem of resource need forecasting can be decomposed into three dimensions. In this section we map these dimensions to a set of predictive models, and define their corresponding output. Below paragraphs in bold correspond to predictive components, whereas non-bold paragraphs refer to intermediate data processing steps.

Opportunity Win/Loss Classification

A binary classification model applied on the sales pipeline $O$ to identify for each opportunity $o \in O$ to either result in a sales win ($o \in O_{\text{win}}$), or not ($o \in O_{\text{loss}}$).

*Model Input:* a sales opportunity $o$

*Model Output:* a tuple with a binary variable indicating the expected class and a propensity score on the interval $[0, 1]$: $(o, \text{win/loss}, \text{propensity})$.

For the set of predicted pipeline wins $O_{\text{win}}$, a data transformation $o \rightarrow p$ is required to transform opportunity $o$ to the format of a project $p$. For simplicity, we hereby assume that each sales opportunity converts to one work project and parent engagement. Some values are missing in the opportunity phase, yielding the challenge of finding replacement values for missing values. Values such as the ‘number of resources’ or the ‘planned project length’ can be replaced by average values from historical similar projects.

The forecasted backlog $B$ is constructed out of the actual current backlog $B_{\text{actual}}$ and the backlog $B_{\text{predicted}}$. 

---

**Figure 6.1:** A model decomposition into three sub-models, each covering different aspects.
composed out of transformed projects yielding from pipeline wins $O_{\text{win}}$, hence $B = B_{\text{actual}} \cup B_{\text{predicted}}$. Projects in $B$ have assigned one out of four risk buckets (qualified, planned, deal, or actual), referring to the sales stage (see chapter ???) in which they are positioned at the moment of the forecast.

**Workload Prediction** A set of regression models to estimate the workload of a project i.e. the number of hours needed for project delivery. A distinct model is required to score projects in each risk bucket, and corresponding data quantity and quality.

*Model Input:* Backlog project $p$

*Model Output:* the expected value in hours and a corresponding standard deviation of the predicted value: $(p, \text{expected hours}, \text{standard deviation})$

**Resource Mix Prediction** Through clustering, projects can be identified that require a similar resource mix i.e. required consulting hours, architect hours etc. Finding commonalities among projects in the same clusters, on early-stage available features, allows to give an early forecast on future expected roles.

*Model Input:* Backlog project $p$

*Model Output:* the expected resource mix: (consultants: 0.62, architects: 0.24, project managers: 0.10, other roles: 0.04)

**Start Date Slip Prediction** A binary classification model predicting if a project will face delay in its start.

*Model Input:* Backlog project $p$, including the expected workload.

*Model Output:* a tuple with a binary variable indicating the expected class and a propensity score on the interval $[0, 1]$: $(p, \text{delay/nodelay}, \text{propensity})$.

**Project Length Prediction** Regression model to predict the length $l$ of project $p$.

*Model Input:* Backlog project $p$, including the predicted start date slip.

*Model Output:* the expected project length in days and a corresponding standard deviation of the predicted value: $(p, \text{expected length}, \text{standard deviation})$

Each predictive model gets discussed in detail in chapter 7, 6, 9 and 10. Chapter 7 discusses the approach to sales classification models, chapter 8 discusses workload prediction, chapter 9 our approach regarding determining a project’s resource mix and chapter 10 discussed predictive models regarding the aspect of time.

### 6.5. Modeling Risks

Forecast quality is influenced by various factors of risk. Risk that is either introduced by the distinct predictive models, the propagation of risks through the machine learning framework, and as well because of varying data quality. A common rule in data-related projects is that the better the data quality is from the basis, the better the outcomes of predictive models will be [36]. The following subsections describe the risks involved in predictive modeling, and the implications of combining various models through a machine learning framework in specific.

#### 6.5.1. Data Quality

From the source data might be messy, inconsistent, not standardized and containing many missing and incorrect values. Sometimes incorrect data might even not be visible by looking at the raw data. Domain analysis shows that process-wise data can be filled in wrong into the system, and people know to bypass the systems (also see chapter ???), or from the front-end system view it might not be possible to enter data correctly. Data quality issues like these have a direct influence on modeling outcomes and are therefore a large risk in data science projects.

Low data quality leads to predictive models that cannot pick up as much correlation from the input features as would be possible in the case of high quality data. This leads to biased and incorrect predictions, low predictive power of the model, and since the variance in the data underlying the model is broader, confidence intervals will be broader.
6.5.2. Propagation of Errors

Regarding a machine learning framework, in which multiple predictive models are chained, confidence in predictions, as well as errors get propagated through the framework. The connection of in and output of multiple models, inherently means the propagation of errors through the models.

For example a false classification of a certain opportunity in the first model, will have its effect on a second model that uses the output of the first model. Suppose the classification of an opportunity \( o \) results in the following tuple: \((o, \text{win}, 0.52)\); with a relatively low propensity an opportunity gets classified as a win. Consequently, data transformation steps will convert opportunity \( o \) in project \( p \) for which during workload prediction the following values are determined: \((p, 90, 5)\); an expected workload of 90 hours with a standard deviation in the prediction of 5, corresponding to a .95 confidence interval of [82, 98]. In reality however, if \( o \) had can been classified as a loss this would have led to 0 hours of work.

This example shows the propagation of errors through the framework. Moreover, error propagation cannot be easily overcome, but it is a factor that can be shown to the end-user by choosing the right method of visualization. Knowing the certainty of a particular prediction for example, upper and lower confidence bounds for the dependent variable can be drawn, showing the uncertainty in the prediction. Also through interactive visualizations, the end user can analyze what the result would be if certain predictions become reality, and hence determine for himself, backed by his domain knowledge, whether certain predictions are reliable or not. Chapter 11 describes confidence intervals in more detail. Chapter ?? discusses visualization methodology in more detail.

6.6. Evaluation

Featuring a model that is composed out of multiple machine learning models, an assessment of model performance must be made on both the individual model components, as well as on the performance of the chained model as a whole. A methodology to do so, is to evaluate the forecast demand for resources for a specific period of time, to what the demand has actually been. This requires the introduction of an evaluation framework in respect to aggregated demand forecast. In this work, we distinguish between two types of demand aggregations:

1. An aggregated forecast on demand from the predicted backlog \( B_{\text{predicted}} \) in respect to a time period \( T \) according to project start dates.
2. A combined aggregated view of expected delivery per time period where all projects in the backlog \( B \) are spread out over time according to their project length.

More details on aggregated forecasts are discussed in chapter 11. Since the concept of evaluation is similar, we discuss evaluation of the first type in this section.

For a given time period \( T \) with a start date \( t_{\text{start}} \) and an end date \( t_{\text{end}} \) we can assign multiple projects \( P_1, P_2, \ldots, P_n \) that start in period \( T \): \((T, \{P_1, P_2, \ldots, P_n\})\). The expected hours for the period \( T \) can now be described as follows.

\[
H_{\text{predicted}} = \sum_{p=1}^{n} H_p
\]  

Similar, \( H_{\text{actual}} \) refers to the sum of hours of projects that has actually been delivered for this time period. Error metrics like the Mean Squared Error as described in section 3.4 can now be used to calculate the error between \( H_{\text{predicted}} \) and \( H_{\text{actual}} \) for period \( T \), referred to as \( E_T \).

\[
E = \sum_{t=1}^{n} E_t
\]

Given a forecast horizon, containing time periods \( T_1, T_2, \ldots, T_n \) the full error on the forecast can be defined as in equation 6.2. Calculation of errors over multiple forecast horizons allows for comparison of model performance over time. Test data would be relatively scarce if limiting ourselves to only the last six months.
for model evaluation. Therefore it is desirable to test the model performance on a larger amount of data from our sample, by back-testing on historical data. A rolling window analysis approach would work here.

Figure 6.2 shows the construction of a set of rolling window period $T_i$ from our sample $T$. Since windows are overlapping, the projects contained in the data are different for each time period. As well projects that are in the training set for the one rolling window, are used in the training set for another window.

Evaluating the model performance for each rolling window forecast, allows to compare the model performance over time. In a traditional approach, we would assume that the modeling coefficients remain constant over time. By using a rolling window approach we allow the model to be trained with the most recent and representative data in respect to the forecast horizon. Examining whether the coefficients are time-invariant allows us to discover if the model is stable over time [42]. Domain analysis shows that robustness is hard but important under a fast changing environment and changing products and services.
To make a complete forecast on the future demand for resources, we must look at work that has already been committed and is residing in the work backlog, as well as to look at future work that is about to result from the sales process. Hence, we must review the current sales pipeline to identify work and determine the possibility that an opportunity will result in a sales win in the near future.

Sales forecasts are an important asset for Microsoft to identify the value of the deals and work that is currently pending in the pipeline. Besides keeping the value of opportunities referred to as 'pipeline value' in the CRM-system, sales managers make forecast on the likelihood of sales to take place on a regular basis. Domain knowledge and data exploratory work yielded the observation of seasonality towards the end of the fiscal periods in June, October, December and April, and as well for holiday periods and the end of the calendar year. Interviews confirmed that this seasonality is the result of organizational stress to reach targets. Similarly the end of the calendar year is the end of the fiscal year for many customers. This seasonality leads to an increase in sales opportunities. Moreover, trends can be discovered for projects with certain characteristics. For example the data insights domain is shown to be growing linearly over the years.

In this chapter we discuss our approach to identify future work that is likely to result from the sales pipeline, and discuss to what extend we can predict future pipeline wins. Section 7.1 defines the problem formally. Then, section 7.2 discusses earlier work at Microsoft. Lastly, section 7.3 describes the incorporation of earlier work.

### 7.1. Problem Formulation

In this problem we are looking at sales opportunities that are relying in the sales pipeline. Since each successful sales opportunity converts into new work to deliver, we wish to predict whether an opportunity has a large likelihood to convert into a success, or not. Moreover we wish to identify factors that contribute to sales wins. In more specific we wish to find an answer to the following questions:

1. How do pipeline insights contribute to a forecast on the demand for resources?
2. To what extend can we predict sales pipeline wins?
3. What are the most influencing factors that determine a pipeline win?

This is a classical classification problem, for which we can define its outcome as in equation 7.1 with a binary outcome.

\[
y = \begin{cases} 
1, & \text{Win - Signed Contract} \\
0, & \text{Loss - Otherwise} 
\end{cases}
\]  

(7.1)

1. Customer Relationship Management System
Dependent on the chosen classification algorithm, the model results in an outcome \( y \). Conceptually, the classifier takes a vector \( x \) of input features, and through model training we determine the optimal weights vector \( w \) fitted on the training data. In the case of a linear classifier, this model is described by the equation below. Other types of linear classifiers can be described as a variant on equation 7.2, whereas tree-based models like decision trees infer their decisions by tree structures.

\[
y = f\left(\sum_{i=1}^{n}w_i x_i\right)
\] (7.2)

In equation 7.2 \( f \) is a threshold function that assigns to all values above the determined threshold the outcome one, and otherwise zero. Its threshold value can be set to either optimize for recall (equation 3.8) or precision (equation 3.7) as evaluation metric for the classifier. From a business perspective we wish to maximize the identification of opportunities that actually become a win. Overestimation of the actual resource need will lead to a surplus in employed resources, while underestimation can be compensated with external resources or through partnering. Hence we wish to maximize the ratio of true positives to all positively classified opportunities. In other words, we prefer precision over recall.

Literature regarding quantitative sales analytics in business-to-business is recent (since 2010) and is limited to publications by IBM on sales pipeline classification work [56, 57]. Furthermore [44] describes the reviewing process of win and loss cases for analysis purposes. Research at IBM [56] describes sales pipeline predictive models as observed in both the business-to-client markets and in business-to-business markets. However, the authors mention predictive models for business-to-business environments to be more challenging due to the relative smaller amount of transactions, often noisy data and a fast-changing market environment. Regarding model implementation, the researchers describe features related to geography, deal size, sectors, new/old clients, lead age, industry and products as correlated to a sales win. Moreover, and significant for the case study at Microsoft, the pattern is observed that the closer an opportunity approaches the end of a fiscal period, the lower the chance of winning. The authors describe logistic regression as the optimal model choice since it can directly produce a probability rather than a hard-to-understand score, as well as for the reasons of cost-effectiveness and understandability. The gain score has been used as the metric for evaluation.

### 7.2. Discussion of Earlier Work at Microsoft

Investigation on earlier work within Microsoft’s organization led to active work on this topic performed by two other teams at Microsoft. Especially one team, as described in chapter 2, performed work that focused on forecasting the sales revenue, aggregated from a project-level. Given the existence of earlier work and the limited time frame, exploring the possibility to leverage on this work was the most logical choice.

The Enterprise Services Business Intelligence team earlier worked on sales win/loss classification as part of their sales revenue forecasting model. In this work a two-fold approach has been taken, first classifying sales opportunities to become either a win or a loss, and secondly by identifying the burn rate of project value. The first part, can be regarded as similar to the sales win/loss classification intended for this work, and thus implied the opportunity for integration in this work. In the remaining part of this section we describe the importance of features that the team found to be influencing the likelihood for an opportunity to become a win. Also we discuss their model performance, and moreover we discuss the integration of results.

#### 7.2.1. Feature Importance

Out of an initial pre-selection of features, a subset of features have been selected that appeared to be of influence on the chance of an opportunity to either become a win or not. Table 7.1 shows the most important features for the sales classification model, with the listed the specific type of feature.

Related to the service offering, a separate predictive model has been built by yet another team at Microsoft and is leveraged as input to the classification model. The model has been trained to assign a probability score of winning, based on product-related features. This score is referred to as the SATT score, assigned by the Services Account Targeting Tool, and is built up by a decision tree model. This decision tree model takes as input 67 various input variables related to the primary product and business scenario of a service, and in this
### Table 7.1: Most important predictor variables for the Sales Win/Loss Classification model.

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer-related</td>
<td>'Industry', 'Segment', 'Sector', 'Account priority'</td>
</tr>
<tr>
<td>Deal-related</td>
<td>'Opportunity Length', 'Forecast Recommendation', 'Deal Size'</td>
</tr>
<tr>
<td>Seasonal</td>
<td>'Quarterly Slip Count', 'Month to close opportunity', 'Fiscal Period Seasonality Flags'</td>
</tr>
<tr>
<td>Service-related</td>
<td>'SATT Score'</td>
</tr>
</tbody>
</table>

way derives business rules to best decide on the propensity of a combination of features to become a win or not.

### 7.2.2. Model Performance

Evaluation of model performance is done by comparing predictions with the actual win or loss cases, and evaluating the classification error by looking at the confusion matrix, comparing between true positives, false positives, true negative and false negatives.

As an evaluation metric, for sales win and loss classification we prefer precision (equation 3.7) over recall, and can optimize for it by applying a threshold function over the prediction outcome of a linear classification outcome as represented through variable $f$ in equation 7.2. In this equation $w_i$ refers to the weight of feature $x_i$. Evaluation is performed on quarterly data from the past. For example a classification is made on sales pipeline data in the second quarter of 2016, where the model has been trained on data before this moment in time, and then evaluated with the actual sales win and losses of the second quarter of 2016. Due to the unavailability of high-quality historical sales pipeline data, only small amounts of training and test data i.e. since 2016 are available.

The geographical area tend to have influence on the overall performance of the model. To optimize the performance, 13 individual different models were build and tested for each area, each with a different threshold function. Speaking in terms of total sales revenue, aggregated over all areas, the model could predict with a bias of 3% to the actuals for the second fiscal quarter of 2016, and with a total bias of 0% to the actuals for the third fiscal quarter of 2016. However, a large variance in error rates over the different areas can be observed ranging from -10% to 15% for the second quarter of fiscal year 2016, and from -110% to 124% for the third quarter of fiscal year 2016. The high accuracy score in terms of total revenue for the total is a result of errors that do cancel out each other: false negatives are compensated by false positives.

Regarding the prediction of pipeline revenue, also known as 'New Work Sold', the following results were obtained. For the second quarter of fiscal year 2016, relative differences between actuals and predictions were within a range of -31% and 26% over the different areas with a aggregated error over the areas of 3% and an error of -19% for the Western European region. The larger part of the variances is caused by the classification of large deals as a false positive or a false negative.

### 7.3. Evaluation and Integration of Work

Pipeline classification is an important predictive element within the demand forecasting framework. Since analysis on historical data shows that only 15% of the pipeline opportunities eventually turn into wins, it is important to predict the opportunities that will actually convert into a win right. Moreover, a correct prediction of an opportunity with a specific service offering serves as input to later models that predict on the dimensions of expected workload and resource mix. From a business perspective, sales classification may also lead to insights on deals that should be more focused on by the business to increase their propensity of winning.

Because of the existence of ongoing work at Microsoft on sales pipeline classification, the most logical choice in our case study was to collaborate with the other team and integrate the results of their sales pipeline classification in our work, rather than reinventing the wheel. Our focus thus relied on how to integrate the sales
classification work within the machine learning framework, and further focus on the predictive models of workload, resource mix and timing.

For each opportunity $o$ as classified in the sales pipeline classification models, an outcome $(o, \text{win/loss, actual/predicted, propensity})$ is returned which can be integrated within the framework as proposed by section 6.4.

Accuracy scores in terms of forecasted revenue for the ESBI classification model are high scoring up to 0% of error. However critically speaking, the variance in the model performance of the sales revenue forecasting indicates a non-optimal accuracy of the sales classification phase of the model. In the use case of ESBI however, this was not necessarily the most important factor for their work since they are looking at aggregated sales revenue, and a false positively classified opportunity could be balanced out against a false negative case. Moreover, the team did not assess their model performance against historical data, which could lead to large enhancements if done in terms of model accuracy since more training data will be available. For future work that extends on this project, we recommend to reconstruct the classification model of the ESBI team to involve more training and testing data. At the moment of writing this is however impossible due to the non-availability of high-quality historical data. Moreover, exploratory work shows that it could be beneficial to adapt the model to train a distinct model for each geographical region within the EMEA\(^2\) time zone and for the Western European area.

\(^2\) Europe, the Middle East and Africa
Workload Prediction

Forecasting the resource hours that will be required to deliver a certain project accounts for the determination of the amount of work involved. On a project-level, a predictive model can be constructed to predict a project’s workload based on delivery requirements from similar projects in the past. Through aggregation of the predictions on a project-level, we later can establish a view on the future demand for e.g. a specific time period, specific products or technical domains. In this chapter we discuss our modeling approach towards predicting the workload on a project-level, and discuss to what extent this figure is predictable.

8.1. Problem Formulation

Predicting the workload for an opportunity or a project consists of analyzing project characteristics, to forecast the work load in absolute resource hours as the unit of measurement. Because of the complexity in hierarchy of engagement, projects and tasks, introducing some formality is required to clearly define the problem. The set of all tasks, all projects and all engagements can be defined as follows.

\[ t \in T, p \in P, e \in E \]  

(8.1)

Since tasks form projects, and projects form engagements, work effort can be defined as follows.

Workload for project \( p \) in engagement \( e \):

\[ h_{ep} = \sum_{i=1}^{n} h_{pi}, \text{ where } h_{pi} \in \mathbb{N} \]  

(8.2)

Workload for engagement \( e \):

\[ h_e = \sum_{p=1}^{n} \sum_{i=1}^{m} h_{epi} \]  

(8.3)

Total workload \( H \):

\[ H = \sum_{i=1}^{n} h_i \]  

(8.4)

Note that since the scope of this research is limited to the granularity of projects, we further disregard tasks and simply refer to the workload for a project \( p \) by a natural number of hours. During domain analysis (see chapter ??) we found that during the different sales stage, multiple quantities of data are available and that similarly multiple process-dependent views can be constructed (see section 5.3).

In this chapter we wish to predict the workload for consultancy project and verify the assumption that data quality becomes better, the further positioned in the services life cycle. More in specific, we would like to answer the following research questions:
1. To what extend can the workload be predicted?

2. Which factors are of most influence to the workload?

3. How does certainty over the sales stages influence the predictive power of the models?

4. Are some service offerings harder to predict than others?

Forecasting workload in absolute hours is a multivariate regression problem. Therefore we can conceptually describe its model by the following equation of a multiple linear regression model, where $h_{ep}$ is the forecast workload for a certain project, $x$ the set of input features to the regression model, $w$ the weight vector for the features in the trained model, and a constant $C$.

$$h_{ep} = \sum_{i=1}^{n} w_i x_i + C \quad (8.5)$$

While building a predictive model to forecast the work load of a certain project, we wish to minimize the error between the predicted workload $\hat{h}$ and the actual workload $h_{ep}$. Section 8.4 further discusses the evaluation methodology and results. Note that the equation 8.5 is conceptual to a regression model and varies for the different types of regression algorithms. The optimal model minimizes the out-of-sample error i.e. the error on previously unseen data.

### 8.2. Feature Selection and Motivation

As described in section 5.3, multiple process-dependent views can be constructed dependent on the data that is available in the various sales stages. Since each successive sales stage contains more variables, with more certainty towards the actuals, it is worth to consider different subsets of features and predictive models. Therefore we have constructed different subsets and models corresponding to each sales stage.

In general, we pursued a feature selection approach based on domain knowledge in the first place. Regarding the project context, the features of ‘business-scenario’, ‘primary product’, ‘capability’ and ‘domain’ are descriptive to the service offering. Given that some type of services take more time to implement than others, the context is relevant to the project size in hours. Moreover to consider are the risk assessment scores as descriptive to the project complexity. The assumption is that a more risk-full project, would probably result in more work.

As well, through application of filter-based feature selection techniques, feature ranks have been identified for features that show the largest correlation with the dependent variable. Many features contain different representations of contracting or labor value of a project. Since these variables are very interdependent, we decided to not provide them all as input to the model. Dependent on the sales stage, the features of ‘Pipeline.Amount.CUS’ and ‘Value.Amount.CUS’ best represent the value of a project, showing a large linear relation to the delivered hours. Exploratory analysis shows (see figure 8.1) that the further in the sales stage, the more correlation can be found between the project value and the delivered hours, but still a lot of variance can be observed. Also there can be observed that many projects get overvalued in the pipeline in respect to the hours that get delivered in the end.

An important factor to note is that contracts might change along the way. For example this happens if project requirements change, or if work takes more time than required. Contract amendments represent these changes in value. Moreover, the reasoning is that periods in the year like ends of financial periods and holiday periods might be of influence to the projects size as these are busy periods and leading to stressful work weeks.

Table 8.1 represent the best-performing subset of features for each sales stage. The contracted value is of large influence on the required hours to deliver a project. For non-billable projects, contracted values are unavailable and need to be estimated based on similar projects.
8.3. Model Training

The definite selection of the right features goes hand in hand with the training of predictive models. Our modeling approach herein was the following. A dataset of 11000 rows has been divided into three subsets following a 60/20/20 distribution. A first subset contains samples for model training, a second for parameter tuning and a third subset serves as a test set to assess model’s performance on former unseen data.

To determine the best performing algorithm for this dataset, we trained and evaluated multiple models by repeatedly training two different models next to each other, with either a different underlying algorithm or with different model parameters set. By intuition, we started with one decision-tree based model, and one regression model. We kept the best performing model, and replaced the second model by another and repeated the process. After multiple iterations, and trying different feature subsets, this eventually led to the best performing model. Figure 8.2 shows a high level overview of our predictive modeling experiment.

Machine learning models like decision forests allow for optimization through different parameter settings. Hyper-parameter tuning allows to find the optimal performing model parameters, and takes away the need to try out the large amount of parameter combinations manually, as earlier described in more detail in section 2.3. Through 10-fold random-sweep cross-validation the optimal model parameters could be found, while optimizing for the Root Mean Squared Error (equation 3.2). This error measure allows us to penalize larger errors.
more than smaller errors. Best parameters differ per model, but for example for the backlog variant and the decision forest regression algorithm, optimal parameters are the following: 15 decision trees, 43 maximum tree depth, 691 random number of splits per node and 7 samples per leaf node. To make sure the model does not over-fit, a 60/20/20 split has been made separating the data set into a training, validation and test set.

8.4. Performance Evaluation

For determining the project size in hours, the definition of a well-performing predictive model is one that predicts the delivered hours as accurately as possible to their actual value. The smaller the generalization error, the better the model’s performance. For this problem we take a look at four evaluation metrics in specific: mean absolute error (equation 3.3), root mean squared error (equation 3.2), relative absolute error (equation 3.4) and r-squared (equation 3.5). The MAE gives insight into to what extend a prediction in hours defers on average from the actuals. RMSE is interesting as large errors are weighted more. The RAE is the approximation error. Lastly, R-squared tells how much of the variance is explained by the model.

In table 8.2 the results are shown of an experiment, testing multiple different regression and/or decision tree algorithms. For backlog projects, this type of problem scores very high values for the performance metrics as can be observed for the decision-tree based models, as well as for linear regression variants. Scores of 0.99 for R-Squared in linear regression based models implies a linear relation and suggests that one can better use a simple calculation with the ‘planned hours’ feature rather than a machine learning model in terms of efficiency. Planned hours however are not completely linear with the delivered hours and in terms of accuracy a machine learning approach proved to perform better. Moreover, linear regression based models predict negative values for some projects. Hence, Decision Forest Regression can be considered as the best performing model for backlog projects, with low mean absolute errors (25 hours), and also a low approximation error.

More interesting is to look at forecasts on the delivered hours at the moment that projects were in the dealing phase (80%). For 80% pipeline predictions we observe up to .74 scores for R-Squared as the optimal algorithm. Performance is however far less accurate than for backlog projects. Neural networks score very low, and likely require a better set up to achieve a good data fit. The well performance of the other models makes this however unnecessary due to the time constraints in this project. Further investigation of the test data shows that
8.5. Findings

Table 8.2: Modeling performance for estimation of the project length.

<table>
<thead>
<tr>
<th>Model</th>
<th>Algorithm</th>
<th>MAE</th>
<th>RMSE</th>
<th>RAE</th>
<th>R-Squared</th>
</tr>
</thead>
<tbody>
<tr>
<td>Early Pipe</td>
<td>Linear Regression</td>
<td>479.97</td>
<td>1247.79</td>
<td>0.89</td>
<td>0.18</td>
</tr>
<tr>
<td></td>
<td>RPart</td>
<td>389.78</td>
<td>1247.31</td>
<td>0.73</td>
<td>0.31</td>
</tr>
<tr>
<td></td>
<td>Random Forests</td>
<td>249.34</td>
<td>961.07</td>
<td>0.46</td>
<td>0.45</td>
</tr>
<tr>
<td>80% Pipe</td>
<td>Bayesian Linear Regression</td>
<td>141.35</td>
<td>561.18</td>
<td>0.37</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>Boosted Decision Tree Regression</td>
<td>219.79</td>
<td>663.07</td>
<td>0.52</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>Decision Forest Regression</td>
<td>137.10</td>
<td>567.17</td>
<td>0.32</td>
<td>0.73</td>
</tr>
<tr>
<td></td>
<td>Linear Regression</td>
<td>219.96</td>
<td>663.19</td>
<td>0.52</td>
<td>0.63</td>
</tr>
<tr>
<td></td>
<td>Neural Network Regression</td>
<td>432.12</td>
<td>1091.02</td>
<td>1.03</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>Poisson Regression</td>
<td>336.11</td>
<td>943.90</td>
<td>0.80</td>
<td>0.25</td>
</tr>
<tr>
<td>Backlog Projects</td>
<td>Bayesian Linear Regression</td>
<td>40.60</td>
<td>126.08</td>
<td>0.10</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>Boosted Decision Tree Regression</td>
<td>45.38</td>
<td>358.84</td>
<td>0.11</td>
<td>0.89</td>
</tr>
<tr>
<td></td>
<td>Decision Forest Regression</td>
<td>24.75</td>
<td>152.59</td>
<td>0.06</td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td>Linear Regression</td>
<td>40.41</td>
<td>126.01</td>
<td>0.10</td>
<td>0.99</td>
</tr>
</tbody>
</table>

metrics alone cannot replace a human judgment. The Bayesian linear regression and linear regression model score very well on average in the metrics, but do in practice predict large absolute errors, large standard deviation and also negative project sizes. Decision Tree Regression models do not have these issues, which make the Boosted Decision Tree Regression also for the optimal model for projects in the dealing phase.

Regarding qualified/early pipeline projects, views for training data have been constructed combining states of projects during their opportunity phase and delivered hours for all projects resulting from an opportunity. In this way contract amendments are covered. Random Forests score the best as a model algorithm for this sales stage, and score amongst other evaluation metrics a .45 score for R-Squared, hence showing that the problem is generally unpredictable with a large share of the variance that cannot be explained by the model.

Figure 8.3 shows an histogram for the errors for subsequently the early pipeline, compass and backlog process phases predictive models. One can observe that the variance in error is very large for early pipeline projects, and very small for projects that are in the last, backlog, phase. This observation confirms the assumption that the forecast becomes more reliable and certain in a later process stage.

8.5. Findings

Regarding the model performance on the test set, multiple conclusions can be made regarding the predictive power of the model. First, as mentioned before and as can be seen in figure 8.3, the quality of results is very dependent on the process stage and on the quality of the data. Earlier stage models cannot explain as much variance through the size of project value. In later phases, the project value feature will be more towards its actual value. This is due to the project value that might change until a project has been fully delivered. Project value is an estimation of the seller until a contract has been signed, and after that moment contracts might still amend.

Large projects i.e. more than 5000 hours or of more than 10000 USD in value, are hard to predict and have a large error in respect to the fitted regression line (see figure 8.4). Smaller projects can be predicted reasonably well, however with many outliers. Moreover, the project size expressed in the hours of work is largely defined by the size of the contract. For latter stage models, up to 80% of the variance in the amount of delivered hours can be explained by the value of the contract size. Not every big-valued project however requires many hours to be delivered as can be seen in the right plot figure 8.4.

Regarding different project types, indicated by a project's 'Primary Product Group', the observation can be made that projects of certain types are harder to predict than others. Figure 8.5 shows the log error by the various product groups. Some groups such as 'On Premise Productivity', 'Security' and 'Dynamics ESS' show many outliers. Further, figure 8.6 shows that the product groups of especially 'Dynamics ESS', 'Applications & OLTP', 'CRM On-premise' and 'Dynamics AX' are relatively hard to predict.

Seasonality appears to have none or minimal influence on the project size i.e. hours needed to deliver a
Figure 8.3: The further in the sales/delivery process, the less variance in the prediction error.

project. Only projects that start in the months of June or December are slightly influenced, which be can prescribed to the busy holiday periods and the end of the fiscal year.

Although forecasting the absolute workload in hours can be accurate in terms of expected values, from a business perspective it is more comprehensible to understand the expected work load as expressed by an interval within a lower and an upper limit. By taking the expected value of a prediction, and the standard deviation of a prediction, a confidence interval can be constructed. The range of the interval is dependent on the confidence interval that we choose. For a normally distributed variable it holds that with 95% certainty can be said that the expected value will lie within two standard deviations from the mean. The same holds for one standard deviation and a confidence level of .68, and for three standard deviations and a confidence level of 0.99 [4]. Hence, the confidence interval can be constructed accordingly. More on the construction of confidence intervals can be found in chapter 11 where we combine all models together into a full forecast on the demand for resources.
8.5. Findings

Figure 8.4: Scatter-plots showing 1) the prediction of delivered hours (scored) versus actuals and 2) a plot of project value vs actual hours.

Figure 8.5: Log prediction errors per Primary Product Group.

Figure 8.6: Prediction errors per Primary Product Group (cut off errors > 80).
Predicting Project Resource Mix

A forecast on the expected workload of a project alone is not sufficient to motivate resourcing or hiring decisions. Hence, insights are lacking regarding the required resource mix i.e. the distribution of workload over the roles. Domain research (see also chapter ??) raised the assumption that typical resource mixes can be found for certain types of projects. In this chapter we describe an approach to identify a project’s resource mix, thereby answering related research questions as defined in section 1.1.2 and 1.2.2 to assess to what extent we can predict the resource mix of a project. In section 9.2 we describe a clustering approach towards the identification of taxonomies for project delivery. Then in section 9.3 we describe the classification of early available service-specific characteristics with the identified clusters.

9.1. Problem Formulation

Consultancy projects typically require resources with different types of roles. Both the over-supply of resources and not having the right people on the right project, leads to under-utilization and thus to the loss of profit. Predicting the resource mix for future projects helps to decide better on the required resources for the delivery of a project. The problem can be formalized as follows.

Recalling from chapter 6, for a set of projects $P_1, P_2, ..., P_n$, each project $P_i$ is delivered by a number of resources $R_1, R_2, ..., R_m$, where each resource has a primary role and technical domain $R \rightarrow (J, D)$. Predicting the resource mix means determining the proportion of the delivered hours $H_p$ for a project $p$, that can be prescribed to a certain role $j$. Formally, we denote the subset of resources with a certain role by the following equation.

$$R_j \subseteq R \tag{9.1}$$

$$S_{p,j} = \sum_{r=1}^{m} \frac{H_{p,r}}{H_p}, \text{ where } r \in R_j \tag{9.2}$$

In equation 9.2, $S_{p,j}$ refers to the share of work delivered by resources with a certain role type $j$ for a project $p$ e.g. ‘Architect’ or ‘Consultant’. Further, $H_{p,r}$ refers to the project work delivered by an individual resource.

$$S_p = \sum_{j=1}^{m} S_{p,j} = 1, \text{ where } S_p \in S \tag{9.3}$$

$S_p$ is the proportion vector that can be defined for a certain project. It defines the resource mix for a project $p$. Moreover, the set $S$ contains the resource mixes for all projects.
In this problem we try to find groups of projects that show similar resource mixes. Characteristic to this problem is the a-priori unknown groups of project mixes, which opts for a clustering approach. Supervised training algorithms like classification are not suitable to be applied for this problem since there is no labeled set of training examples.

Our methodology here is based on similar work by [43], which describes a clustering approach to identify IT-projects with similar resource usages over the project life-cycle. The authors describe an approach to build a project taxonomy that can be linked to project resource requirements, and describe the problem as a sequence clustering problem. Here each sequence contains weekly observations of the resource mixes for a certain project. Then by assessing the clusters with domain experts, project resource mix templates are inferred.

Although we recognize that a project may contain different phases, requiring different resource mixes, for the case study at Microsoft we only regard the total resource mix for a project for the reason of limited time. An analysis of different project phases is an interesting research topic for future work. In summary, we aim for the following objectives with this model.

1. Find clusters of projects having similar resource mixes.
2. Derive a taxonomy of project resource mixes from the clusters found, to serve as a template for future projects.
3. Find a number of early-stage available features / service characteristics that show correlation with the found staffing templates.

By finding correlation between project-related features that are available at an early stage in the services life-cycle, we are able to identify the resource mixes for upcoming projects already at an early moment, which is a valuable observation from a business perspective. The next section discusses the identification of clusters containing projects with similar resource mixes.

### 9.2. A Clustering Approach to Project Taxonomy Identification

For this problem we analyzed the resource mix for 5731 historical projects, delivered within the period covering fiscal year 2014 to fiscal year 2016. A number of processing steps were required as data preparation. Amongst others, we reduced a set of 177 different roles to 34 general role types, and subsequently from this 34 types we selected 21 role types as relevant from a domain perspective. Multiple data transformations have been applied on the dataset in order to construct a proportional view on the delivered hours for each role and project.

Table 9.1 shows the format of a sample of the observations. In addition to the resource mixes, we added a feature regarding the length of a project’s delivery phase in the number of weeks. Through adding this feature, we could achieve a better separation of clusters.

<table>
<thead>
<tr>
<th>Architect</th>
<th>Consultant</th>
<th>Project Manager</th>
<th>...</th>
<th>Software Engineer</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.24</td>
<td>0.36</td>
<td>0.05</td>
<td></td>
<td>0.10</td>
<td>12</td>
</tr>
<tr>
<td>0.40</td>
<td>0.20</td>
<td>0.07</td>
<td></td>
<td>0.00</td>
<td>20</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0.10</td>
<td>0.65</td>
<td>0.00</td>
<td></td>
<td>0.05</td>
<td>14</td>
</tr>
</tbody>
</table>

Table 9.1: Observations of resource mixes for historical projects.

We used k-means as a clustering method, which is a methodology that aims to partition $n$ observations into $k$ clusters where each observation gets assigned to the cluster with the nearest mean. Since k-means is a distance-based algorithm, all features have been normalized to serve as input to the model. While modeling, we evaluated multiple initialization methods and configurations regarding the number of centroids. Clusters found depend on the number of centroids, starting points of the initialization method and evaluation metric.
9.3. Classifying Early Available Features to Taxonomies

The exact model parameters including the number of centroids has been determined in combination with the performance of later classification results in section 9.3.

An initial attempt to include information regarding project’s technical domain to the model did not lead to satisfying results. Since k-means is not well-suited to handle categorical values, we applied a transformation from the categorical ‘domain’ feature to a number of binary features for each category. This transformation helped to further separate clusters. Figure 9.1 shows a figure with the clusters in a principal component graph. In this graph, the values in each cluster are mapped to two component axes. The first axis is the combined set of features that capture the most variance in the model. The second axis combines the features that are orthogonal to the first component. Through this graph, one can see the maximum separation that can be attained between the clusters [7].

![Figure 9.1: Clusters of similar projects found through k-means clustering on project resource mixes.](image)

Figure 9.2 shows a resource mix template for each cluster found. Various runs under multiple model configurations typically led to one cluster with primarily consultants, one cluster around twice the amount of architects than consultants, one cluster with around twice the amount of consultants than architects, a cluster with primarily technology sales professionals, and a number of clusters with a relative large share of subcontractors. Some algorithm configurations led to clusters based on project length, rather than differences in resource mix.

We constructed the templates as follows. From each cluster found, we averaged the resource mixes by roles over all observations. Then we identified the 8 roles that were the most significant, and assigned all other roles to the ‘other’ category. Since for most clusters the ‘other’ role is already included in the data this results in a template including 8 different roles. In the figure mean project lengths are noted for each cluster.

9.3. Classifying Early Available Features to Taxonomies

The more accurate the forecast towards future required roles, the earlier hiring decisions can be made. If early-stage available features can be found that are correlated to specific resource mixes, then an early and reliable forecast can be made on the future resource mix. To identify correlated features, we have evaluated a classification approach on the resource mix taxonomy as identified in section 9.2. Service type-related features, already early available in the sales process are features related to product/service types and business
scenarios. Also expected project value is known. Considering the clusters found in 9.2 we aimed to find correlation between these early available features and the clusters. Our approach herein was as follows.

We applied a splitting operation on our dataset consisting of 5731 projects to form a 70/30 distributed training and test set. Then we evaluated a number of classification algorithms in order to find a model that reduces the generalization error on our data set. To evaluate a model's performance we evaluated the model's predictive power on a the constructed test set with before unseen data. Figure 9.3 shows a high-level overview of the modeling experiment.
For a k-means clustering configuration with six centroids, and a k-means initialization method, we were able to push our classification model towards 100% accuracy. Accuracy however is dependent on the modeling algorithm in the clustering phase. Configurations with more than six centroids led to clusters with a small number of projects, and thus to model over-fitting. Moreover, configurations using a random initialization technique led to less performance than using the default k-means initialization method for the classification model.

These highly accurate results implied that a simple rule-based approach would be possible. Analysis shows that the simplest set of rules can be produced when model accuracy is 100%. However, this is the case when the found resource mixes are not very diverse but rather based on project length. Since we aim to find a diverse set of resource mixes we rather compromise a bit of accuracy for diverse resource mixes. Through the evaluation of the constructed decision trees by the ‘Multi class Decision Forest’ [1] algorithm, we identified a hierarchy of decision rules to identify a project’s resource mix taxonomy. The final model scores 95% on overall accuracy, and 98% on average accuracy. Comparison between actual and predicted classes can be seen in figure 9.4. One can observe that for some cases projects get assigned to cluster 4 instead of 0, and vice versa.
Figure 9.3: A predictive modeling experiment to identify a taxonomy of resource mixes and correlated product features, through a clustering and classification approach.
Figure 9.4: Comparison of actual versus predicted classes for the best-performing taxonomy classification.
Timing is an important factor in business forecasts. The difference of one day can make that a specific project will account under a different fiscal period. Moreover, knowing the expected duration of a project, will help to plan succeeding projects. For this reason it is important to accurately forecast when a project will take place, if it will face any delay, and how much time it will take to deliver. In this chapter we discuss our approach to cover each of these aspects, and discuss to what extend we can predict forecast delay and to what extend we can predict the actual length of a project.

For the dimension of timing we have investigated two aspects. Firstly, this is the identification of the actual project start date. A project start date is called to be 'slipping' when it is delayed with respect to the planned start date. Secondly we wish to determine the project length in days. The motivation for taking the project start date into account first, before estimating the project length is driven by possible seasonal influence. To investigate seasonal influences on delivery we performed a time series analysis on the daily delivered hours, which is discussed in section 10.2

10.1. Definitions

In this chapter we use a notion of timing, which is briefly introduced here. The actual timing of a project and its expected start date are defined by the sum of the planned project start date and its expected start date delay. We define dates as a moment in time, represented by an integer value since the beginning of time:

\[ t \in T \in \mathbb{N} \]  

(10.1)

Then for example a project’s planned start date is some moment in time.

\[ t_{\text{start planned}} \in T \]  

(10.2)

Start date slip and project length are defined as an integer value, representing delay in days (also refer to equation 10.13). These definitions as dates and lengths as integers allow for easy calculation of e.g. expected project start dates or end dates later on.

\[ \text{slip} \in \mathbb{N} \]  

(10.3)

\[ l_{\text{predicted}} \in \mathbb{N} \]  

(10.4)

A further definition of the determination of the start date slip and the expected project length is given later in more detail in section 10.3 and section 10.4.
10.2. Seasonal Influence

Given the subject of workforce planning, intuition tells that the delivery of work over time is influenced by seasonality originating from holidays and fiscal periods. To verify this assumption we took the registration of work hours into account on an aggregated level. Through exploratory data analysis and a time-series decomposition this led to a number of valuable insights as described in more detail in the following sections.

10.2.1. Observations

To get insights into seasonal influences we have performed exploratory data analyses on the time registration data. In this case we had observations stretching over a period of 10 years, from which we determined the last five years as reliable in terms of data quality. Plotting the delivered hours over time showed a significant presence of seasonality on a yearly interval. A large drop in delivered hours is observed in the week of Christmas and the first week of the year. As well a drop in delivered hours can be seen around Eastern. Its size however largely dependents on the region. Furthermore public holidays cause similar drops in the delivered hours in the months of July, August and September. Figure 10.1 shows a plot of the delivered hours for Sweden and Spain.

When zooming in onto a country-level, holiday periods can be observed more precisely. Typical for Northern-European countries are large decreases in delivered hours in July, where Southern-European countries rather celebrate their holidays in August or towards the beginning of September. Other countries, situated more centrally in Europe geographically speaking, like the Netherlands, Ireland or Switzerland do show decreases in work delivery, but these are less significant and more spread over the summer. Remarkable is a decrease in delivery that can be observed for Austria in February. Probably this is due to winter holidays.

Disregarding the weekend, we observe over the week a typical normal distribution with a peak of labor in the middle of the week. This level of detail is however outside the scope of this research.

![Figure 10.1: Seasonality can be observed in the delivered hours over the year for Sweden (left) and Spain (right).](image)

10.2.2. Time Series Analysis

A technique to further analyze our data from a timing aspect is to perform a time series analysis and model on our data, representing the delivered hours over time. As a data set may potentially exhibit a huge variety of patterns, it is useful to try to split up a time series on this data into several components, each representing a category of similar patterns. STL stands for Seasonal and Trend decomposition using Loess, and is a methodology that allows to decompose a time series into a seasonality, a trend and an irregular component [18].

To analyze any seasonality or trends in the data we performed such an analysis on the delivered hours over fiscal year 2012 to 2016. Figure 10.2 shows the outcome with in the top stratum the original time series with delivered hours on a weekly basis for the period. The lower three strata show the decomposition of the time series into the three components. Obviously seasonality is present for the vacation periods and around the period of Christmas and new year. Also a slight upward trend is recognized in terms of delivered hours, confirming the information obtained through stakeholder interviews.
10.2.3. Time Series Forecast

The time series lends itself for a predictive model on the delivered hours for the following fiscal year. We have trained three model variants (‘Seasonal Naive’, ‘ARIMA’ and ‘STL-ETS’) and evaluated their performance against the actual delivered hours. ARIMA [3], autoregressive integrated moving average, is the classical time series model for which we used the seasonal variant. Moreover the STL-ETS (STL - Exponential Time Smoothing) [5, 8] model leverages on the decomposition as described above. Table 10.1 shows the performance evaluation of the three models predicting fiscal year 2016, fitted on data from 2011 to 2015. Evaluation metrics used are the mean absolute scaled error and the mean absolute percentage error, as commonly used to evaluate time series forecasts, representing the average relative difference between the actual and forecasted amount of hours. For the evaluation metrics we aim for a value towards 0 for a well scoring model. Evaluation of the weekly forecast does not lead to very satisfying results as can be seen in table with relative large percentage errors 10.1. Moreover, the relative performance of forecasts on monthly data are better when comparing to a granularity of weeks.

<table>
<thead>
<tr>
<th>Model</th>
<th>MASE</th>
<th>MAPE (weekly)</th>
<th>MAPE (monthly)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arima</td>
<td>0.97</td>
<td>0.47</td>
<td>0.50</td>
</tr>
<tr>
<td>Seasonal Naive</td>
<td>0.94</td>
<td>0.51</td>
<td>0.49</td>
</tr>
<tr>
<td>STL-ETS</td>
<td>1.58</td>
<td>0.58</td>
<td>0.42</td>
</tr>
</tbody>
</table>

Table 10.1: Time Series prediction performance.

Figure 10.3 shows the time series prediction on the delivered hours for the period of fiscal year 2012 to 2016, but extended with a forecast from the ‘snaive’ model for the fiscal year of 2017 indicated by the blue line and confidence intervals of 80% and 95%.
On itself this kind of forecast does only provide insights into trends and seasonality aspects. However, it cannot tell us about work that actually will yield from sales, or about required resources, project characteristics, delay or project length. Therefore a time series forecast is not suited to make a prediction on the demand for resources. It does however provide input regarding seasonality and trends, and thus to busy periods during the year. This is potentially influential to the project length, and the actual start date of a project.

10.3. Start Date Slipping

Knowing the likelihood that a project’s start date is about to be delayed, is important to create a more accurate forecast, and thus to better understand when resources will be required. Ideally, if the delay can be predicted in the exact number of days, this allows to predict the actual start date of a project. In this section we discuss the approach to construct a model for predicting the slipping of a project’s start date and evaluate the performance over multiple model variants.

10.3.1. Problem Formulation

Start date slipping accounts for the amount of days between the first day of project delivery and the initial planned start date. From a business perspective we can call a start date as being ‘slipped’, once a week of time has past since the planned start date. Literature [38] describes multiple causes of project start delay e.g. such as delay as a result of preceding projects that were delayed, or resources with certain capabilities that are not available. Further, much work exist on the topic of project and resource scheduling [30]. No work has however been found on predicting start delay through a machine learning approach in specific. Hence we wish at the first place to study which predictive modeling approach would be the most appropriate for this problem. We define the following objectives for this problem, where we view the problem as not limited by resource constraints.

1. To what extend can we predict a project’s slipping of start date?
2. To what extend can we predict the delay of a project’s start date in days?
3. What is the best performing algorithm for this kind of problem?

Earlier in equation 10.3 we already defined the start date slip as a natural number. Determining the actual start date slip in days is a regression problem, which can be defined as in equation 10.5, where \( x \) refers to the feature vector and \( w \) refers to weight vector in the regression problem.
Initial low-performing predictive modeling results (see section 10.3.6) show the challenge of predicting delay: it is hard to make accurate predictions on a continuous interval. Therefore, a more abstract definition of start date slipping has been adopted, wherein we define start date slipping as a binary problem, where a project’s start can be slipping or not as defined in equation 10.6 building on equation 10.5.

\[
slipped = \begin{cases} 
  s \geq 7: & \text{slipping} \\
  s < 7: & \text{not slipping}
\end{cases} = f(\sum_{i=1}^{n} w_i x_i) \tag{10.6}
\]

Function \(f\) represents a threshold function that divides the regression outcome into the cases of slipping or not, for a defined cut-off value. For our case study we define a project’s start as slipped if the start slip was greater or equal than seven days. This value corresponds to the business understanding of a start date slip, and as well leads to a balanced distribution of training cases for each possible outcome. This balance of cases in the training set is highly desirable for the performance of a classification model [54].

10.3.2. Feature Selection and Engineering

Exploratory data analysis helps to learn more about the data set, and helps to provide a better understanding in which machine learning model will likely work the best on a set of data. For start date slipping we found a data set with a widely spread distribution of values and many outliers. Therefore we considered a number of approaches. In the ideal case, as mentioned above, the start date slipping in absolute days can be predicted. Alternative approaches would be to predict if a project will get delayed using a binary classifier, or via a multi-classification approach making the distinction between none, small, medium or long delay in the project start.

From domain knowledge we know that the slipping of start dates is influenced by seasonality from holidays and fiscal periods. As well this seasonality is dependent by region because of different holiday periods. For this reason we constructed a number of seasonality features. Also our assumption is that issues or delays in the preceding process might be an indicator for delay in the project delivery. For example, because of the way the delivery management process (section ??) is organized, resources requests will not be considered unless contracts have been signed. Another aspect that is likely to be of influence on the project start date slipping is the type of customer. Governmental institutions for example are mentioned to be slow in their decisioning which might be of influence to the project start date as well. Also, the order of a project within an engagement might be an interesting factor. Hence, if a preceding project gets delayed it is likely that the project start of the current project will be slipped if project outcomes are dependent on each other or utilize the same resources.

Moreover, when looking at historical data, we can observe delays that on average are correlated to specific primary products, business scenarios, weekly and monthly periods and customers. For this reason we calculated the average delays for various aspects and used these new metrics as features to generate more model input.

Based on above considerations, we selected a number of features which we assumed relevant based on knowledge from the domain. Table 10.2 shows the features that are used as input to the model. The next sections describe the training of multiple machine learning models and further discuss the feature importance.

10.3.3. A Regression Approach

Predicting the delay in the absolute number of days is a typical regression problem. Using the features from section 10.3.2, we fitted a number of machine learning algorithms on a training set that consisted for 70% of the data. Table 10.3 shows the performance of the different algorithms, based on the evaluation of a test data set, consisting of the remaining 30% of the data set.
### 10. Project Timing Models

#### Table 10.2: Features provided as input for the Start Dates Slipping model.

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seasonal</td>
<td>'Fiscal Month', 'Christmas / New Year', 'Is Holiday'</td>
</tr>
<tr>
<td>Planning-based</td>
<td>'Engagement Manager', 'Days.Since.Engagement.Start', 'Project Planned Start Week', 'Project Start Month'</td>
</tr>
<tr>
<td>Average Values</td>
<td>Avg delay for respectively 'Primary Product', 'Business Scenario', 'Customer', 'Start Week', 'Start Month', 'Engagement Manager'.</td>
</tr>
</tbody>
</table>

#### Table 10.3: Modeling performance for multi-class classification

<table>
<thead>
<tr>
<th>Method</th>
<th>MAE</th>
<th>RMSE</th>
<th>RAE</th>
<th>R-Squared</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boosted Decision Tree Regression</td>
<td>29</td>
<td>63</td>
<td>0.75</td>
<td>0.33</td>
</tr>
<tr>
<td>Decision Forest Regression</td>
<td>31</td>
<td>67</td>
<td>0.81</td>
<td>0.23</td>
</tr>
<tr>
<td>Linear Regression</td>
<td>32</td>
<td>65</td>
<td>0.82</td>
<td>0.28</td>
</tr>
</tbody>
</table>

The low coefficient of determination (equation 3.5) for the constructed models shows that only around 30% of the variance in the data can be explained by the model. Moreover, the relative absolute error shows large error values and looking at the mean absolute we observe a mean error of 30 days in start date slipping. For a model in which we want to predict the exact start of a project these are not satisfying results.

Due to this, the problem has been put more abstract and has been pivoted to a classification problem, thereby reducing the complexity of the problem. The next section discusses a binary classification approach. Then the subsequent section explains a multi-class classification approach.

### 10.3.4. A Binary Classification Approach

Scoping down from a continues scale of \( y \)-values for the regression model, with binary classification a prediction is made if a project will either get delayed or not in its start date. In order to use this methodology, the continuous scale of start date slip in days from the former problem, has been converted into a binary value called 'Is.Slipping'. A project here is marked as 'non-slipping' when delivery has started within a week of the planned start date, and 'slipping' otherwise.

During experimentation we have evaluated multiple binary-classification machine learning algorithms to find the best performing model. Table 10.4 shows the performance of the different models on a dataset with the best performing selection of features. Used evaluation metrics are Area Under the Curve (AUC; equation 10.8), accuracy (equation 10.7), Precision (equation 3.7) and recall (equation 3.8) as represented in the table. The AUC is a threshold-independent measure of how well the model is able to distinguish between two possible outcomes [17]. A good classifier is characterized by a true positive rate that will increase quickly and an Area Under the Curve that will be close to 1. A classifier with an AUC score of 0.5 could rather be seen as guessing, and should better not be considered as a classifier [24].

\[
\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (10.7)
\]

\[
\text{AUC} = \int_0^1 f(x) \, dx \quad (10.8)
\]

Slight variances in performance can be observed with Area-Under-the-Curve scores against .70. Since accuracy, precision and recall values are dependent of the chosen threshold, their representation in table 10.4 area displayed under a threshold of .5 and thus may differ under different threshold values.

A final model with tuned features based on the two-class decision forest algorithm, led to a prediction precision of .75 and the confusion matrix represented by table 10.5. Hereby we chose to optimize for precision, since we want to maximize the quality of our forecast i.e. projects classified as slip will actually slip.
10.3. Start Date Slipping

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
<th>AUC</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two-Class Decision Forest</td>
<td>0.64</td>
<td>0.71</td>
<td>0.62</td>
<td>0.71</td>
</tr>
<tr>
<td>Two-Class Decision Jungle</td>
<td>0.57</td>
<td>0.66</td>
<td>0.53</td>
<td>0.95</td>
</tr>
<tr>
<td>Two-Class Logistic Regression</td>
<td>0.64</td>
<td>0.67</td>
<td>0.62</td>
<td>0.60</td>
</tr>
<tr>
<td>Two-Class Support Vector Machine</td>
<td>0.60</td>
<td>0.65</td>
<td>0.60</td>
<td>0.61</td>
</tr>
</tbody>
</table>

Table 10.4: Modeling performance for binary classification

<table>
<thead>
<tr>
<th></th>
<th>Not-Slipped</th>
<th>Slipped</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not-Slipped</td>
<td>2018</td>
<td>870</td>
</tr>
<tr>
<td>Slipped</td>
<td>1116</td>
<td>2509</td>
</tr>
</tbody>
</table>

Table 10.5: Confusion matrix for binary start date slip classification.

Features for this final model that were observed to contribute the most to the binary classification outcome are 'Days.Since.Engagement.Start', 'Primary Product', 'Capability', 'Business Scenario', 'Region', 'Account', 'Industry', 'Is SMSP', 'Is Public Sector', 'Planned Start Week', 'Fiscal Month', 'Christmas/New Year', 'Is Holiday', 'Engagement Manager', 'Project Manager' and 'Domain'.

Interesting are the validation of the assumptions made in domain analysis that the public sector often causes delay, and that seasonality and specific service groups actually influence the start date of a project. Also the number of days since the engagement start implies the conclusion that a project is more likely to delay if it is not the first project in a customer engagement to start. In other words: a project might be delayed because of delay in another project.

10.3.5. A Multi-Class Classification Approach

Given the improved results from the binary classification model, the assumption was that a multi-class approach might be fruitful as well. Instead of subdividing the linear outcome space into the classes of 'non-slipping' or 'slipping' for the binary approach, here we tried to classify our data in multiple buckets describing ranges of start date slipping. First delivery within a week of the planned project start is again marked down as 'none' slipping. A delay of less than 2 months is described a 'short' delay, and any slippery longer than that is categorized as a 'long delay'.

The following results show the performances that we obtained during our experiments with various machine learning algorithms. For each algorithm the shown values represent the model performance after parameter optimization. Used evaluation metrics are accuracy 10.7, average accuracy 10.9, micro-averaged precision 10.10 and micro-averaged recall 10.11. Micro-averaged based evaluation metrics are useful metrics classes to overcome the issue that classes differ in size.

Averaged Accuracy = \[ \frac{\sum_{i=1}^{n} \text{Accuracy}_i}{n} \] (10.9)

Micro-averaged Precision = \[ \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (TP_i + FP_i)} \] (10.10)

Micro-averaged Recall = \[ \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} (TP_i + FN_i)} \] (10.11)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiclass Decision Forest</td>
<td>0.55</td>
<td>0.70</td>
<td>0.55</td>
<td>0.52</td>
</tr>
<tr>
<td>Multiclass Decision Jungle</td>
<td>0.58</td>
<td>0.72</td>
<td>0.58</td>
<td>0.58</td>
</tr>
<tr>
<td>Multiclass Logistic Regression</td>
<td>0.55</td>
<td>0.70</td>
<td>0.56</td>
<td>0.56</td>
</tr>
</tbody>
</table>

Table 10.6: Modeling performance for multi-class classification
10.3.6. Modeling methodology

As described above, the problem complexity of predicting delay implied an approach to evaluate different levels of abstraction. From regression, we shifted to binary classification, to multi-class classification. Figure 10.4 gives a high-level overview of the modeling experiment comparing a regression, binary classification and a multi-class classification approach.

![Figure 10.4: Simultaneous prototyping and evaluation of Two-Class, Multi-Class and Regression start date slipping model variants.]

10.3.7. Evaluation

Estimating project delay is less predictable than it seems at first sight. An initial regression approach led to disappointing results and a change of approach, shifting towards a classification model. Through a binary classification approach an eventual precision score of .75 could be obtained, while multi-class classification models score precision values between 0.55 and 0.60. Regression models can only explain up to 33% of the variance in the data. Concluding, the best approach to start date slipping is binary classification since it provides the most accuracy and precision, which is the most valuable from a business perspective. The start date slip in days as outcome from the regression model can complementary give an indication of the length of delay.

10.4. Estimating Actual Project Length

An accurate estimation of the actual project length has a significant impact on the total forecast and the resource planning. Given the expected project length of a project, resources can be scheduled more optimal over time, and aggregated forecasts spread over the months can be made with a higher accuracy since we know how much time a project will require.

10.4.1. Problem Formulation

Determining the length of a project is a multivariate regression problem that can conceptually be described by equation 10.13, with feature vector \( x \) and weight vector \( w \) as seen in earlier problems.

Typical for the project length is that the project length is determined at various stages of the services life-cycle: for the first time at the contracting phase, then during project planning, and as well during project
10.4. Estimating Actual Project Length

execution during which the project end date gets regularly adjusted, and thus does the forecasted project length. To differentiate between the different definitions and forecasts of project lengths we introduce the following definitions.

The actual length of a project:
\[ l_{\text{actual}} = (t_{\text{end}} - t_{\text{start}}) \] (10.12)

The predicted length of a project:
\[ l_{\text{predicted}} = \sum_{i=1}^{n} w_i x_i \] (10.13)

The human forecast on the project length (before adjustments):
\[ l_{\text{forecast}} = (t_{\text{end planned}} - t_{\text{start planned}}) \] (10.14)

Both the human forecast of the project length \( l_{\text{forecast}} \) and the predicted project length \( l_{\text{predicted}} \) aim to approximate the actual value as close as possible. Hence, this allows for comparison for the best performing estimation. Logically a successful predictor is defined by its capability to outperform the human forecast.

10.4.2. Feature Selection and Considerations

By domain knowledge the project length is dependent on a number of factors. Among these factors we can differentiate project size, technical requisites, complexity, and seasonality as the main influencing factors on the project length based on our domain knowledge understandings. As well as the availability of resources and customer’s readiness and/or willingness to cooperate are influential on the length of a project.

Especially project characteristics like primary products and business scenarios are factors wherefore data is available. From a customer perspective, information regarding the industry in which a company is operating can serve as model input, as well as information regarding project lengths that were required to deliver work for a specific customer in the past. Seasonality features like holidays and fiscal periods are constructed as for the start date slipping models, as before discussed in section 10.3. Regarding project complexity, output from the risk assessment can be leveraged that are made in the dealing phase of the sales process. This risk assessment is performed through a survey on risk factors and yields a standardized risk score from an aspect of project, engagement and environment. Moreover, planning-based features like the number of resource probably are of influence on the project length.

Interesting for the project length is that the Engagement Manager has to give an estimation of the project length by determining a project’s planned start and end date. A realistic goal is to outperform this human forecast by the model. For projects that are forecasted from the early pipeline however, no planned project lengths are yet available. As an alternative to the planned project length, average project lengths from historical projects can be calculated and leveraged based on similar project characteristics.

Regression is the typical type of machine learning model that is well-suited to face a linear problem like determination of the project length. In the next section, the selection of features is described that is used as model input.

Since there is not much domain knowledge on what determines the project length, filter-based feature selection techniques have been used to identify the features having the largest correlation to the project length.

Through feature engineering multiple additional features have been constructed such as the average contracted hours per resource, and average project lengths for similar historical projects. Also we have constructed features regarding the number of foreign resources, the number of non-European resources and the number of associate, senior or principal resources indicating their experience. These kind of features were yielded by domain research as potentially influential. Seasonality features for the vacation periods are specifically constructed taking into account the vacation periods for each country. Table 10.7 describes the features used as input to the project length model.
10.4.3. Modeling Methodology

Figure 10.5 shows a high-level overview of the predictive modeling experiment to prototype and evaluate diverse machine learning models for the determination of the project length. As this is a regression type of problem, the performance of various regression algorithms have been assessed. Moreover, we have evaluated two types of model forecasts:

1. A model based on the features from table 10.7.
2. And a similar model including the forest from the Engagement Manager.

In this way an assessment can be made to what extend the project length can be predicted without the judgment of the engagement manager.

Parameter tuning techniques moreover have been used to find the optimal model parameters. The data has been splitted into three data sets, according to a 70/15/15 distribution. A part for training, a part for parameter tuning and a part for testing the model performance on unseen data.

10.4.4. Evaluation

Model performance has been evaluated through a number of evaluation metrics. Primarily the Root Mean Squared Error (RMSE), the Mean Absolute Error (MAE), the Mean Squared Error (MSE) and R-Squared. Moreover, we compared the three different forecasts. The baseline forecast made by the Engagement Manager, the model forecast excluding the human (engagement manager) forecast, and a model forecast taking into account the human forecast.

Given by R-Squared, the variance in the project length can be explained for up to 58%. However different performance can be observed for the models for the different regions. For Switzerland for example the model describes 74% of the variance, where this is only 43% for projects in Norway. The forecast of the engagement manager as feature to the model increases the forecast accuracy with 10%. Most ideally, this forecast is however not taken into account as it is a forecast that is adjusted continuously during project delivery.

Below evaluation metrics are shown for the model performance, excluding the Engagement Manager’s forecast, and including the Engagement Manager’s forecast. One can observe that for both models the initial forecast of the engagement manager gets outperformed. Random forests as regression algorithm gives the best performance, and shows no further significant improvements after 50 model iterations.
10.4. Estimating Actual Project Length

Figure 10.5: Prototyping experiment in Azure ML Studio to model the project length.
In chapter 6 we introduced a machine learning framework to forecast the demand for resources. Then, in chapter 7, 6, 9 and 10 we described each set of predictive models in detail, and discussed how the models are used to make predictions on an individual project-level. In this chapter we combine the modeling efforts from the last chapters and show how an aggregated forecast is constructed to create a future view on the resource demand. Having an aggregated forecast allows us to evaluate to what extent the actual demand for resources is predictable, and to identify practical model limitations.

In the following sections we introduce two types of demand forecasts.

1. An aggregated forecast on demand from the predicted backlog $B_{predicted}$ in respect to a time period $T$, aligned towards project start dates.

2. A view on expected delivery from the backlog $B$ per time period, combining all introduced predictive models in this work i.e. sales pipeline classification, workload prediction, resource mix and timing models.

The first forecast provides a periodical view on work that is coming, expressed through expected outcomes for each model and confidence intervals. It provides a periodical view on expected work and its likelihood. Stakeholders can leverage the forecast primarily for the capacity planning as the forecast helps to understand the amount of work that is expected per month. Being aware of this, may help to better spread the demand for resources, by setting more viable project start dates or by taking hiring decisions. It also helps to understand which demand and projects are still uncertain, and show which opportunities should be focused on to become a win or for example to mitigate the chance of delay.

In the second type of forecast, the results of each predictive modeling component in this work are combined. In this way a weekly forecast on the expected resource need is created, under the assumption that all predictions become true i.e. sales pipeline classification, workload, resource mix, project length and start date slip. This type of forecast shows the ‘real’ demand for a given period in the future, and helps to understand how the workload would look like if no further actions are taken from a project and resource scheduling perspective.

### 11.1. A Periodical View on Future Resource Demand

For a given time period $T$ with a start date $t_{start}$ and an end date $t_{end}$ we can assign multiple projects $P_1, P_2, .. P_n$ that start in period $T$: $(T, \{P_1, P_2, .., P_n\})$. The expected hours for the period $T$ can now be described as follows, with by $h_p$ noted the expected hours for a project.

$$H_T = \sum_{p=1}^{n} h_p$$  \hspace{1cm} (11.1)
The expected hours can be made more specific by summing only the hours for certain roles, countries or specific product types for example. Similar summations can be performed for different types of granulation.

### 11.1.1. Confidence Intervals

Unmotivated forecasts are hardly useful for decision supportive practices. Assigning an expression of confidence regarding a forecast made helps to provide deeper insight into the confidence of a certain prediction. Hence, for each project and expected resource demand we construct confidence intervals as an expression of confidence and risk, calculated as follows.

\[
CI = \bar{x} \pm z_{a/2} \frac{\delta}{\sqrt{n}}, \text{ where } CI \in [0, +\infty) \tag{11.2}
\]

In equation 11.2, CI represents the confidence interval with lower and upper confidence boundaries, \( \bar{x} \) refers to the expected value, \( z_a \) the critical value corresponding to a certain confidence level \( a \). Moreover, \( \delta \) to the standard deviation of the sample and \( n \) to the sample size. Since the workload is expressed in hours, a lower bound on the confidence interval can be set as zero as hours cannot be negative.

Besides a confidence interval on the expected workload, a similar calculation for the confidence interval can be applied on the project length. Communication of the project length through an expected interval gives a more clear view on the prediction certainty towards the end-user.

### 11.1.2. Determination of Role-specific Workload

Having both a prediction on the expected workload, and as a well a prediction towards the expected resource mix, the two can be combined into the expected workload per role. Since the resource mix is described by proportions, we multiply the proportion vector \( r \) of representing the distribution of resources roles with the total expected hours.

Given the resource mix distribution per role \( r_i \in [0,1] \) and \( \sum_{i=1}^{n} r_i = 1 \), and the expected hours for a project \( H \), the absolute expected hours per role can be computed as follows.

\[
H_r = r_i \ast H, \text{ where } \sum_{r=1}^{n} H_r = H \tag{11.3}
\]

A further specification regarding confidence intervals can be made analogously for roles, for each boundary of the confidence interval.

### 11.2. A Three-Dimensional Aggregated Forecast

A combination of all the predictive models as proposed in this work within the proposed machine learning framework, provides a complete view on the future demand for resources. Through the combination of the three dimensions as introduced in chapter 6, and the corresponding predictive modeling outcomes of expected start dates, project length, workload and project length, the resource need can be analyzed over time. A number of data transformations are required to combine model outcomes and calculate expected project start and end dates. Continuing on the definitions in section 10.1 we can determine expected project start and end dates as follows.

\[
t_{\text{start expected}} = t_{\text{start planned}} + \text{slip} \tag{11.4}
\]

\[
t_{\text{end expected}} = t_{\text{start expected}} + l_{\text{predicted}} \tag{11.5}
\]

For a project \( P \) with predicted length \( l_{\text{predicted}} \), workload \( H \), and resource mix \( S \) we calculate the workload \( H_t \) per time period \( t \) in days as follows. The vector \( H_{t,J} \) describing the workload per role \( j \) is defined by equation 11.7.
\[ H_i = \frac{H}{H_{\text{predicted}}} \cdot t \]  

\[ H_{t,j} = \frac{H}{H_{\text{predicted}}} \cdot t \cdot S_{p,j} \]  

Correspondingly from the detail level of days, workload can be summed up to weeks, months, years or any period of time. In chapter ?? we discuss interactive visualizations and dashboards that allow users to set the level of aggregation themselves and see the results in an interactive way.

Since we combine multiple models with different levels of uncertainty, errors can propagate from the one model to the other. We earlier described this in section 6.5.2. In order to better understand the reliability of the aggregated forecast under error propagation, we performed an analysis on how errors behave through the model in practice as further discussed in the next chapter.

### 11.3. Forecast Limitations

When connecting the in and outputs of multiple predictive models, errors will propagate through the models. In some situations an error produced in the one model will lead to an accumulation of errors in further models. In other cases, the one error will cancel out the other error. Especially for aggregated forecasts, the latter described case is desirable since significant errors on the first sight, may not have a large impact on the aggregated forecast.

For resource demand forecasting, we found that the errors do cancel out each other. The one project that gets predicted as a win wrongly, gets balanced by a project that will actually be predicted as a loss but is a win. An important share of the demand that is not covered by the models, is defined by contract extensions that are arranged by the consultants themselves. It happens that consultants arrange their own work through up-selling at location with the customers. This additional demand is not, or only very late visible in the sales pipeline which causes that it is less likely to be included in the forecast.

Furthermore, regarding timing aspect holds the following. The further in time, the larger the ratio of qualified pipeline work. Also the later the forecast in time, the less accurate it is in respect to the actuals. Hence work that will fall in a certain month, but that is not yet known at the time of the forecast as a sales opportunity, will not be covered in the prediction. Therefore from a business perspective, it is very important to maintain a good administration of sales opportunities, also referred to as pipeline quality.
Due to the confidentiality of the information represented in this chapter, the content of this chapter has been removed from the public version of this work.
Conclusion

In this work we have proposed a machine learning framework to forecast the demand for resources in the environment of a consultancy organization. Specific characteristics for such an organization are the presence of a sales stage, a work backlog, a pool of resources, a planning and a delivery phase. Through domain research and after having conducted a significant number of interviews, we found the three dimensions of expected workload, resource mix and timing as important aspects of the resource need. Accordingly, we proposed a decomposition of the modeling approach into these three dimensions, thereby reducing the complexity of the modeling effort.

Domain analysis during a case study at Microsoft showed that current resource planning and capacity planning practices are only minimally supported by reports or BI tools and are largely based on experience. In short, any insight in the future demand for resources could already help to make the services life-cycle more insightful. For the capacity planning process in specific, insights regarding upcoming types of projects and the expected resource mixes were found to be useful. In this way an aggregated forecast can be made on the required resources, specific to roles and technical domains. For the resource planning process, insights into project delivery are rather useful. Knowing for example when a specific resource will be needed, or being aware that a project is likely to face delay is considered useful information. Utility of the research from an organization point of view is thus not solely provided through the predictive elements, but also for a large part by business intelligence applications such as dashboards and visualizations.

A large effort in this project has been devoted to the combination of data sources available at the various stages in the process, to reconstruct the process from initial sales lead to delivery for each project. The particular situation at Microsoft was such, that these data sources had never been combined before. Due to missing common database logic and shared key columns among the different data sources, the quantitatively combination of data appeared to be difficult. Due to low data quality, not the whole trajectory from sales to delivery could be reconstructed for each historical project, resulting in less data that could be used for the training of the predictive components. To deal with this fact, we were able to design our model as such that the reduced possibility to combine data did not influence the completeness of the forecast. In short, we have built forecasting models for the different project stages and corresponding data sources, reducing dependency on the proper combination of data sources.

Predictive models have been proposed for each of the tree dimensions that contribute to the resource need i.e. workload, resource mix and timing. One or more predictive models have been proposed for each dimension, taken together forming a forecast on the future demand. A first model is a classifier that identifies upcoming sales opportunities to become either a win or a loss. A second set of models predicts the total hours and the resource mix per project in the work backlog. Lastly, models on the aspect of timing predict when a project is expected to take place. In specific a prediction can be made on a project’s expected duration and if a project will face delay in its start date. We found that the further a project is in the process from sales to delivery, the more data is available, the better the quality of the data is and also the more accurate predictions are. This observation motivated the construction of a number of variants on the models in correspondence with the varying data quantities. In this way we leverage the predictive performance for projects for which we have
more data available, and which we are more certain about.

Through exploratory data analysis and time series analyses on the aggregated delivered hours over the last years, we discovered trends and seasonality in the data, that are different for the sales phase and the delivery phase. Sales numbers showed to be dependent on the moment in the fiscal year, whether delivery is rather influenced by holiday periods.

The extend to which the demand for resources is predictable, is defined by the predictive power of each of the individual components. Sales classification showed to be a problem that can be predicted through a classification approach. Evaluation shows that predictions are reasonably predictable with .74 precision and .73 accuracy. Also it is known that the effect of classification errors do only have a small impact on aggregated forecasts since false positive and false negative cases do cancel out each other.

Workload in hours showed to be a dimension that is generally well-predictable. The total amount of delivered hours shows to be mainly influenced by project values. This means that the predictive power of models is largely dependent on the subjective estimations of stakeholders along the process. Due to its subjective nature, early estimations of project value show very low correlation with the final delivered amount of work. Generally the further a project is situated in the services life-cycle, the better the accuracy of the workload predictions. Regression models on workload for qualified sales projects, contracted projects and planned projects, relatively score .45, .74 and .98 in terms of R-Squared.

A clustering approach has been used to identify clusters of a-priori unknown groups of similar project resource mixes. A project taxonomy could be identified of similar projects based on the found clusters. For each of the six found clusters of resource mixes, we subsequently derived a staffing template characteristic for that project based on the cluster's mean resource mix. A classification approach showed that the found taxonomy of resource mix distributions is strongly correlated to service offering-related features and is therefore already identifiable in an early sales stage with .96 accuracy scores for identification of the right resource mix.

The predictability of a project's length showed to be dependent on a combination of product-related features and prior estimates by stakeholders. Through replacements of missing project lengths with average project lengths of similar projects, accuracy scores can be pushed towards 0.7. This outperforms the human forecast on the project length by 48% in terms of RMSE.

Prediction of delay in project start date proved to be a difficult problem to solve in terms of absolute number of days. With merely .35 of the variance explained by the model, a pivot to a classification approach has been pursued. In this way, projects that will face delay in start date can be predicted with .7 precision. Domain analysis provided that also low-confidence predictions already lead to useful business insights.

A machine learning framework and corresponding data pipeline has been proposed, that combines the individual modeling efforts into an aggregated solution. Through a sequence of data preparation steps and machine learning models, a chain of models is created in which the output of the one model is leveraged as input to another. In this way, the outcome of the win/loss classification on sales opportunities is leveraged to forecast the full work backlog for the forecasted period. Then the full backlog is used as input for the workload and resource mix forecasting models, and subsequently this result is used as input for the timing models. Finally, the output of the timing models is used to create a forecast on a chosen level of granularity e.g. weeks, months or years.

Besides the testing of individual model performance for the separate sub models, we constructed an evaluation framework to assess the performance of the chained model in respect to the actuals. In order to test with more data, and to assess the model's robustness to change over time, we propose to use rolling window forecasts that take as input a three-year period of training data, to forecast the year that is subsequent to this period.

Next to the evaluation of quantitative work, qualitative work like the effectiveness of the provided insights has been validated with stakeholders in the business. Dashboards and visualizations have been composed in collaboration with resource planners and capacity planners to maximize utility and the value delivered. Management dashboards were proved to be useful and have been provided for multiple contexts, amongst 1) an aggregated view on expected workload per time period, specified towards roles and technical domains 2) a combined view where actual and predicted work is combined, including the spread of a project's work-
load over time, into a 'full-dimensional' forecast 3) insights on a project-level into historical project delivery trajectories 4) insight into future resource needs on a project-level.

A notion of risk has been integrated into the forecast in two ways. Primarily since predictive models are constructed for each sales stage, we can assign a risk bucket to each prediction on a project-level, indicating its sales stage. This provides insights into risks, since we found that the later a project is in the life-cycle the more certainty is involved. Moreover for regression-based predictions, .95 confidence intervals specify a notion of certainty in the prediction. As well for classification models a propensity score is provided with the assigned class label.

Concluding, a machine learning framework has been proposed to forecast the demand for resources for service-delivering organizations. Visualizations provide insights that are proven to be useful in a business-context. Predictive power of the models showed to be varying, and highly dependent on the problem domain and the quality of the underlying data. This led to varying results and different levels of satisfaction. Hence, we do recommend to use the provided insights in the capacity and resource planning processes, but only in a decision-supportive way. Combined with the knowledge of capacity and resource planners, more motivated decisions can be made supported by the delivered insights.

The scope of this research has been relatively broad, providing a baseline solution for future work on resource demand forecasting. Especially we see opportunities for future work, to dive deeply into each sub model. More recommendations on future work are discussed in the next chapter.
14

Recommendations for Future Work

This work serves as a proof-of-concept for decision-supportive predictive analytics applications within the Consulting Services organization of Microsoft. Concluding, from a business perspective the most value for the current business can be obtained through the information that is obtained through data visualization i.e. business intelligence rather than from machine learning applications.

With these final words we want to indicate areas of improvement and opportunities for the current models, indicate where the most business value can be obtained and also recommend the changes that are required to simplify and improve the quality of future data science initiatives within the organization. As well we want to give recommendations how to build further on this work in the future. Recommendations are presented for a short, mid and long term horizon, and subsequently discussed in the below sections.

14.1. Short Term

On a short term, the most value can be obtained through visualization and data-driven insights. Current reports are static and do not provide the user to click through. Interactive reports allow to do so, and hence allow for root cause analyses regarding risks, and to identify issues and bottlenecks with the current data and business.

Regarding the machine learning and forecasting part the main technical focus should lie on productionizing the prototype - automating data processing steps and automatically rerunning models. Furthermore, attention for data collection and referencing between the systems is a key to enhance the base data quality. Especially we recommend to focus on the points below.

- **Extend business intelligence capabilities** on the services life cycle through visualizations and interactive dashboards on the delivery process, but also on the sales process.

- **Productionize the predictive models** through deployment into the cloud, and the periodical retrieval of data and predictive models through automated data pipelines.

- **Direct consumption of sales classification model outcome** from the ESBI team. Currently outcome is consumed through csv files. Most ideally the running of models is synchronized.

- **Enhance data source coupling** by validating that references are made between the different systems; that projects refer to the right sales opportunity; and that opportunity-id’s are filled in without errors.

- **Include subcontractor analytics** to show insights into vendors that get contracted the most.

- **Attention for data quality** on multiple levels: from the source, process-wise and culture wise.

- **Prefer a data-driven culture** above a target-driven culture. Or put targets on the correct registration of data.
14.2. Mid Term

Recommendations for the mid term horizon mainly rely on improving sub models to enhance model accuracy, as well as on the up-scaling of the model scope to a wider area. Furthermore, structural improvements of data collection practices is expected to lead into further improvements of data and thus model accuracy.

- **Further specify sales-classification** by involving more training data and incorporating different seasonality over the Western European countries. Exploratory work and other predictive models show that it could be beneficial to adapt the model to train a distinct model for each geographical region within the EMEA time zone.

- **Scale up** the model to make forecasts for the full EMEA or Worldwide region. Account for the different seasonal influences by the regions.

- **Enhance project length models** by incorporating factors like availability of resources, seniority levels and concurrent projects.

- **Enhance start date slip models** by incorporating the factors of concurrent and preceding projects, dependent resources and likelihood of delay of preceding projects.

- **Further specify resource mix models** to cover multiple project phases; often a project can be subdivided into multiple phases, requiring a different mix of resources.

- **Validate input of data** to be of the right data format e.g. format checking of an opportunity-id, date format checking, required field checking, checking that booked hours do not exceed contracted hours.

- **Reconsider risk assessments** to be made by resources that are experienced and capable in the area of the deal. Current risk assessment scores seem not to significantly influence the project length.

- **Redesign data collection processes** by adapting systems and collection processes to what you want to collect in the future, rather than to analyse the data that is available now. Many processes are currently described through unstructured text documents and spreadsheets (early staffing requests, proposals, etc). Very limited information is available in the current systems regarding project complexity or detailed work descriptions for the engagements. As well capture both success cases and unsuccessful cases to allow development of unbiased and less skewed machine learning models.

- **Assign a data officer** to be responsible for the data quality within the organization. Someone that is concerned with the overall quality of the data, problems with systems, and sets direction to new collection-practices.

14.3. Long Term

To allow for model robustness, develop a practice to retrain predictive models on the newest data in correspondence with organizational changes with a minimal effort. Moreover opportunities exist to create smart workforce planning tools incorporating the outcomes of the predictive models e.g. proposing the most optimal project planning, the most optimal resource planning, or recommending deals to focus on in order to optimize resource utilization.

- **Complement pipeline opportunities** with predicted sales opportunities by forecasting leads through identifying similar customers and product/service adoptions.

- **Leverage model outcomes in smart workforce planning tools** to provide e.g. machine-learning recommended project scheduling, resource scheduling or to recommend deals to focus on to create the best fit with available resources.

- **Scale out** the model to make forecasts for the full Services department, including Premier Support besides Consulting Services.

- **Redesign information systems from scratch** to allow for the optimal collection of data, while keeping in mind objectives for future predictive models. Reduce the use of free text fields in data collection

---

1 Europe, the Middle East and Africa
processes and aim for consistency in business logic over the different systems. Capture all data updates, rather than overwriting the original date entry, to allow for better practices in modeling over time.


[57] Junchi Yan, Chao Zhang, Hongyuan Zha, Min Gong, Changhua Sun, Jin Huang, Stephen Chu, and Xiaokang Yang. On machine learning towards predictive sales pipeline analytics. 2015.