A Monitoring System for Machine Learning Models in a Large-Scale Context

Version of August 21, 2020

MyeongJung Park

A Monitoring System for Machine Learning Models in a Large-Scale Context

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

MyeongJung Park
born in Gumi, South Korea

Software Engineering Research Group
Department of Software Technology
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
www.ewi.tudelft.nl

ING Bank Personeel B.V.
Acanthus, Bijmerdreef 24
Amsterdam, the Netherlands
www.ing.nl

A Monitoring System for Machine Learning Models in a Large-Scale Context

Author: MyeongJung Park
Student id: 4750705
Email: M.Park@student.tudelft.nl

Abstract

Since building a machine learning model costs a lot while following 9 stages, the automated machine learning model creation became a crucial role in a large-scale context. At the same time, a monitoring system became an essential factor for machine learning models.

This thesis presents the monitoring system for machine learning models at ING in an enterprise context with new features required by users. Moreover, the thesis describes a case study of ING, a large global banking company that develops software solutions in-house. We conducted a mixed-methods study, consisting of data collection of the monitoring system and a survey with the users of the monitoring system. Our research shows that challenges found by the actual users of the monitoring system and mapped challenges discovered by the Microsoft study are related to machine learning model monitoring, the perception of the users on the importance of the monitoring system, and the impact of the monitoring system.

We found that the monitoring system at ING supports relatively efficient model management in terms of checking model validation and evaluation. Moreover, the users of the monitoring system perceived that it is an important system, and it supports the models regarding quality, the trust of the automated model creation, and usability. Additionally, compared to the existing solution, the monitoring system at ING supports useful model management.

Thesis Committee:

Chair:	Prof. Dr. A. van Deursen, Faculty EEMCS, TU Delft
University supervisor:	Prof. Dr. A. van Deursen, Faculty EEMCS, TU Delft
Company supervisor:	Dr. Hennie Huijgens, ING
Committee Member:	Prof. Dr. Georgios Gousios, Faculty EEMCS, TU Delft
	Dr. Asterios Katsivodinos, Faculty EEMCS, TU Delft

Preface

My master's degree journey has been a fantastic experience which I would never forget. I believe that I could not have been achieved without support from my family, my friends, my advisors and my colleagues who played an important role in my life.

I want to thank my supervisor, *Prof. Arie van Deursen* for providing me with the right direction for my thesis, guidance on numerous occasions and constant support at all times.

I would like to express my deepest gratitude to my manager and advisor, *Hennie Huijgens* for allowing me to be a part of the software analysis team at ING, for continually guiding me and motivating me.

My master thesis journey became easier through their constant help and support whenever I needed it. I would also like to thank the entire *1-to-1 Analytics team at ING* for helping out with brainstorming discussions for this project and giving me the right direction of the project during my internship.

MyeongJung Park
Delft, the Netherlands
August 21, 2020

Contents

List of Figures

Chapter 1

Introduction

As people's interests and requirements are heterogeneous, it is imperative for business today to understand their customers, for instance, to predict the customers' future behavior and needs in terms of satisfaction[13]. To comprehend the clients, today businesses leverage big data technologies and approaches to gain unique and differentiating insights on their clients. Also, they treat customers' data by building a machine learning predictive model for forecasting the future behavior of customers. Such a model is created by different people, for instance, a researcher, data scientist, or a business analyst. Their tasks include giving an example, formatting a correct input dataset, choosing the appropriate model/algorithm, and picking a software package that is used to train models[50].

Building a machine learning model is a trial-and-error based repetitive process in an enterprise context[54]. A model is built based on a hypothesis, and then the model is trained and tested. After this, the results redefined the hypothesis, and then the refined hypothesis revised the model. Amershi et al.[3] describe a machine learning workflow shown in Figure 1.1. Amershi et al. illustrate that the machine learning workflow is composed of 9 stages:

1. Model Requirements
2. Data Collection
3. Data cleaning
4. Data labeling
5. Feature Engineering
6. Model Training
7. Model Evaluation
8. Model Deployment
9. Model Monitoring

In this thesis, the final stage, model monitoring, is the key topic. In the machine learning workflow, there are many feedback loops in some stages, such as Feature Engineering, Model Evaluation, and Model Monitoring. The thick arrows in Figure 1.1, represent that this stage can loop back in any other previous stage. The thin arrow denotes that the model training stage can loop back to the feature engineering stage. Consequently, when people want to create a machine learning model, they have to go through all the machine learning workflow stages manually.

While many enterprises are now actively employing machine learning solutions and in-

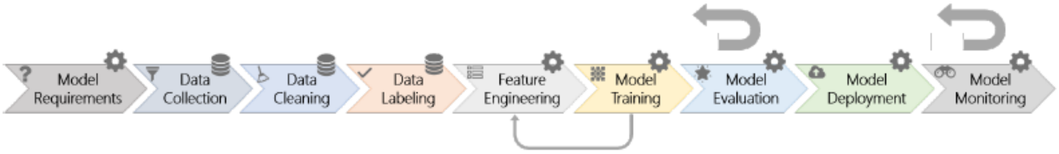


Figure 1.1: The nine stages of the machine learning workflow[3].

corporating machine learning models in a business setting, it brings along a set of various challenges. These challenges regard to managing the models' life-cycle from model initiation to model deployment, and also collaboration working across the several business teams and tools[3, 51].

Amershi et al.[3] describe eleven challenges, regarding building large-scale ML applications and platforms:

1. Data Availability, Collection, Cleaning, and Management
2. Education and Training
3. Hardware Resources
4. End-to-end pipeline support
5. Collaboration and working culture
6. Specification
7. Integrating AI into larger systems
8. Guidance and Mentoring
9. AI Tools
10. Scale
11. Model Evolution, Evaluation, and Deployment

Also, Spooner and John demonstrate[51] three difficulties of building the framework for modeling, namely, Inconsistent Data in the Wrong Format, Cross-Functional Involvement, and Different Technologies Being Used. Plus, Schelter et al.[44] illustrate three challenge categories: Conceptual challenges, Data management challenges, and Engineering challenges. Within the three groups, there are nine challenges on machine learning model management:

1. Machine Learning Model Definition
2. Model Validation
3. Adversarial Settings
4. Decision on Model Retraining
5. Lack of a Declarative Abstraction for the hole Machine Learning Pipeline
6. Querying model metadata
7. Multi-language Code Bases
8. Heterogeneous Skill Level of Users
9. Backwards Compatibility of Trained Models

Some of the critical challenges mentioned in papers[51, 3], are data availability, collection, cleaning, and management. Many machine learning techniques deal with large tables that include different data used by different roles. Therefore, machine learning projects rely on data availability, quality, and management[40]. Also, the performance of models relies on trained and tested data. The performance of the models is getting worse due to the changes in data. In order to prevent the degradation of model performance, we need to track the data and model performance to manage the model properly[54].

Another difficulty is to ensure the validation[44], evolution, evaluation, and deployment of the models in a production environment and across the lifetime of the models[3]. This particular problem is even more prominent in an enterprise where different roles engage with the model's process at a large scale because users have different backgrounds[44]. Therefore, a lot of manual work may lead to producing errors. Additionally, re-training the model every day is a hassle and costs a lot, and also users may not have enough time to do this.

An automated model building framework is critical to address these challenges, remove all hassles, and accelerate the model creation process. Plus, since the machine learning parts are grown and combined into the large scale system, the automation framework needs to cover all different nine stages of machine learning workflow described in Figure 1.1, and it needs to be aligned with the day-to-day workflow of software engineers[3].

1.1 Motivation

Achieving a fully automated machine learning framework is challenging due to the inherent uncertainty of data-driven learning algorithms[36] and hidden feedback loops, which can occur due to the component entanglements[46]. Furthermore, as the amount of data is increasing exponentially, it is also getting more challenging to scale up systems in order to deal with abundant data[12]. Besides, even though all the machine learning workflow is fully automated, the automation framework still has to solve several issues such as trust, engagement, and the fear of automation[20, 29].

Since a monitoring system aims at keeping a model in a valid status, such a monitoring system can address these challenges by tracking the model creation process and checking model validation and evaluation metrics[51]. Therefore, in this thesis, we seek to know the importance and the impact of the monitoring system in terms of addressing these challenges.

In general, models are evaluated by ROC (Receiver operating characteristic), AUC (Area under the ROC curve), and Lift charts[56]. ROC and AUC are model performance measurements for classification models used in machine learning at various threshold settings. They indicate how models can distinguish between classes, and measure predictive quality. The Lift chart is another way of visualizing the performance of a classification model. It is a measurement of the effectiveness of a predictive model calculated as the ratio between the results obtained with and without the predictive model (model cumulative sum / random guess) [56].

1.2 ING

ING is a large multinational financial organization with about 54,000 employees and over 38 million customers in more than 40 countries[23]. In 2011, the bank introduced DevOps teams to get developers and operators to collaborate in a more streamlined manner. Currently, ING has 611 globally distributed DevOps teams that work on various internal and external applications written in Java, JavaScript, Python, C, and C[26]. The bank is in the midst of a technology shift from a pure finance-oriented to an engineering-driven company. As ING is a large corporation, there are many different roles, such as data analyst, data engineer, and software developer, who in data driven projects work together to meet stakeholder needs.

1.3 Objectives

Having valid models is critical in the decision-making process in an enterprise because the enterprise needs to predict future preferences and behavior of the users. Building effective prediction models requires high-quality model evaluation and validation, and addressing the needs of each requirement and role. Therefore, model monitoring plays a critical role throughout the model creation process. Thus, it must be supported by a monitoring system. Different users will require various features in a monitoring system.

So, we focus on addressing all requirements that can give a better model diagnosis to satisfy the different users. Based on addressing these requirements, we add new features in a monitoring system that deals with model lifetime management. Moreover, in order to explore the impact of a monitoring system, we examine and test the monitoring system within ING as a large-scale context. Additionally, we compare to the existing solutions from open-source systems such as MLflow[35], Polyaxon[39], Prometheus[41], which cover most major deep learning frameworks and give visualization features.

Our exploration resolves the following research questions:

RQ1: *"To what extent can a monitoring system support stakeholders in terms of maintaining machine learning models?"*

The main reason for having a monitoring system is that it facilitates managing and maintaining the models. In order to see the impact of the monitoring system, we collect the data in a quantitative way and perform a survey among stakeholders involved in machine learning model development, maintenance, and use.

RQ2: *"How do users of a monitoring system perceive the way it supports the models with regard to usability, quality, and trust?"*

A monitoring system needs to be utilized by users, and therefore it is necessary to see how the users perceive the monitoring system. Hence, we would like to see the effects of the monitoring system in perspectives such as quality, usability, and trust by surveying the actual users of the monitoring system at ING. Additionally, we also want to see how the users of the monitoring system perceive the importance of such a system.

1.4 Study Approach

In this section, we describe the approach to achieve the goal of this thesis. Also, we illustrate the structure of the thesis.

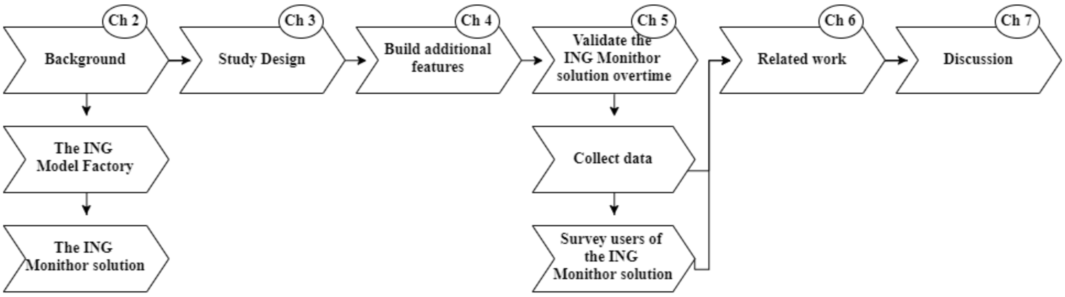

Ch 2
Background
The ING Model Factory
The ING Monithor solution
Ch 3
Study Design
Ch 4
Build additional features
Ch 5
Validate the ING Monithor solution overtime
Collect data
Survey users of the ING Monithor solution
Ch 6
Related work
Ch 7
Discussion

Figure 1.2: The structure of the thesis.

1.4.1 Background

In chapter 2, we illustrate the ING model factory, which is an automated model creation system. In order to give a better understanding, we explain the current ING business needs and status. Additionally, we demonstrate the ING Monithor solution, which is the monitoring system for machine learning models at ING in Chapter 2. In this thesis, when we use the word ’model,’ it refers to machine learning models created in the ING Monitor solution.

1.4.2 Study Design

We describe the study design of this thesis in Chapter 3. Our study consists of 3 parts: 1) Feature implementation, 2) Data collection, and 3) Survey. The feature implementation has been done based on the challenges of the existing ING Monithor solution that we identified. We validated the quality of the ING Monithor solution, including additional features over time in a mixed-method study, using both quantitative and qualitative ways for addressing our research questions[7].

1.4.3 Build additional features

In chapter 4, we investigate the challenges of the existing ING Monithor solution. Based on the results of the challenge investigation, we build new features to address these challenges in the ING Monithor solution. The design, features, and implementation are described in Chapter 4.

1.4.4 Validate the ING Monithor solution over time

We examine and verify the ING Monithor solution, including the additional features over three months. This is described in Chapter 5.

Collect data

In order to comprehend the effectiveness of the monitoring system on machine learning models in terms of managing the models, we collect data such as the number of models, and the number of users of the monitoring system in ING. After that, we evaluate the influence of the monitoring system quantitatively. Plus, we conduct a quantitative survey among the users of the actual monitoring system at ING.

Survey

In order to know how the actual users of the monitoring system in ING perceive the monitoring system itself, we execute a qualitative survey with them. We ask several questions to the users in terms of how the monitoring system supports their work. Moreover, we also investigate how the users perceive the monitoring system regarding the machine learning model management. Based on the survey results and collected data, we analyze the impact and the importance of the monitoring system.

1.4.5 Related work

Related work can be found in Chapter 6, which describes state of the art on research related to software engineering for machine learning, machine learning models and data lifecycle management, and existing solution for machine learning model management such as MLflow[35], Polyaxon[39], Prometheus[41].

1.4.6 Discussion

In the final chapter, we compare the challenges we examined and the monitoring system solution with the existing solutions. Furthermore, we discuss further research, and any threats to validations in Chapter 7.

Chapter 2

Background

In this chapter, we describe background information, needed to understand the remainder of this thesis.

In recent years, a DevOps method for software development has been developed, in order to give appropriate feedback to application developers working within rapid development cycles, and to reduce the gap between development and operations[43]. DevOps is a software process that emphasizes collaboration within and between different teams involved in software development[10]. As the DevOps method became widespread, continuous monitoring became increasingly important for dependability, quality assurance, and resilience. It collects data and metrics that are coming from the various stages of the application lifecycle. As a result, people who are involved can react quickly to improve or change functionalities[48].

ING Data Analysts also aimed to achieve a streamlining campaign selection and evaluation. They found that machine learning models are useful in accomplishing their goals. However, when users want to create a machine learning model, they have to go through all model creation steps manually. This manual process takes a lot of effort and time from users and developers. In order to decrease development time and effort, ING proposed the ING Model Factory.

The ING Model Factory is an automated system that supports the development of predictive models at ING. The model creation consists of nine stages, described in Figure 1.1 in the previous chapter. In the traditional way of machine learning model development all nine stages are followed manually. In order to make the model creation process more efficient and reduce manual work, the ING data scientists proposed the ING Model Factory. The users need to enter a configuration file into the system, with various options: business objective (e.g., acquisition, deepsell, retention), business objective specification (e.g., which product to acquire?), features, customers, and time specification (e.g., how long before a customer makes a decision?). Once done, the ING Model Factory generates the model automatically and gives the predictive model as an output. Due to this, the users do not need to follow all model creation steps, and they can make use of the created model in an easy way.

As a monitoring system for machine learning models plays a vital role in the ING Model Factory, ING tries to improve their monitoring system for the ING Model Factory. ING

Monithor is the name of the monitoring system for machine learning models at ING. The ING Monithor solution can check the existing models once they are deployed into a production environment. Also, it provides several information that helps to manage the models, such as the status of the creation process, logs, and visualized results. Before we propose improvements to the monitoring system, understanding the architecture of the ING Model Factory and the current architecture of the ING Monithor solution is essential.

2.1 The ING Model Factory

2.1.1 Requirements

In order to remove all manual processes for model creation, ING data scientist decided to develop the ING Model Factory. It needs to support accelerating model creation and building models without reinventing all the model-building processes. The ING Model Factory only provides support for models based on supervised learning methods, which uses pre-trained data[31]. The ING Model Factory requires to give a fully automated system to the users. Therefore, using the ING Model Factory makes the users do not need to go through all the model creation steps, as depicted in Figure 1.1 manually for creating a model. Also, the ING Model Factory needs to give reliable models with high performance. The ING Model Factory requirements that we identified are the following:

1. Create templates for different target labels depending on business objects (i.e. acquisitions, deepsell).
2. Comply with GDPR (General Data Protection Regulation)[55] that is a regulation in EU law on data protection and privacy for all individual citizens.
3. Model building pipeline with embedded feature selection.
4. Evaluation dashboards in ING Monitor & IBM Cognos that supports the entire analytics cycle, from discovery to operationalization[22].
5. Storage management and model versioning on the Hadoop Distributed File System (HDFS), which is a distributed file system designed to run on commodity hardware[6].
6. Smooth technical productionalization process.
7. Export result sets to PDA (IBM PureData for Analytics)[21], which offers easy data management and provides faster processing of the most complex algorithms and real-time control information.

2.1.2 Stakeholders

As ING is a large financial business company, many different roles contribute to the ING systems. In this thesis, we focus on the ING Model Factory, and a specific part of that, the ING Monithor solution. We have identified the main contributors of the ING Model Factory and the ING Monithor solution, based on the following categories: 1) Data Analysts, 2) Customer Journey Experts, 3) Data Scientists, and 4) Data Engineers.

First, the task of a data analyst is to take the data from the big data source and use it to ING stakeholders to make better business decisions. The data analysts mainly have responsibilities for the creation, management, and maintenance of machine learning models within the department they work for. Due to that, they are the main users of both the ING Model Factory and the ING Monithor solution.

Secondly, the duties of a customer journey expert are to optimize and innovate ING products and ING customer engagement. Due to that, the customer journey experts are somewhat less interested in the model creation process itself. They solely use the output of the models created by features they selected. For instance, customer journey experts request for a model for predicting the future behavior of the customers and use the output of the model that is created by data analysts.

Thirdly, Data engineers are the data experts who prepare the required big data infrastructure that is used by data analysts. They focus on designing, building, integrating, and managing this data.

Lastly, Data scientists utilize statistics, machine learning, and analytic approaches to resolving critical business problems. These two roles, data engineer and data scientist, are also the core developers of the ING Model Factory, as well as the main users of the ING Monithor solution together with the data analysts.

2.1.3 Architecture

Context View

The context view describes the dependencies and relationships of the system with its environment. We divided these into groups derived from our stakeholder analysis and external entities. In the diagram in Figure 2.1, we illustrate the ING Model Factory only. Therefore, we will describe the ING Monithor solution in a later section.

The ING Model Factory has been used only internally in ING. The core developers are the data scientists who mainly develop all processes, and other developers are the data engineers who are responsible for deployment.

The Python programming language is the primary language used in the ING Model Factory development with the Jupyter Notebook. The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text[25]. Also, the ING Model Factory is developed using GitLab for version controlling and managing continuous integration.

Structure

The structure of the ING Model Factory is shown in Figure 2.2. The ING Model Factory is composed of building blocks, a model building process, scoring functions and customers, and monitoring. In the building blocks, based on the user input, which is a configuration file, users can select the blocks such as target templates, shared features, pipelines, validation methods, classifiers, and evaluators. After that, the Model Factory assembles the selected

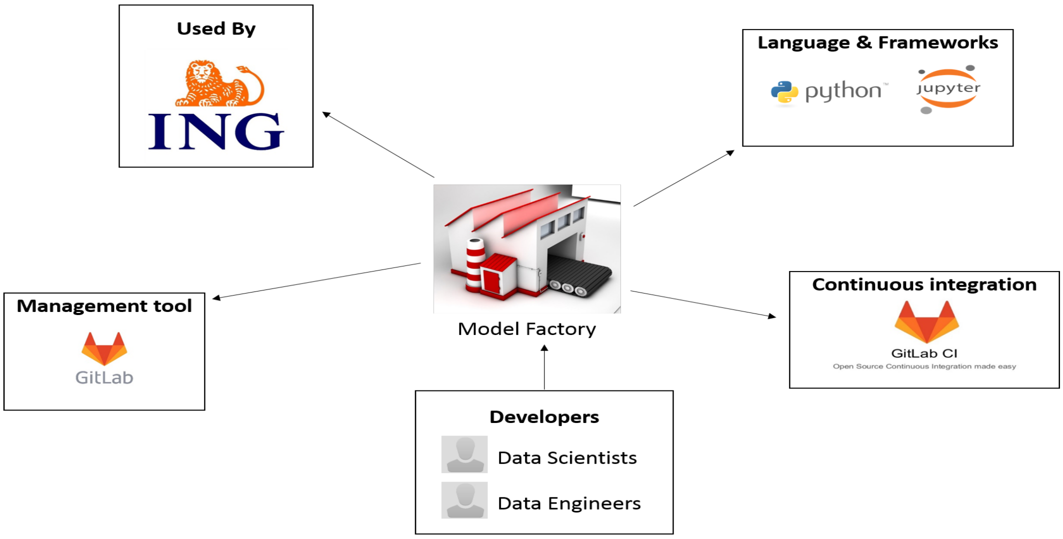


Figure 2.1: The context view of the ING Model Factory.

blocks in the model building process and trains the models. Lastly, the models are scored and checked the during evaluation and validation through the monitoring system.

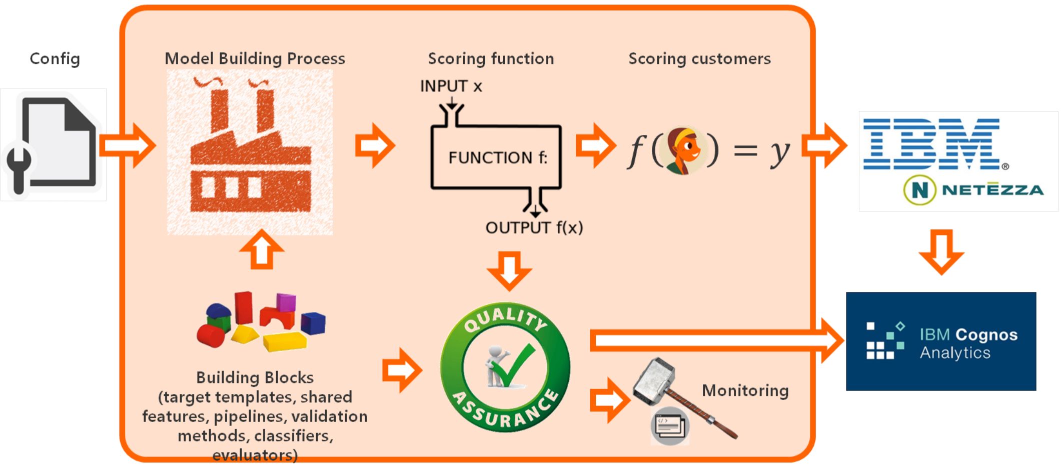


Figure 2.2: The structure of the ING Model Factory.

The details of each component in Figure 2.2 are as follows:

- Config: The input for the ING Model Factory. The input is a model specification in the form of a 10-15 lines JSON file. The model specification is prepared by a Data Analyst, based on requirements given by a Customer Journey Expert.
- Building Blocks: In a building block process, various tasks are included: 1) Creating the model feature sources, 2) Transferring the data to the model building environ-

ment, 3) Breaking down of training- and test sets based on an input dataset sorted on timestamps, 4) Preparing data for training, and 5) Filling missing values.

- Model Building Process: The model building phase is divided into two parts: 1) Training models on cross-validations and 2) Training models on the entire training set for prediction.
- Quality Assurance: Performing cross-validation to find which hyperparameters perform the best with auto-sklearn that Scikit-learn[45], which is an open-source Python library for data analysis and data mining. Also, Quality Assurance evaluates the model performance validation under the Spark MLLib evaluation metrics[49] such as classification model evaluation, or regression model evaluation.
- Scoring function & customers: Score models and customers using the ROC curve and AUC score on the test set and Precision-recall[56]. The models report the probability score for each customer to match the label in order to evaluate the predictive performance of the models, based on the most recent information on the customer.
- IBM Netezza Analytics: An IBM Netezza appliance consists of a high-performance hardware platform and an optimized database that empowers analytic enterprises to meet their business needs[21].
- Monitoring: The ING Monitor solution gives a continuous monitoring function. It will be discussed more in detail in the next section.
- IBM Cognos Analytics: A platform that supports the entire analytics cycle, from discovery to operationalization. IBM Cognos Analytics visualizes, analyzes, and shares actionable insights about data with anyone in an organization[22]. The user of the IBM Cognos Analytics can see how a threshold changes within the Cognos interface, with relation to the number of selected customers, and the expected conversion rates.

2.2 The ING Monithor solution

2.2.1 Requirements

The main goal of the ING Monithor solution is to continuously perform model validation and evaluation checks. Besides functionality for users to manage their models, the ING Monithor solution also supports users in tracking the automated process within the ING Model Factory. Therefore, even though the Model Factory fully automates the model creation process, users are still able to monitor the model creation process through the ING Monithor solution. The ING Monithor solution, built as a dedicated ING solution within the ING Model Factory, offers the following functionalities:

1. Track and access logs regarding any stages within the model creation process. This supports to decrease debugging time in case a job within the model creation process fails.

2. Perform quality checks in order to give data scientists full control over the quality of their models.

3. Synchronize the distribution of datasets and the parameters of models created by the ING Model Factory.

4. Analyze model performance over time, in order to understand any changes in model performance.

5. Track all user-defined metrics with regard to models and define thresholds in order to make sure that the models are valid and work properly.

6. Analyze the model creation process and model execution time in order to examine any error occurrences in the model creation process and model runs.

2.2.2 Architecture

Context View of the ING Monithor solution

The context view describes the dependencies and relationships of the system with its environment. We split these into groups derived from our stakeholder analysis in the previous section and external entities. The ING Monithor solution diagram is shown in Figure 2.3.

The ING Monithor solution has been used internally within ING for continuously checking model validation and evaluation. Two developers implement the ING Monithor solution with a few contributors, and the primary users of the ING Monithor solution are data analysts in ING.

In order to build the ING Monithor solution, developers utilize various libraries, languages, and frameworks. Python is the main language. Also, the ING Monithor used several frameworks and libraries such as Spark, Bootstrap, Pandas, D3, and Flask. Additionally, the ING Monithor solution also used GitLab to control the versions and manage continuous integration.

Structure of the ING Monithor solution

The ING Monithor solution consists of a back-end and front-end. In the front-end, the ING Monithor solution offers a dashboard, which shows all the models owned by a user. In the back-end, in order to enable the users to monitor their models, the ING Monithor solution offers an API. The structure of the ING Monithor solution is shown in Figure2.4.

The monitoring process consists of the following steps: 1) A user creates a project and add metrics and models in the project, 2) all data from all underlying machine learning tools is aggregated in Logtash[11] that is a server-side data processing pipeline that ingests data from a multitude of sources simultaneously, transforms it, and then sends it to storage as a JSON file, 3) the aggregated data is made searchable by using Elastic search, 4) the data is presented by using Kibana.

Both the back-end and the front-end systems are implemented in python 3[42], as follows:

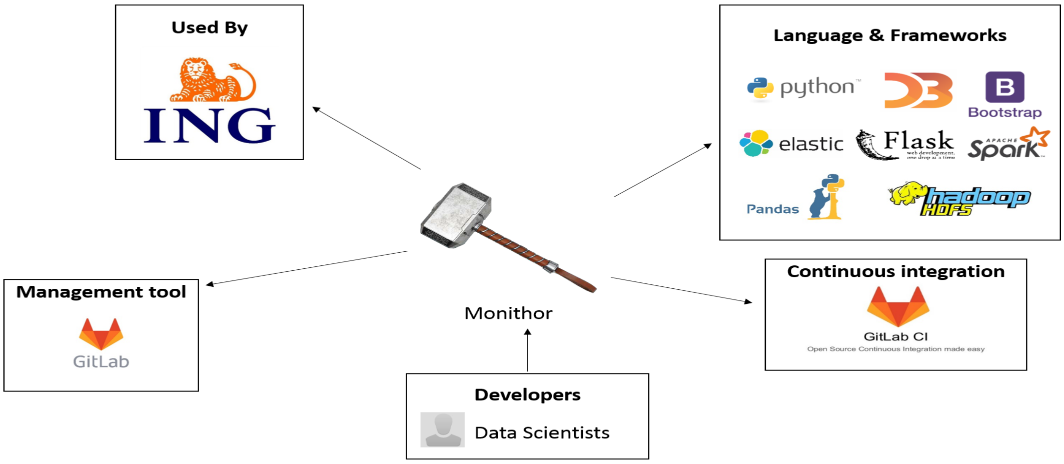


Figure 2.3: The context view of the ING Monithor solution.

- Back-end: The data is stored in a Hadoop Distributed File System(HDFS)[6]. It relies on MapReduce[2] jobs to process the data. The back-end system is based on Spark[4], and used the Python data analysis toolkit called Pandas[32]. Additionally, project files are saved as a JSON file in Logtash, and took Elastic search as a search engine[19].

- Front-end: The web dashboard is designed by using D3[8] and Bootstrap[5]. The developers chose Flask[14] as a web framework, and used Kibana as a dashboard[19].

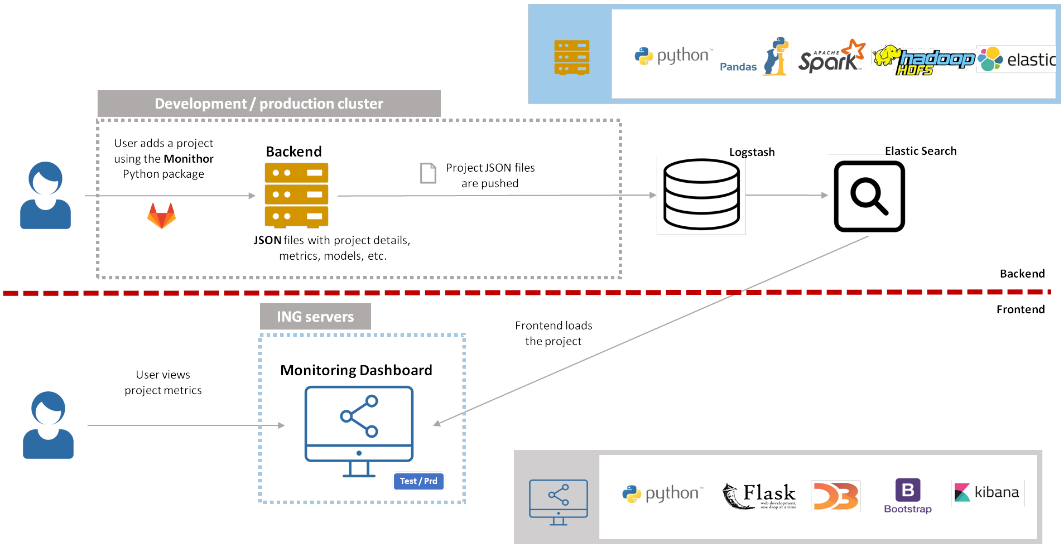

python
Pandas
Spark
hadoop HDFS
elastic
Development / production cluster
User adds a project using the Monithor Python package
Backend
Project JSON files are pushed
JSON files with project details, metrics, models, etc.
Logstash
Elastic Search
Backend
Frontend
ING servers
Frontend loads the project
Monitoring Dashboard
User views project metrics
Test / Prd
python
Flask
Bootstrap
kibana

Figure 2.4: The structure of the ING Monithor solution.

Based on the red dotted line in the middle, the above is the back-end and the below is the front-end.

In the study that we carried out as part of this thesis, we investigated the challenges faced by users of the existing ING Monithor solution, we developed additional features as solutions to these challenges, and we validated the quality of these additional features over time in a case study. In the next chapter we will look more closely at the case study design and the various components in it.

Chapter 3

Study Design

In this chapter, we illustrate the study design to answer our research questions. Our study is structured in the following consecutive steps (see Figure 1.2 for an overview of the study approach):

1. Inventory of challenges in the existing ING Monithor solution;
2. Build and implement additional features in the ING Monithor solution as an answer to the challenges;
3. Analyze and validate the ING Monithor solution—including the newly built additional features—over a period of three months;
4. Compare the results of the validation over time with other comparable monitoring solutions described in related work;
5. Discuss the results of our study with regard to implications for research and industry, and any threats to validity.

In this section we start with an inventory of our research questions, after which we further explain the five successive steps in our study.

3.1 Research Questions

For our study, we are following two research questions.

RQ1: To what extent can a monitoring system support stakeholders in terms of maintaining machine learning models?

The main reason for having a monitoring system is that it facilitates managing and maintaining the models and gives dependability, quality assurance, and resilience. In order to analyze the influence of the ING Monithor solution, we divided the quantitative research into two sections: data collection and a survey. We use the result of both sections to address our first research question.

RQ2: How do users of a monitoring system perceive the way it supports the models with regard to usability, quality, and trust?

From a model quality perspective, we want to analyze the impact of the ING Monithor solution. The survey questions address how the users perceived the ING Monithor solution. We adopted the definition of usability described in ISO 9241. The definition of usability in the ISO 9241 standard is: ”The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use”[24]. Since Frokjaer et al. described that effectiveness, efficiency, and satisfaction should be considered an independent aspect of usability[16], we made survey questions for each aspect.

3.2 Step 1: Inventory of challenges

We organize workshops and interviews for users of the ING Monithor solution in order to investigate the difficulties of the ING Monithor solution from the users. The challenges found are discussed in Section 4.1.2.

3.3 Step 2: Build and implement additional features

From the workshops for the users of the ING Monithor solution, we are able to discover several challenges of the ING Monithor solution. Based on these challenges investigated by the users, we build and implement new functionalities into the ING Monithor solution in order to address these challenges. These new functionalities are described in Chapter 4.

3.4 Step 3: Analyze and validate additional features over time

After adding the new features that dissolved the challenges, we examine and verify the ING Monithor solution over three months by collecting data and conducting a survey to users of the ING Monithor solution.

3.4.1 Data Collection

During the first part of the quantitative research, we collected the data, including, the number of models over time, and model usage over time. Since the ING data scientists deployed the ING Model Factory and the ING Monithor solution in a production environment in the middle of June, we collected the data in the period June 21 to September 7, 2019 (the end of the internship). In total, we collected data for 19 models.

3.4.2 Survey

To assess the usage of the ING Monithor solution, we conducted an online survey[15] to measure the reviewing time and the usage frequency of the ING Monithor solution. The details of the survey design are described in the following subsections.

Survey Operation

The survey has been implemented in Survalyzer, which is a survey management platform internal to ING. The candidate participants were invited using an invitation letter featuring the purpose of the survey. We e-mailed 28 users and obtained 20 responses (71.42% response rate). Respondents had a total of three weeks to participate in the survey. We sent two reminders to those who had not participated yet at the beginning of the second and third weeks. The survey ran from September 3 to September 26, 2019.

Survey Design

The survey was organized into three sections:

1. Part 1: Demographics

    1.1 Job role In ING*

    1.2 Years of ML experience

    1.3 Years of monitoring ML models experience

    1.4 Years of ING Monithor solution experience

2. Part 2: Quantitative

    2.1 The number of models to verify*

    2.2 Time spent on ING Monithor solution*

    2.3 Time spent on managing models via ING Monithor solution*

3. Part 3: Qualitative

    3.1 Challenges*, **

    3.2 ING Monithor solution features for an effective model management**

    3.3 Ease of use*

    3.4 Efficiency of use*

    3.5 Model quality*

    3.6 Trust of the automated model creation*

    3.7 Importance of ING Monithor solution*

An asterisk indicates that an open-ended is included. A double asterisk indicates that a ranking question is included. The details are illustrated in Appendix A.

Survey Part 2: Quantitative

After collecting demographic information of the respondents, the second part of this survey links to RQ1. For this purpose, we collect the quantitative information on the number of models, time spent on ING Monithor solution, and time spent on managing models through the ING Monithor solution. All three questions are open-ended questions mixed with questions with five options. Also, we provided respondents with a set of open-ended questions to help us understand their choices.

Survey Part 3: Qualitative

The qualitative survey questions are composed of open-ended questions mixed with a set of five optional 5-level Likert scale questions format from Strongly Disagree(1) to Strongly Agree(5)[30], and ranking questions. We are aware of the complex meaning of 'Neutral' options, which can be divided into two groups, such as opinion neutrality and no opinion[18]. Therefore, to further explore the opinions of respondents, we included a 'Do not know' option in all Likert scale questions.

To be able to map the results of question 3.1 asking about challenges with the results of the Microsoft study[3], we decided to use the same challenges as discovered by the Microsoft software engineering team in a ranking list. Also, we asked the users to write 3 challenges. For question 3.2, we asked the participants to rank seven significant features of the ING Monithor solution. We used our own judgments to pick these seven features.

The remaining survey questions are about the user perception for the ING Monithor solution features on machine learning model management. To address RQ2, we asked how the users perceived the ING Monithor solution in terms of supporting easy and efficient model management, increasing the quality of the models, and giving trust in the automated system. Also, we asked how they perceived the importance of the ING Monithor solution. We provided respondents with a set of open-ended questions to help us understand their choices. The detailed survey design is described in Appendix A.

Selection of Participants

To explore the impact of the ING Monithor solution, we sent the survey to the actual users of the ING Monithor solution. All respondents are working at ING. The participants must have experience with the ING Monithor solution and with machine learning models in general and within ING. In total, we contacted 28 participants, each working on their ING models.

3.5 Step 4: Compare the results with related work

We found 3 existing open-source monitoring systems that cover most major deep learning frameworks and give visualization features. We compare the ING Monithor solution to these 3 existing open-source solutions with regard to functionalities. The related work will be described in Chapter 6, and the result of the comparison will be discussed in Chapter 7.

3.6 Step 5: Discuss the results

To wrap up our study, in Chapter 7, we describe the implications of our study on research and industry, and we illustrate the internal and external threats to validity.

Chapter 4

ING Monithor Solution Extensions

This chapter describes the new functionalities of the ING Monithor solution that we built and implemented within the scope of this thesis. We implemented the new features into the existing ING Monithor solution, based on the existing challenges we identified. The inventory of challenges and building and implementation of all additional features described in this section were within the scope of our study and, therefore, have been done during the internship period.

4.1 The ING Monithor solution Features and Challenges

In this section, we describe the features of the ING Monithor solution as they were before the start of our study, and inventory any challenges that ING wants to solve as part of our study. Since the goal of our study is to build, implement, and validate some additional features into the ING Monithor solution, in order to solve limitations and flaws of it, it is important to understand the challenges of the current system and current features. However, first, we summarize the features that were part of the system before we started our study (here identified as the existing ING Monithor solution).

4.1.1 Features in the existing ING Monithor solution

The existing ING Monithor solution included the following features:

- Tracking user-defined metrics associated with the model, and users define thresholds overtime for these metrics.
- Viewing and analyzing parameters of trained Scikit-learn[38] and Spark models.
- Synchronization of the distribution of dataset and parameters of the model, and comparison of the data distribution of predicted values and features.
- Track the progress of the running data science pipeline.
- Search and display of log messages related to the data science pipeline.

4.1.2 Inventory of Challenges

Together with ING data scientists and data analysts, we participated in workshops and interviews with the ING Monithor solution stakeholders about functions and requirements. The interviewees were selected by two criteria: 1) they had to be an expert in their field, and 2) all roles of the stakeholders had to be included. Based on the results of the workshops and interviews, we found that the ING Monithor solution has several flaws and challenges in terms of supporting model management to users. We describe three of the most mentioned problems in the following paragraphs:

Lack of one single overview for multiple projects

Even though the ING Monithor solution is able to show a list of all projects with their running status, it is not possible to see multiple projects in one single overview. Therefore, when users want to check whether their projects are valid or not, they need to check each model separately, which is not intuitive. Besides that, checking all projects one by one takes a lot of time and effort.

Thresholds are not stored

Although the developers use thresholds in metrics, these are not stored. Hence, if the users want to analyze the thresholds at a later stage, they need to re-do the model creation process and check their input by themselves, since they cannot check it in a dashboard. Besides that, the thresholds can be changed every run. If that is the case, recognizing the thresholds and managing the models is even more complicated than when thresholds did not change.

The solution is not intuitive and user-friendly

The ING Monithor solution is experienced by its users as not intuitive enough, and not as a user-friendly monitoring system. The users are able to handle a lot of information about the projects and models such as logs, output graph, model validation in the dashboard. However, the overview of the dashboard is confusing and lacks any explanation of information and metrics.

4.1.3 Overview of additional features

We aimed to address the challenges mentioned above and drawbacks in additional features in the ING Monithor solution. After analyzing the challenges in relation to the features in the existing ING Monithor solution, we decided to include the following additional features as extensions to this existing solution:

1. Overview of Multiple Projects;

2. Expanding the thresholds;

3. Manage Projects by tags;

4. Improve presentations;

The various new features are described in detail in the following sections.

4.2 Design of Features

4.2.1 Overview of Multiple Projects

The existing ING Monithor solution showed a detail of one project only, so it was hard for users to manage multiple projects in one time. In order to provide better model management, we designed and developed an intuitive status overview dashboard. In this dashboard, we provide a list of multiple projects with each status in many phases and stages.

A vital design of the dashboard is monitoring the main stages of a project that users can define. Examples of such stages are building models, input data, scoring models, and evaluation of models. Each project has a flag section that gives an overview of the issues investigated, including a colored indication of errors and warnings. The flag supports users to identify errors or warnings that need action. The following flag colors occur: 1) Red: errors, 2) Yellow: warnings, and 3) Green: ok. For example, when the data drops or raises more than 5%, or when the data violates a threshold, the ING Monithor solution raises the red flag with a detailed description.

Additionally, we designed a linking function to give users an opportunity to quickly go to particular stages and logs. When some errors occur, the system displays the red flag in the label with some explanation. Since the label is clickable, the user can go to a particular stage and a log where the error occurred by clicking the label.

4.2.2 Expanding the Thresholds

We expanded two things for thresholds: 1) Store thresholds data and visualize that, and 2) Add an upper bound threshold function.

The existing ING Monithor solution did not store the thresholds. For that reason, users could not analyze threshold information once the model creation process was finalized. In order to solve this problem, we decided to store the thresholds and represent them within the metrics graph in order to create a more intuitive visualization. Consequently, users can see both metric values and thresholds in the metrics graph at the same time.

The existing ING Monithor solution has functionality for the lower bound threshold only. However, we sometimes need to use the upper bound thresholds or both lower bound and upper bound thresholds in order to have a strong validation. Accordingly, we determined to add the upper bound threshold function and visualized it in the same way as the lower bound threshold.

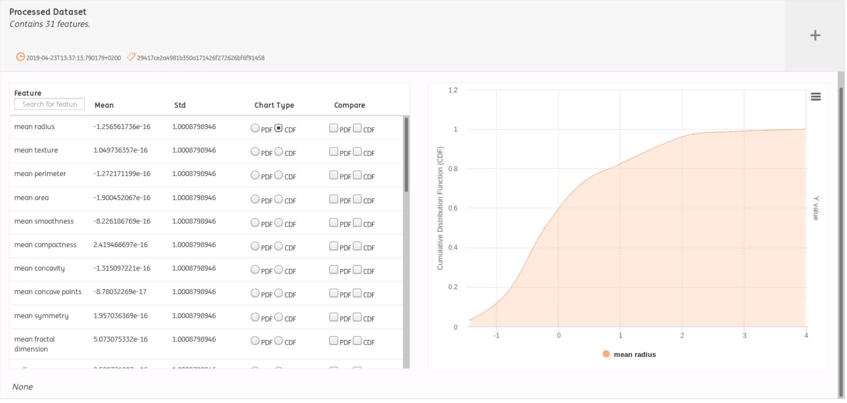
4.2.3 Manage the Projects by Tags

The increasing number of projects might lead to another difficulty in managing projects. For example, when the number of projects is over 50, the way of presenting the projects can be a problem. Since the system shows a list of all projects in a view, users need to search the projects one by one or scroll down all projects to find their projects.

In order to address this problem, we found that tags are useful to improve model management by filtering the projects. So we decided to implement the tag function. Each model can have multiple tags, and the tags can be defined by the users when they create the projects.

4.2.4 Improving the Presentation

While the ING Monithor solution has been used to maintain models, the users had 3 requirements described in the previous chapter regarding managing their models. Moreover, the users found minor practical issues that need to be solved. We made changes to address them.



Processed Dataset

Contains 31 features.

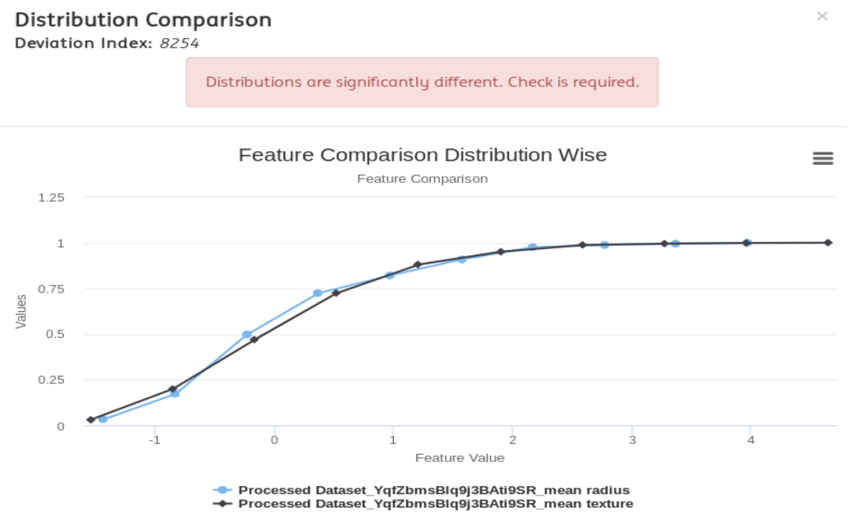
2019-04-23T13:37:13.790179+0200 29417ce2a4981b350a171426f272626bf6f91458

Feature Search for feature	Mean	Std	Chart Type	Compare
mean radius	-1.256561736e-16	1.0008798946	○ PDF ● CDF	□ PDF □ CDF
mean texture	1.049736357e-16	1.0008798946	○ PDF ○ CDF	□ PDF □ CDF
mean perimeter	-1.272171199e-16	1.0008798946	○ PDF ○ CDF	□ PDF □ CDF
mean area	-1.900452067e-16	1.0008798946	○ PDF ○ CDF	□ PDF □ CDF
mean smoothness	-8.226186769e-16	1.0008798946	○ PDF ○ CDF	□ PDF □ CDF
mean compactness	2.419466697e-16	1.0008798946	○ PDF ○ CDF	□ PDF □ CDF
mean concavity	-1.315097221e-16	1.0008798946	○ PDF ○ CDF	□ PDF □ CDF
mean concave points	-8.78032269e-17	1.0008798946	○ PDF ○ CDF	□ PDF □ CDF
mean symmetry	1.957036369e-16	1.0008798946	○ PDF ○ CDF	□ PDF □ CDF
mean fractal dimension	5.073075332e-16	1.0008798946	○ PDF ○ CDF	□ PDF □ CDF

None

Figure 4.1: The result of the CDF in the ING Monithor solution.

The left area is a data summary section, and it describes all features and mean values. Users can choose a chart type probability density function (PDF) or cumulative distribution function (CDF), and then the chart is shown in the right area. A data detail field, which is the left area, enables the users to click the compare checkbox to see the difference between results.



Distribution Comparison

Deviation Index: *8254*

Distributions are significantly different. Check is required.

×
Feature Comparison Distribution Wise
Feature Comparison
1.25
1
0.75
0.5
0.25
0
Values
-1
0
1
2
3
4
Feature Value
Processed Dataset_YqfZbmsBlq9j3BAti9SR_mean radius
Processed Dataset_YqfZbmsBlq9j3BAti9SR_mean texture

Figure 4.2: The result of the CDF comparison in the ING Monithor solution.

This figure is a sequence result of Figure 4.1 when users select a compare checkbox at two values. A chart draws the values that the user selected. A label above the chart shows the status of the comparison. If the comparison results are the same, the label represents a green color. However, if the comparison results are different, the label depicts a red color.

First, the users require a description field because when the users did not engage in a model creation process, only giving results was not enough to make them understand. Thus, we designed the description field in which the users can add an explanation when they run or create the model.

Second, the users want to have a cumulative distribution function (CDF) to give a better understanding together with a probability density function (PDF) that the ING Monithor solution already had. As a result, we implemented the CDF in the same area with PDF. The users can make a comparison graph, which helps to get intuitive visual comparison results between different runs. The results are shown in Figure 4.1, 4.2.

Lastly, the ING Monithor solution had practical issues regarding the user interface (UI) and functions: buttons, bugs, data loading fails, and resizing fails. Therefore, we solved the UI problems and functional problems in the ING Monithor solution. For example, we changed the ways to show the metrics and tables to help the users so they can easily read the result. Also, we solved a couple of bugs that existed in the ING Monithor solution.

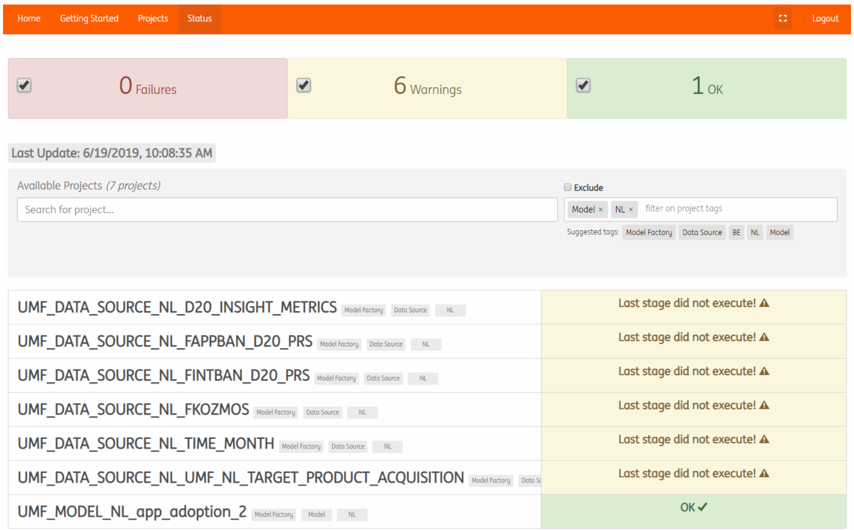
4.3 Implementation of Features

The ING Monithor solution needs to give an intuitive user interface design and compliance with the requirements of the stakeholders. Moreover, we preserve the underlying architecture, and add new features based on this architecture described in the previous chapter.

All implementations of the ING Monithor solution are classified into four groups based on the features: 1) Status view, 2) Thresholds extensions, 3) Tags, and 4) Presentation improvements. The presentation improvements are mostly minor changes. Therefore, we do not describe them in this section. The leading developers of the ING Monithor solution at ING reviewed the feature implementation code.

4.3.1 Status View

One of the essential features of the monitoring system is maintaining the status of all projects easily. As the results of the implementation are depicted in Figure 4.3, 4.4, we implemented the status view, which provides an intuitive overall status of all existing projects. The project status covers the overall status range of the model creation and model validation and evaluation within the project.



☑ 0 Failures ☑ 6 Warnings ☑ 1 OK

Last Update: 6/19/2019, 10:08:35 AM

Available Projects *(7 projects)*

Search for project...

□ Exclude

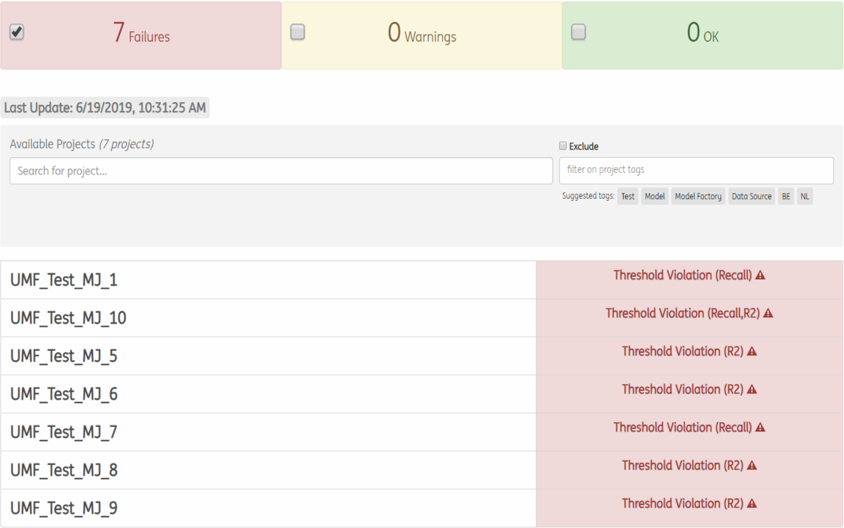
Model × NL × filter on project tags

Suggested tags: Model Factory Data Source BE NL Model

UMF_DATA_SOURCE_NL_D20_INSIGHT_METRICS Model Factory Data Source NL	Last stage did not execute! ⚠
UMF_DATA_SOURCE_NL_FAPPBAN_D20_PRS Model Factory Data Source NL	Last stage did not execute! ⚠
UMF_DATA_SOURCE_NL_FINTBAN_D20_PRS Model Factory Data Source NL	Last stage did not execute! ⚠
UMF_DATA_SOURCE_NL_FKOZMOS Model Factory Data Source NL	Last stage did not execute! ⚠
UMF_DATA_SOURCE_NL_TIME_MONTH Model Factory Data Source NL	Last stage did not execute! ⚠
UMF_DATA_SOURCE_NL_UMF_NL_TARGET_PRODUCT_ACQUISITION Model Factory Data Sc	Last stage did not execute! ⚠
UMF_MODEL_NL_app_adoption_2 Model Factory Model NL	OK ✓

Figure 4.3: The status view in the ING Monithor solution.

The top of the image shows a status label. The label is composed of 3 statuses: 1) failures, 2) warnings, and 3) ok. The label presents the total number of projects in each status. In the middle of the figure, there is a search area and a tag field. Users can search their projects through the search section, and the results are shown the projects that include the retrieved text. Plus, the number of projects next to the Available Project label shows the retrieved number of projects. Furthermore, users can filter a list of projects by tags. The results of filtering with tags are represented in the bottom part of the figure. Each line shows the name of the project, the status of the project, and tags in order.



☑ 7 Failures ☐ 0 Warnings ☐ 0 OK

Last Update: 6/19/2019, 10:31:25 AM

Available Projects *(7 projects)*

Search for project...

☐ Exclude

filter on project tags

Suggested tags: Test Model Model Factory Data Source BE NL

UMF_Test_MJ_1	Threshold Violation (Recall) ⚠
UMF_Test_MJ_10	Threshold Violation (Recall,R2) ⚠
UMF_Test_MJ_5	Threshold Violation (R2) ⚠
UMF_Test_MJ_6	Threshold Violation (R2) ⚠
UMF_Test_MJ_7	Threshold Violation (Recall) ⚠
UMF_Test_MJ_8	Threshold Violation (R2) ⚠
UMF_Test_MJ_9	Threshold Violation (R2) ⚠

Figure 4.4: An example of the model failures in the status view.

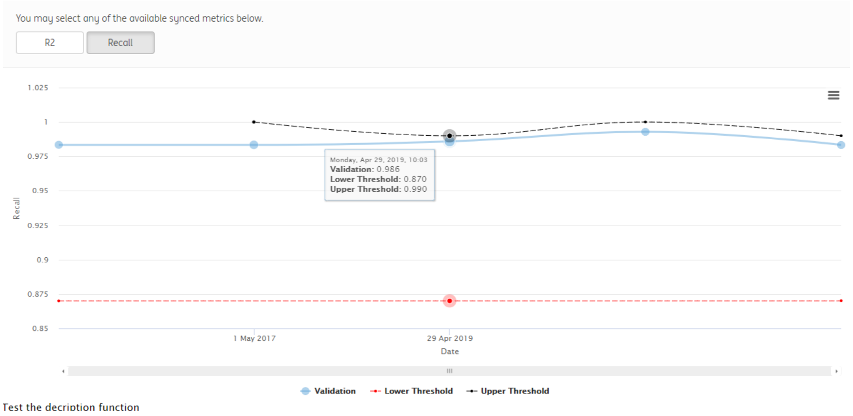
This example shows that the status view is filtered a list of projects by clicking the status labels.

4.3.2 Thresholds extensions

The main goal of the ING Monithor solution is verifying and evaluating the models. In order to check the model validation from the outputs, we needed to set up the thresholds and examined the results of the model runs. Thus, we implemented the visualized thresholds lines on the graph to give an intuitive way of showing the values. The thresholds are stored in Logtash as a JSON file, and we get the data from Logtash using Elastic Search to Kibana, then draw it on the Highcharts graph. The outcome is shown in Figure 4.5.

4.3.3 Projects Management by Tags

Since the number of projects increases, users want to have a better way of handling their projects. As Figure 4.6 shows, we implemented the tag function that helps to manage the list of the projects easily.



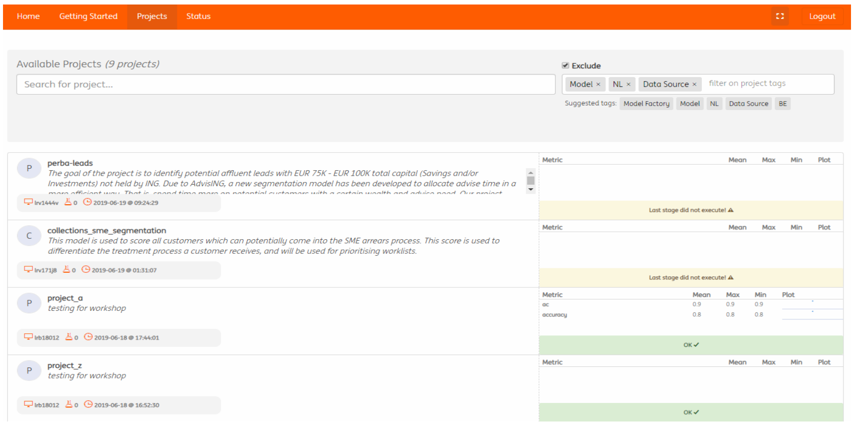
You may select any of the available synced metrics below.

Monday, Apr 29, 2019, 10:03
Validation: 0.986
Lower Threshold: 0.870
Upper Threshold: 0.990

Test the decription function

Figure 4.5: The result of the thresholds lines in the ING Monithor solution.

The result depicts a metrics section, and the buttons configure metrics. Each metric consisted of 3 components: 1) upper bound thresholds, 2) lower bound thresholds, and 3) metric value. A black dotted line depicts the upper bound thresholds, and a red dotted line indicates the lower bound thresholds. Both thresholds are not must-have value; both or one threshold can be empty. On the other hand, a blue line represents the value of the metric run, and the value cannot be empty.



Available Projects *(9 projects)*

Search for project...

☑ Exclude

Model × NL × Data Source × filter on project tags

Suggested tags: Model Factory Model NL Data Source BE

P perba-leads

The goal of the project is to identify potential affluent leads with EUR 75K - EUR 100K total capital (Savings and/or Investments) not held by ING. Due to AdvisING, a new segmentation model has been developed to allocate advise time in a more efficient way. That is, spend time more on potential customers with a certain wealth and advice need. Our project

lrv1444v 0 2019-06-19 @ 09:24:29

Metric	Mean	Max	Min	Plot

Last stage did not execute! ⚠

C collections_sme_segmentation

This model is used to score all customers which can potentially come into the SME arrears process. This score is used to differentiate the treatment process a customer receives, and will be used for prioritising worklists.

lrv171j8 0 2019-06-19 @ 01:31:07

Metric	Mean	Max	Min	Plot

Last stage did not execute! ⚠

P project_a

testing for workshop

lrb18012 0 2019-06-18 @ 17:44:01

Metric	Mean	Max	Min	Plot
ac	0.9	0.9	0.9	
accuracy	0.8	0.8	0.8	

OK ✓

P project_z

testing for workshop

lrb18012 0 2019-06-18 @ 16:52:30

Metric	Mean	Max	Min	Plot

OK ✓

Figure 4.6: An example of the tagging feature in the Projects view.

The Projects view is composed of three sections: 1) search area, 2) tag area, and 3) project list area. The top of the image involves the search area and the tag area. Moreover, the bottom of the result represents the project area. Each project has a description section, an auxiliary information field, a metrics area, and a status label. The project description is defined by the users when they create the project. The auxiliary information field, which is a small gray box inside of the project area, indicates the server information, the number of unique projects, and creation time. The metrics area depicts all the metrics that the project has. Lastly, the status label illustrates the status of the project. Additionally, when the users choose the tags, the outcome is filtered by tags.

Chapter 5

Validation over time

In this chapter, we report results from 1) the analysis of collected data, and 2) the analysis of a survey. Plus, we answer our research questions based on analyzing and evaluating the results of the collected data and the survey.

5.1 Collected Data Analysis

We gathered two types of data of the ING Monithor solution: 1) the number of models over time, and 2) the model usage over time.

5.1.1 The Number of Models Over Time

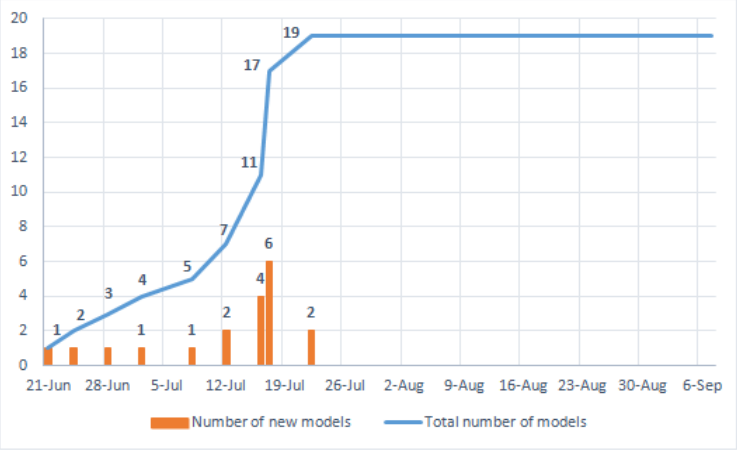


Figure 5.1: The Number of ML Model deployed in ING Monithor solution over time.

The x-axis illustrates the date when the model is created, and the y-axis describes the number of models. The orange bar represents the number of new models deployed, and the blue line shows the total number of models in the ING Monithor solution.

First, as explained in Figure 5.1, the number of models deployed in the ING Monithor solution, increased gradually over time. The number of models highly increased around July 16 due to new users. As the graph shows, after June 21, the trend is flattening out because new models are not deployed anymore after this date. We assume that the fact that workshops for new users are still ongoing, the workshops are done in a development environment, and overlapping with the holiday period, might be a reason for this.

5.1.2 The Model Usage Over Time

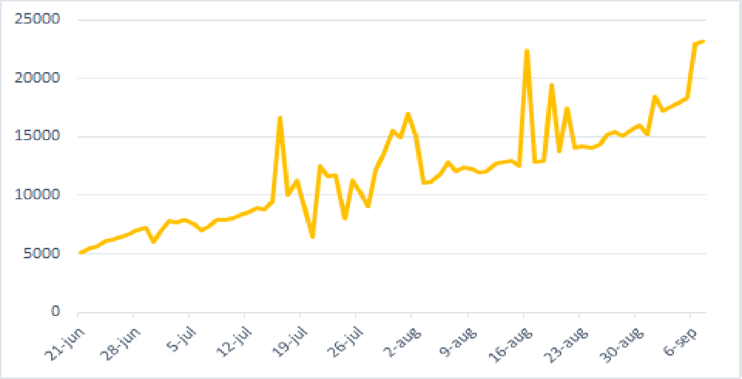


Figure 5.2: ML Model Usage in ING Monithor solution.

The x-axis represents the date when models used, and the y-axis demonstrates the usage of models.

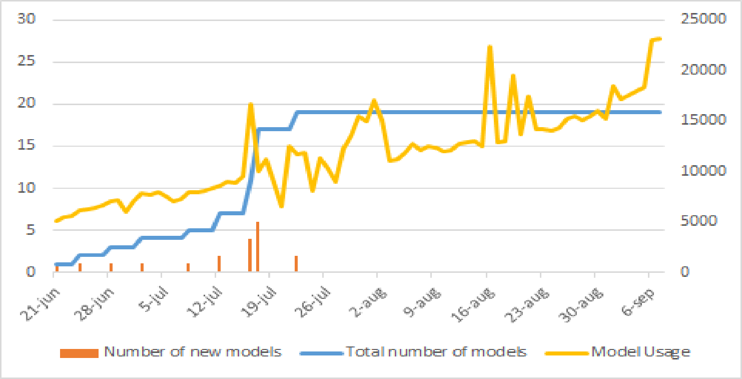


Figure 5.3: Combination Chart for the Number of Models and Model Usage.

The x-axis designates the date. The left y-axis illustrates the number of models for the orange and blue data, and the right y-axis represents the number of model usage for the yellow line.

The model usage perspective, as depicted in Figure 5.2 shows an increasing tendency of model usage in general. In order to see the reason for the increasing tendency of the model usage, we combined the two charts Figure 5.1 and 5.2. The result is shown in Figure 5.3.

From the combination chart, it shows that at first, both model deployment and model usage, are increasing equally. Around July 19, a relatively high number of models are deployed. After this period, the deployment flattens out, at the same time, the model usage value is increasing further. We argue that the model usage is influenced by both factors: the number of models and the number of users.

5.2 Survey Results

In this section, we describe the three sections resulting from our survey.

5.2.1 Demographics

Job Role

As Figure 5.4 shows, 4 out of 20 respondents (20%) work as a data analyst, and 2 out of 20 survey contributors (10%) have a data engineer job role in ING. A majority (60%) of our participants (12 out of 20) work as a data scientist, while the rest have a machine learning engineer or a software engineer (5% each) role at the IT department at ING called ING Tech.

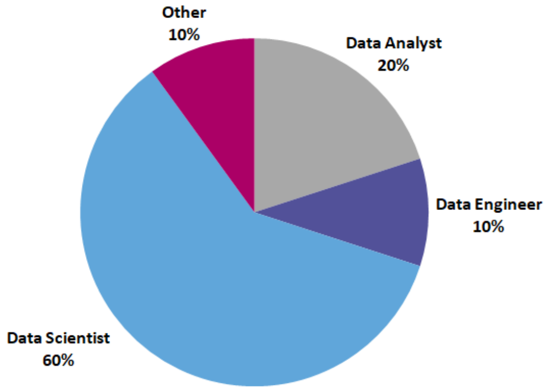


Figure 5.4: Job Role of the Survey Respondents.

ML Experience

Three respondents dropped out of the survey because they did not have any experience in Machine Learning models. Therefore, collected responses that we used are 17 out of 28 users (60.71%). As Figure 5.5 shows, the machine learning model experience of the survey participants is scattered. 6 participants (30%) have 1 year or less, 3 respondents (15%) have

2 to 3 years, 6 contributors (30%) have 6 to 7 years, and 1 respondent (5%) each in 8 to 9 years and 10 years or more.

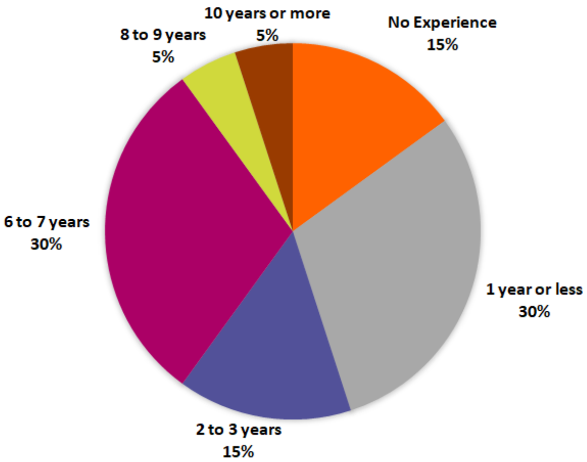


Figure 5.5: ML Model Experience of the Survey Respondents.

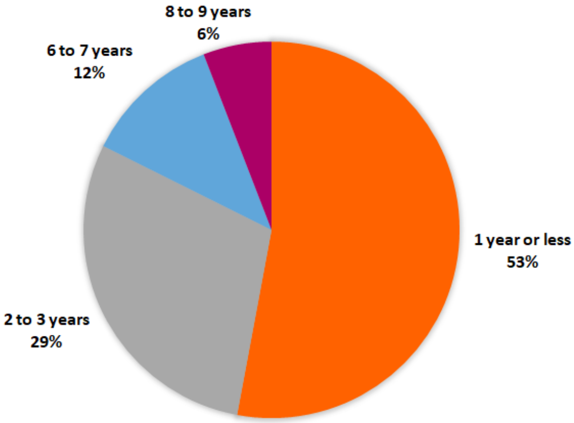


Figure 5.6: ML Model Monitoring Experience of the Survey Respondents.

Monitoring ML Models Experience

As presented in Figure 5.6, most respondees have 1 year or less of experience with monitoring machine learning models in general in- and outside ING (53%). 5 participants (29%) have 2 to 3 years experience, and a minority of respondents have over 6 to 9 years of experience with monitoring machine learning models (12% and 6%). Thus, a majority of the respondents have less than 3 years of experience (72%).

ING Monithor solution Experience

As shown in Figure 5.7, most participants (76%) reported having 1 year or less experience with the ING Monithor solution. Only 1 respondee has 6 years or more experience. Thus, most respondents have less than 3 years of experience with the ING Monithor solution (94%).

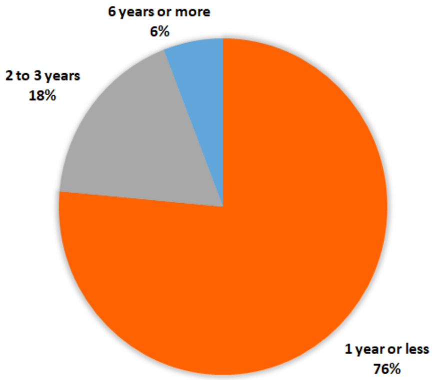


Figure 5.7: ING Monithor solution Experience of the Survey Respondents.

5.2.2 Quantitative Results

In order to collect more diverse data on the number of deployed models and the time spent on modeling, we asked three quantitative questions in the survey, as shown in Subsection 3.4.2 Part 2. The analysis of the quantitative survey section resulted in the following observations.

The Number of Models to Verify

A majority of the respondents reported having between 1 and 6 models in scope for verification. Only 1 respondee reported having more than 10 models in scope for verification. The result of this survey question is shown in Figure 5.8.

Time Spent on the ING Monithor solution

A majority of the participants mentioned that they use the ING Monithor solution less than once in a week, once or 2-3 times in a week, as presented in Figure 5.9. We also asked open-ended questions to clarify the answers, indicating that some of the respondents used the ING Monithor solution once in two weeks. Plus, only one of the respondees spent time

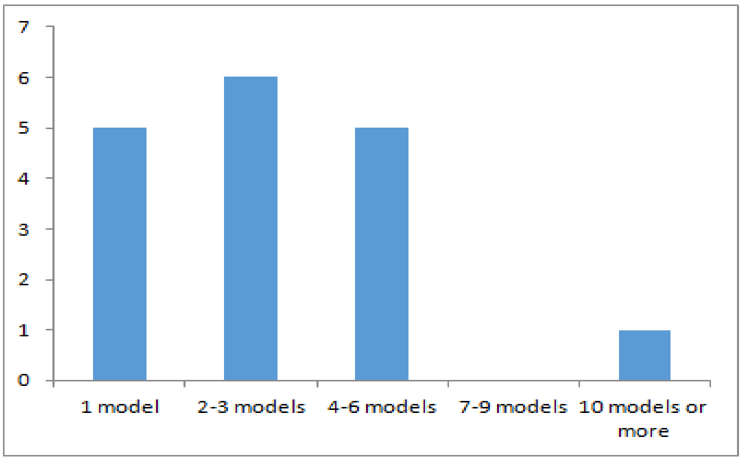


Figure 5.8: Responses of the Number of Models to Verify.

4-6 times in a week on the ING Monithor solution. The contributor who used the ING Monithor solution 4-6 times in a week, has 4-6 models, which is a relatively high number of models. Thus, we assume that the included number of models may affect the time spent on the ING Monithor solution. However, the involved number of models is not a single reason because the participant who has the highest number of models spent less than once in a week on the ING Monithor solution.

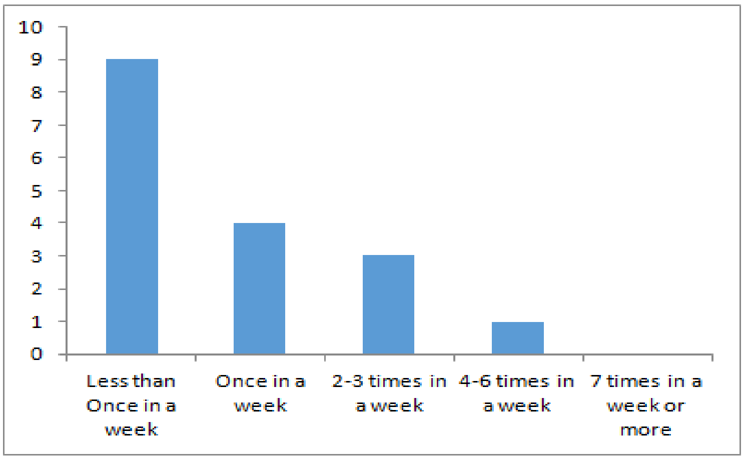


Figure 5.9: Responses of the Time Spent on ING Monithor solution.

Time Spent on Managing Models through the ING Monithor solution

As we can see from the result of the statements in Figure 5.10, a significantly high number of respondents mentioned that they spent approximately less than 5 minutes on managing models via the ING Monithor solution (94%). Only one respondent spent approximately 30 minutes on this. However, the respondent has the highest number of models within all participants. Therefore, we argue that the ING Monithor solution supports agile model management, which can be finished within 5 minutes.

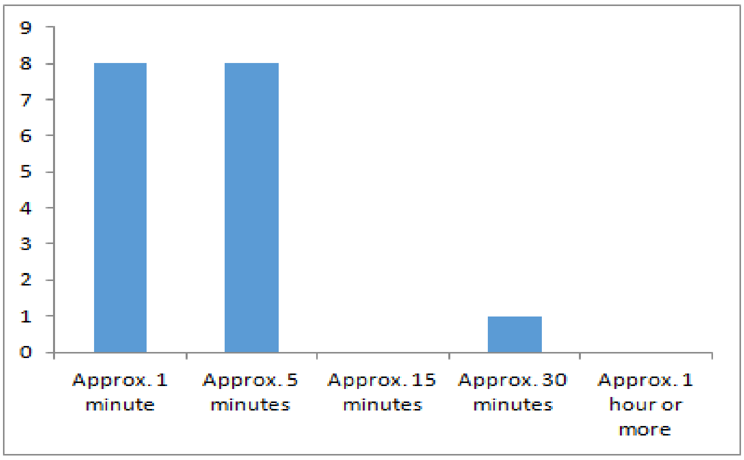


Figure 5.10: Responses of the Time Spent on Managing Models.

5.2.3 Qualitative Results

The qualitative part is divided into 7 sections of the survey structure shown in previous section 3.4.2 Part 3: 1) ING Monithor solution challenges, 2) ING Monithor solution features ranking, 3) Ease of use, 4) Efficiency of use, 5) Model quality, 6) Trust of the automated model creation, and 7) Importance of the ING Monithor solution.

ING Monithor solution Challenges

We asked respondees to answer the challenges encountered while using the ING Monithor solution. Furthermore, we requested to rank the given challenges. We are going to explain these two results individually later. We used a list of challenges resulting from a Microsoft study[3] for this purpose. Since the prior experience of participants with machine learning and data science is heterogeneous as well as their experience adjusted the knowledge of participants, Amershi et al.[3] divided their survey data into three groups: Low, Medium, High, depending on the AI experience of respondents. We reused their method to analyze our survey data. We grouped respondees into three groups: 1) Low: respondents with 1 year or less experience (n = 9, 53%), Medium: respondents with 2 to 3 years experience (n = 5, 29%), and 3) High: respondents with 6 or more years experience (n = 3, 18%).

First, concerning the challenges encountered, the participants mentioned challenges of the ING Monithor solution. We clustered the answers of the open-ended question to similar responses. We collected 13 different challenges in total. The results of clustering challenges are shown in Figure 5.11, and only results that occur more than once are included in the figure. The most mentioned challenge by respondents is the Ease of implementation. As some participants put in, "Not easy to add custom features" and "Different components need to be connected." The challenge Error Tolerance and Robustness comes next.

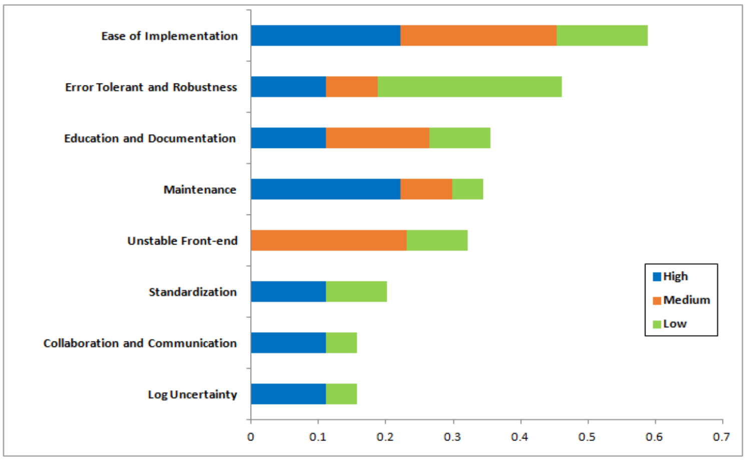


Figure 5.11: Ratio of Responses of Challenges for all Experience Groups.

The graph shows the accumulated ratio of the challenges mentioned by respondents for each experience group(Low, Medium, High). The results are sorted in decreasing order.

Figure 5.12 shows the Ratio of responses for each experience group. Two things are worth noticing here. First, the most concerned challenges are different regarding the experience of respondents. Low experienced respondees rank Error Tolerance and Robustness as an important challenge, while high experienced participants rank Ease of Implementation and Maintenance as important challenges. Second, various challenges are not mentioned depending on the experience level of respondents. For example, high experienced respondents did not mention Unstable Front-end as a challenge, while low and medium experienced respondents did.

When we interpret these results, the perceived challenges are heterogeneous depending on the experience level of respondents. Additionally, we assume that the respondents have different tasks for machine learning models.

Second, we selected 7 challenges out of 11 challenges that can connect with machine learning model monitoring from the Microsoft study[3]. Note that only one of the remaining Microsoft challenges is directly related to monitoring, that some challenges are hard to connect with machine learning model monitoring, and that other challenges can integrate into one challenge like Education. The chosen challenges are presented in Table 5.1.

There are two interesting things to notice. First, most respondents in all experience groups rank End-to-end pipeline support topmost. When we look at the rank in Table 5.1,

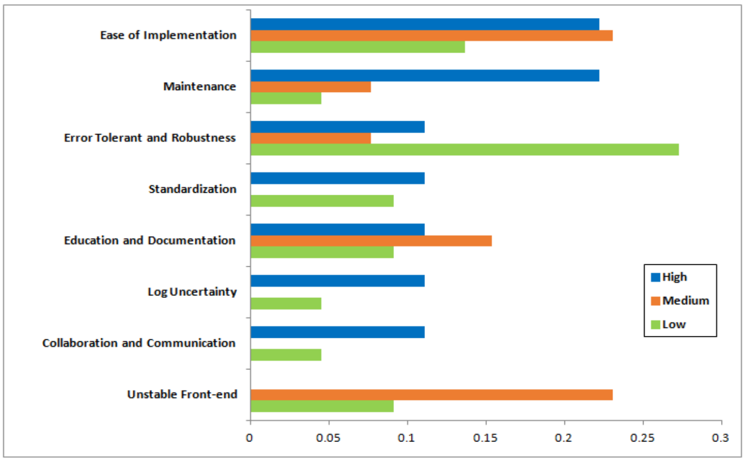


Figure 5.12: Ratio of Responses of Challenges for each Experience Group.

The results depict the ratio of the challenges for each experience group. The histogram is ordered by the High experience group ratio outcome in decreasing order.

four challenges are similarly ranked medium by all experience groups: 1) Data Availability, Collection, Cleaning, and Management, 2) Scale, 3) Specification, and 4) Education and Training. On the contrary, two challenges (Model Evolution, Evaluation, and Deployment, Collaboration and working culture) show different rankings, depending on the experience of respondents with machine learning model monitoring. Low and medium experienced respondees rank Model Evolution, Evaluation, and Deployment as an important challenge, while high experienced respondents did not. Collaboration and working culture show the opposite tendency. These results indicate that the user experience in machine learning model monitoring can affect the user perception of the importance of challenges.

Additionally, we also differentiate the frequency of each challenge with three experience groups. A few things are interesting to see. First, although some challenges are ranked similarly by all respondees, the frequency is different for each experience group. For instance, even though all respondents rank End-to-end pipeline support as the highest, the medium experienced respondents did not rank it as important as low experienced participants(-14%) and high experienced respondees(-4%). Second, high experienced users show a different tendency in some challenges. For example, when looking at Collaboration and working culture or Model Evolution, Evaluation, and Deployment at Table 5.1, the frequency of these challenges from high experienced respondents represents a significant difference compared to other experienced participants.

ING Monithor solution features for an effective model management

In this analysis, we still use the same experience groups: Low, Medium, High. We listed 7 challenges in the survey and asked participants to rank up to 3 challenges. Table 5.2 shows the Ranking and Frequency for each experience group. When we look at the column High

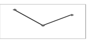
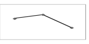
	Frequency			Rank		
				Experience		
Challenge	Medium vs. Low	High vs. Low	Trend	Low	Medium	High
End-to-end pipeline support	-14%	-4%		1	1	1
Collaboration and working culture	-3%	38%		7	7	2
Scale	11%	14%		4	3	3
Data Availability, Collection, Cleaning, and Management	-6%	-14%		2	4	4
Specification	-7%	-3%		5	6	5
Education and Training	10%	-3%		6	5	6
Model Evolution, Evaluation, and Deployment	11%	-26%		3	1	7

Table 5.1: The Top-Ranked Challenges and Personal Experience with ML Model Monitoring.

Respondents were grouped into three categories (Low, Medium, High). The column Frequency depicts the increase or decrease of the frequency in the Medium and High categories compared to the Low category. The column Trend shows the tendency of frequency within experience groups visually, and the column Rank demonstrates the ranking of the challenges within each experience group (the order is increasing order, 1 is the most frequent challenge). The results are sorted by High experience group ranking.

at the rank column, since a few respondents are involved in a high experience group, two rows are blank. Low and medium experienced users rank Synchronization and comparison of the distribution of data as the most useful features that help their work effectively.

On the contrary, high experienced respondents rank this feature as the 5th most important feature. Furthermore, high experienced respondents ranked two features as the most important: 1) Track user-defined metrics, and 2) Thresholds for metrics. We argue that tasks of high experienced users are related to dealing with metrics. On the other hand, low and medium experienced participants work on data processing.

Ease of use

The remaining survey questions consist of the Likert-scale questions. The respondents indicated their level of agreement or disagreement towards five statements (questions Q3.3 through Q3.7 in the survey). They did so on 1 to 5 point Likert-scales (strongly disagree - disagree - neutral - agree - strongly agree), or resorted to an ”I do not know” option if they were unsure whether the aspect mentioned in the question. The spread of scores for each question is depicted in Table 5.3 by descriptive statistics and bar-charts. In order to comprehend any reasons behind the survey scores, we coded the free format text resulting from the survey. The analyses of the survey scores and the text coding resulted in the following notes.

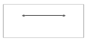
	Frequency			Rank		
				Experience		
ING Monithor solution Features	Medium vs. Low	High vs. Low	Trend	Low	Medium	High
Track user-defined metrics	-1%	7%		2	2	1
Thresholds for metrics	-4%	11%		3	4	1
Track the progress of the running data science pipeline	7%	9%		5	2	3
Search and print the log messages of the data science pipeline	3%	13%		7	6	4
Synchronization and comparison of the distribution of data	-6%	-23%		1	1	5
Overview of Multiple Models	0%	-		4	5	-
Manage the Projects by Tags	1%	-		6	6	-

Table 5.2: The Top-Ranked ING Monithor solution features perceived by the respondents.

Participants were divided into three groups regarding their experience level. The column Rank represents the ranking of the ING Monithor solution features that helped effectively support the model management most. This result shows the perceived ranking by contributors to the survey. The results are sorted by High experience group ranking.

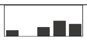
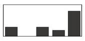
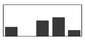
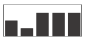
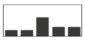
Survey Question	Likert Distribution	Number of Respondents	Percent Agree	Top-Box	Percent Disagree	CV
Q3.6. The ING monitoring solution supports trust in the automated model creation process		14	65%	29%	14%	36%
Q3.7. The ING monitoring solution is an essential tool for developing and maintaining machine learning models		16	63%	50%	19%	41%
Q3.4. The ING monitoring solution makes managing my teams models efficient		16	51%	13%	19%	39%
Q3.5. The ING monitoring solution increases the quality of my teams models		12	50%	25%	25%	43%
Q3.3. The ING monitoring solution makes managing my team's models easy		16	38%	19%	26%	40%

Table 5.3: Overview of the Survey Results.

The table is sorted on the percentage agreed. Column 'Likert Distribution' depicts a graph of the distribution of a 1-5 point Likert scale for each question with from left to right the value 'Strongly Disagree', 'Disagree', 'Neutral', 'Agree', 'Strongly Agree.' The Top-Box demonstrates the percentage of respondents that strongly agreed. The CV means Coefficient of Variation(CV)[1], also known as relative standard deviation. The CV illustrates the standard deviation divided by the mean. Higher CV-values indicate a higher variability.

As Table 5.3 depicts an overview of the survey results, Q3.3 represents the ease of use for the ING Monithor solution. A majority of respondents neither agree nor disagree with the statement. The percentage of disagree subtracted from the agree is 12%. Even though the respondents slightly agree on the statement, the ratio does not represent a big difference. One respondent who strongly agreed on this question mentioned: "I can instantly know which models run into production." On the other hand, the participants who disagreed stated: ”The front-end has performance issues,” and ”We cannot monitor data distributions.” These reasons were noted in the challenges collected by respondents.

Efficiency of use

Even though the ratio of the Top-Box on Q3.4 is the lowest(13%) in Table 5.3, respondents are relatively in agreement with the remark(51%). Therefore, the ING Monithor solution is perceived as an efficient tool in terms of managing machine learning models, as some respondents mentioned: ”Once the system is set up, it is always ready to view at any time of the day” and ”Easily find out where any error or issues occurred in production by not having to communicate to the other teams and investigate.”.

Model quality

As we received more ”I do not know” answers on Q3.5 (5 in total), the number of respondents on Q3.5 is the smallest. We argue that the respondees are less familiar with this aspect. Although the lowest number of participants stated on this question, half of them agreed. However, since this survey question shows the largest coefficient of variation compared to other statements, the answers of respondents can be unsteady.

Trust of the automated model creation

As Table 5.3 shows, the highest number of respondents agreed with the statement expressed in Q3.6 (65%). Additionally, the coefficient of variation and the percentage of disagree represent the smallest result. Therefore, we argue that most respondees perceived the ING Monithor solution as supporting the trust in the automated model creation process. As one of the respondents put it: ”Tracking the log helps us to trust the automation process.”

Importance of ING Monithor solution

As Table 5.3 depicts, there are some interesting things to notice for Q3.7. First, the percentage of agreement is 63%, while the percentage of strong agreement is 50%. Therefore, a relatively high number of respondees perceived the ING Monithor solution as an essential tool. The participants who agreed mentioned: ”It is one of the necessary things in machine learning models” and ”The ING Monithor solution achieves not only investigation purposes but also developing or debugging any issues efficiently.” Second, even though some participants disagreed with the statement, they still agree that monitoring the machine learning model is essential. Therefore, most participants perceived that the monitoring system for any machine learning model is critical.

5.3 Research Question 1 : Model Usage

RQ1: To what extent can the monitoring system support stakeholders in terms of maintaining the machine learning models?

Our results as presented in Section 5.2.2, indicate that with regards to model usage, the ING Monithor solution supports efficient model management based on our collected data analysis and quantitative survey questions results.

First, the total number of ING Monithor solution users is 28, while the total number of deployed models is 19. However, the included number of models in the work scope is different for each user. The sum of all involved models for each user is 65. If we numerically divide them by 17 respondents, each user has approximately 4 models. Therefore, we argue that most users have some overlapping models in their scope.

Second, we calculated the model reviewing time for one model per person shown in Table 5.4. When we compute an average of the model reviewing time for one model (sum of the model reviewing time per model/number of respondents), it is approximately 79 seconds. The result considered the time and user numerically, but did not take into account the background or the knowledge level of respondents.

Additionally, we calculated the man-hour spent on managing models in a month. We computed an average of the man-hour (sum of the man-hour spent in a month/number of respondents). Accordingly, one user spent about 19 minutes managing their models in a month.

Number of models	Frequency of the ING Monithor solution usage	Model Reviewing Time		Man-Hour Spent in a month (mins)
		Total (mins)	Per Model (secs)	
12	Less than Once in a week	30	150	60
6	Less than Once in a week	1	10	2
6	Less than Once in a week	5	50	10
6	Less than Once in a week	5	50	10
6	4-6 times in a week	1	10	24
6	Less than Once in a week	1	10	2
3	Once in a week	5	100	20
3	Less than Once in a week	5	100	10
3	Once in a week	1	20	4
3	2-3 times in a week	5	100	60
3	Once in a week	5	100	20
3	Once in a week	5	100	20
1	2-3 times in a week	1	60	12
1	Less than Once in a week	1	60	2
1	Less than Once in a week	1	60	2
1	Less than Once in a week	1	60	2
1	2-3 times in a week	5	300	60

Table 5.4: Raw Quantitative Data.

We decided to set the number of models and frequency of the ING Monithor solution usage to the maximum value. For example, when the user selects the 2-3 models option, then the value is set to 3. Plus, when the user chooses the ”Less than Once in a week” option, we consider this as once in two weeks, as some users stated. Respondents select the column Total at the Model Reviewing Time, and the column per model is calculated (total time/number of models * 60 seconds). The column Man-Hour Spent in a month represents how much time the respondents spent in order to manage their models. The table is sorted on the number of models in decreasing order.

Lastly, the highest number of model usage count was 23142. The usage of models is raised by increasing the number of users and models. When we numerically calculate the model usage per person, it is 826.5 each.

Consequently, since managing one model takes around 1 minute, we assume that the ING Monithor solution provides fast model management. When we look into the average of man-hour spent in a month, the users only spend 19 minutes in a month in order to maintain the models, which is not too costly. Moreover, since the model usage increases even though new models were not added, we argue that the ING Monithor solution helps to increase model usage, as users are able to quickly check their model validation and evaluation.

5.4 Research Question 2 : User Perception

RQ2: How do users of a monitoring system perceive the way it supports the models with regard to usability, quality and trust?

The second research question is answered by qualitative questions in the survey described in Section 3.4.2. Moreover, the results of the qualitative questions are illustrated in Sections 5.2.3. We argue that respondents perceived that the ING Monithor solution supports machine learning models regarding usability, quality, and trust of the automated model creation.

First, half of the respondents perceived that the ING Monithor solution increases the model quality, even though the result shows high coefficient of variation value, which means that higher variability.

Second, the result regarding the trust of the automated model creation showed the highest agreement from the respondents (65%). Therefore, we insist that the participants perceived that the ING Monithor solution helps to build the trust of the automated model creation process with checking logs.

Last, since usability is composed of effectiveness, efficiency, and satisfaction in the ISO 9241 standard[24], we take into account these three factors. As the respondents noted that some ING Monithor solution features such as ”Synchronization and comparison of the distribution of data”, ”Track user-defined metrics”, and ”Thresholds for metrics”, provide useful model management capabilities as shown in Figure 5.2, we argue that respondents perceived the ING Monithor solution helps the machine learning models management effectively. Moreover, because 51% of the respondents mentioned that they agreed the ING Monithor solution makes managing models more efficient, we conclude that the users perceived that the ING Monithor solution supports efficient model management. Even though the users did not state that the ING Monithor solution is easy to use (most answers were Neutral), the majority of respondents perceived that the ING Monithor solution is essential. Moreover, the users perceived that the ING Monithor solution offers effective and efficient model management, as well as, that it increases the model quality and trust of automated model creation. Therefore, we argue that the participants are satisfied with the ING Monithor solution.

Consequently, we conclude that respondents perceived that the ING Monithor solution supports the models concerning usability, quality, and trust, and recognized the importance of the monitoring system.

Chapter 6

Related Work

A monitoring system for machine learning is a well-explored domain in computer science. We describe technologies related to a monitoring system for an automated model creation system. This chapter is divided into three lines of relevant research work: 1) Software Engineering for Machine Learning, 2) Machine Learning and Data Management, and 3) Existing Monitoring Solutions. We link them with our tasks and discuss the novelty of our work as well.

6.1 Software Engineering for Machine Learning

Since the newest trend to hit the software production field is about combining artificial intelligence (AI) capacities based on advances in machine learning, Amershi et al.[3] set up a study to see how Microsoft software teams develop software applications with customer-focused AI features based on nine-stage workflow. Plus, Amershi et al. discussed three core differences between machine learning-centric components and traditional application domains of software engineering. Lastly, Amershi et al. talked about the machine learning process maturity model for evaluating the progress of software teams towards excellence in building AI applications, and a set of best practices for building applications depending on machine learning.

In order to accomplish these studies mentioned above, Amershi et al. did interviews to gather the essential topics and conducted a survey based on the results of the interviews. Amershi et al. found some of the essential challenges associated with building large-scale machine learning applications and platforms and how the teams solve challenges in their products. Amershi et al. separated the respondents into 3 groups depending on the number of years of AI experience: Low, Medium, High. According to the respondents AI experience, the challenges are ranked variously.

Amershi et al. identified three main differences in building applications for machine learning models compared to the traditional application domain, from the interviews and the survey. First, the behavior of a machine learning system relies heavily on data. Since the machine learning iteration runs at a fast pace, the data schema changes regularly. It takes more effort to discover, source, manage, and version data. Second, machine learning demands in-depth knowledge to build, train, and evaluate models. Sometimes the model requires to redo all these works again. Therefore, the machine learning model requires more diverse skills and more profound knowledge than traditional domains. Third, as machine learning models are not easily extensible, and models interact in non-obvious ways, machine learning models are harder to handle. Each team manages several modules, and the team needs to communicate with the other teams in order to make their modules working with the modules of other teams.

DevOps Capabilities, Practices, and Challenges

DevOps employ continuous software development processes and support an agile software development lifecycle. Even though DevOps has recently been used in a variety of software development domains, the current study on the actual implementation and practices of DevOps is not sufficient enough. Mali et al.[47] describe empirical research into factors influencing its implementation based on an in-depth exploratory case study with interviews. Four main findings are described in this study: 1) Having a consistent understanding of the meaning of the term DevOps within an organization is essential in order to discard all misunderstandings, goal misalignment, and missed benefits, 2) Drivers and realized benefits of an improved frequency of releases and improved application quality, 3) The selection of the technical enablers was a complex, process taking much effort and resources, and 4) Identified a number of challenges that have been studied in the literature such as lack of clear definition, insufficient communication, deep-seated company culture, organization structure, and geographical distribution while finding some new challenges.

Hidden Technical Debt

Even though machine learning offers a critical toolkit for building complex prediction systems fast, David et al.[46] argue that machine learning systems have a unique capacity for incurring technical debt because of having the maintenance problems of traditional code and machine learning-specific issues. This study[46] includes several types of debts: 1) Complex Models Erode Boundaries, 2) Data Dependencies, 3) Feedback Loops, 4) ML-System Anti-Patterns, 5) Configuration Debt, 6) Changes in the external world, and 7) Additional ML-related Debt. David et al. describe several technical debts in each category, and they argue that the most important insight is technical debt awareness for engineers and researchers. Since the full cost of debt becomes apparent only over time, even a few changes of data dependencies can slow further progress. David et al. insist that addressing debts can be fulfilled by team culture changes.

Software Engineering Patterns

Machine learning techniques have been widely used in many domains. The techniques are dependent on mathematics and software engineering in terms of algorithms, implementations, and performances. Even though many researchers study best practices for designing machine learning systems and software in order to address the software complexity and quality of the techniques, collecting, classifying, and discussing software engineering design patterns for machine learning techniques are missing. In order to address these, Hironori et al.[57] collect software engineering design patterns, conduct the survey, and report the results regarding preliminary results of a systematic literature review of good/bad design patterns for machine learning. This study answered four research questions: 1) Software engineering developers consider the complexity of machine learning systems and lack of knowledge of architecture and design patterns that could help them, 2) Many gray documents discuss good/bad practices of machine learning systems design more than academic literature, 3) Software engineering patterns are classified into two dimensions: ML pipeline and SE development process, and 4) Some patterns apply to many stages of the pipeline, some to many phases of the development process.

6.2 Machine Learning and Data Management

Since an analysis of large datasets using statistical machine learning algorithms is fundamental to modern data-driven applications in many domains, the data management research community has built several systems for data analytics[28]. Kumar et al.[28] provide an in-depth review of systems and techniques that handle the data management challenges in the machine learning workloads context. They mainly focused on recognizing and examining the technical challenges and general data management difficulties in order to offer techniques and open issues to help identify systems. Besides, Kumar et al. centered on describing the key ideas, architecture, strengths, and shortcomings of primary systems that address these difficulties. Three lines of work are covered by the study of Kumar et al.: 1) reviewing the systems and frameworks that combined machine learning algorithms, systems, and languages with existing data systems, 2) modifying data management techniques to the new machine learning workloads, and 3) merging data management and machine learning ideas in order to enhance the machine learning life-cycle tasks and the performance of algorithms.

Model Selection

Recently, there is increasing interest in managing critical artifacts of the machine learning process, such as machine learning models. The data model management proposed several data management platforms in order to implement machine learning techniques efficiently[27]. The study of Kumar et al.[27] describes that a model selection is composed of three iterative tasks: 1) Feature engineering, 2) Algorithm selection, and 3) Parameter tuning. In order to make these three tasks efficiently, Kumar et al. propose model selection management systems. The systems repurpose three key ideas: 1) Steering, 2) Consumption, and 3) Execution. The systems are able to reduce the number of iterations and the time per iteration by improving these three fundamental notions.

Data Integration

As the amount of data has increased a lot, we have to analyze more than before. Dong et al.[9] observe that the rise of machine learning and deep learning techniques for data integration tasks in order to utilize the data efficiently and use machine learning to be sufficient.

Dong et al. indicate that machine learning improves the accuracy of the models and reduce costs on data integration. The study of Dong et al. describes the synergy from the perspective of data integration, and how machine learning has been reshaping different tasks of data integration. Dong et al. also illustrate the difference of the techniques between past and present, and how the technique shifting has influenced data integration results. Additionally, they review the impact of the data integration methods and describe open challenges and further opportunities to be more productive: 1) Multi-modal, 2)Fast and Cheap Training Data, 3)Human-in-the-Loop, 4)Efficient Model Serving, 5)Declarative Interfaces, and 6) Effective Data Augmentation.

Machine Learning Model Management

The performance of machine learning models depends on the underlying data. When incoming data is scored against the model statistically deviates from the data where the model is trained, the performance of the model worsens, it may lead to the models are invalid. In order to prevent model performance degradation, tracking the statistics of the model performance is needed. The machine learning model management rises to become more critical.

Building a machine learning model is a trial-and-error based repetitive process in an enterprise context[54]. A model is built based on a hypothesis about the underlying data, and then the model is trained and tested. After this, the results redefined the hypothesis, and then the refined hypothesis revised the model. The machine learning model process is based on the development of tens or hundreds of machine learning models before landing at a model that can be accepted. Tracking a previously built machine learning model and the corresponding insights is difficult. Therefore, there is a need to remember relevant information about previous models to tune the next set of machine learning models. Lack of model and result persistence can also lead to uncertainty on the conclusions of a previous experiment leading to re-running of expensive modeling workflows. This iterative, ad-hoc nature of machine learning model building gives rise to the importance of machine learning model management.

Deep Learning Life-cycle Management

Since deep neural network models are learned using massive amounts of training data, there are many critical large-scale data management issues in learning, storing, sharing, and using deep learning models[33, 34]. The deep learning features are not given by a human but are learned automatically based on the input data. Plus, the features are complex and have a hierarchy along with the network representation, which makes it hard to understand and explain the learned model. In order to address these issues, Miao et al.[33] propose *ModelHub*, which contains five key components: 1) Novel model versioning system, 2) Domain-specific language, 3) Model learning module, 4) Provenance management system, and 5) Hosted service. *ModelHub* is an end-to-end system that introduces new high-level abstractions proper for automating deep learning workflows, and shows the way of an efficient abstraction implementation in order to manage and optimize the different steps in the workflow.

6.3 Existing Monitoring Solutions

MLflow

MLflow[35] is an open-source platform for managing the machine learning life-cycle. *MLflow* is a flexible tool, which works with any machine learning libraries, algorithms, and programming languages. It operates via a REST API(Representational State Transfer Application Programming Interface) and simple data formats. *MLflow* is composed of three components: *MLflow tracking*, *MLflow projects*, and *MLflow models*. *MLflow tracking* is an API and a UI(User Interface), which can be utilized to configure log parameters, version control, metrics, and output files for machine learning code and visualization. *MLflow projects* are used to give a format for offering a reproducible and reusable way to execute the workflows. *MLflow models* are applied to give a standard format for wrapping machine learning models in various models. *MLflow* is able to track the performance of the machine learning models with predefined features.

Polyaxon

Polyaxon[39] is an open-source platform, which provides machine learning life-cycle management. *Polyaxon* works with all major deep learning frameworks such as TensorFlow, MXNet, Caffe, Torch, Keras, which are well-known and used in many fields. *Polyaxon* is a user-friendly tool that can be customized and configured by the requirements of the users. Additionally, *Polyaxon* provides an API, which is developer-friendly and allows tracking code, parameters, metrics, and logs. *Polyaxon* exposes hyperparameters, which help to build the optimized models automatically to ensure performance, scalability, and reproducible. In *Polyaxon*, there are built-in features to track the performance of the machine learning models. Moreover, *Polyaxon* grants a dashboard that includes a list of projects and experiments, visualized results, and logs for experiments.

Prometheus

The Uber company[53] has a monitoring system for machine learning models that they owned. The Uber company decided to use *Prometheus*[41], which is a popular monitoring and alerting solution. In order to use this open-source solution properly, the Uber built *M3*[52], which is a remote storage for *Prometheus*. Many companies and organizations have selected *Prometheus*. It is built as an independent tool that is not affected by storage or services for reliability. Plus, *Prometheus* has an alert function. Thus, when errors are diagnosed during runtime, users can react and treat the errors as fast as possible. *Prometheus* works with any solely numeric time series, and multi-dimensional data model and queries. However, *Prometheus* is limited in scalability and durability due to a single node. *Prometheus* provides a visualization feature by Grafana[17] or console templates.

Chapter 7

Discussion

In this chapter, we are going to compare our results to the related work. After that, we discuss the implications of our study.

7.1 Challenges Comparison

Since we used a list of challenges from the Microsoft study[3], it is worth noticing the ranks and comparing the results. The challenges between our study and the Microsoft study are different due to topic differences. Our challenges are about machine learning model monitoring, and the Microsoft challenges are regarding machine learning. Therefore, since the comparison has been made by ranking only, we did not consider the external factor differences such as topic, an experience level of the survey respondents, backgrounds of the respondees, working environments. The result is shown in Table 7.1.

	Our Rank			Microsoft Rank		
	Experience			Experience		
Challenge	Low	Medium	High	Low	Medium	High
End-to-end pipeline support	1	1	1	4	2	4
Collaboration and working culture	7	7	2	5	6	6
Scale	4	3	3	10	4	3
Data Availability, Collection, Cleaning, and Management	2	4	4	1	1	1
Specification	5	6	5	5	8	8
Education and Training	6	5	6	1	5	9
Model Evolution, Evaluation, and Deployment	3	1	7	15	6	4

Table 7.1: The Top-Ranked Challenges Comparison and Personal Experience with ML and ML Model Monitoring.

The results are divided into three experience groups same as in Chapter 5. However, as we mentioned, the ratio of the experience level groups is different between our study and the Microsoft study[3]. The table is sorted by a High experience group ranking on Our Rank.

When we look at Table 7.1, three noteworthy things can be found. First, a few challenges are similarly ranked depending on the experience of respondents. For example, Scale is ranked similarly by medium and high experienced respondents both in our study and the Microsoft study[3]. Additionally, Collaboration and working culture depict a comparable tendency. Second, two challenges are relatively ranked high in both studies: 1) End-to-end pipeline support, and 2) Data Availability, Collection, Cleaning, and Management represent. These challenges are still highly ranked, even though these challenges show the opposite result in both studies. For instance, End-to-end pipeline support is ranked at the top by all respondents in our study, but this challenge is ranked 2nd or 4th in the Microsoft study.

Moreover, Data Availability, Collection, Cleaning, and Management represents the opposite case. Therefore, these two challenges are considered relatively important for most respondents regardless of the experience of users. Third, some challenges are ranked differently in both studies. For example, Model Evolution, Evaluation, and Deployment show quite considerable gaps in the rank of low experienced respondents. This challenge is ranked 3rd in our study; on the contrary, it is ranked 15th in the Microsoft study. We insist that the reasons for this gap can be their role or background. Also, we argue that the topic difference can be one of the reasons because the topic of the study is different in both our study and the Microsoft study.

7.2 Comparison with Existing Solutions

As there are some open-source monitoring systems for machine learning models and lifecycle management, it is interesting to compare our system with them. As we mentioned in the previous chapter, the existing solutions such as MLflow[35], Polyaxon[39], and Prometheus[41], are chosen by our own judgments, based on the criteria that 1) the solution is an open-source system, 2) it covers the major deep learning frameworks, and 3) it provides visualization features. The comparison results are shown in Table 7.2.

	Solutions			
Feature	ING Monithor solution	MLflow	Polyaxon	Prometheus
Dataset Distribution Synchronization	O	O	O	X
Logging Data to Runs	O	O	O	O
Log and model Tracking	O	O	O	O
Model Customization	O	O	X	X
Model Management	O	O	O	X
Visualization	O	O	O	O
Version control	O	O	O	X
Scalability	X	X	O	X
Alerting	X	X	X	O

Table 7.2: Feature Comparison with Existing Solutions.

The table represents the comparison of major features between existing solutions and the ING Monithor solution. All features are taken from existing solution websites[35, 39, 41]. ”O” means the solution includes the feature, ”X” indicates the solution did not have the feature.

When we look at Table 7.2, the ING Monithor solution offers 7 out of 9 features; it does not support Alerting and Scalability. Because the MLflow solution only provides Scalability in a commercial version, we did note in the overview that this feature is not supported. On the other hand, Polyaxon offers a Scalability feature by having a built-in optimization engine. The engine chooses the best parameter to have the best results with robust scheduling compared to the other solutions. However, Polyaxon does not support Model Customization. The Prometheus solution seems somewhat weaker than other solutions because many features are missing.

Consequently, since the ING Monithor solution is a self-built system, it suits well with the ING environments and the ING Model Factory. The ING Monithor solution offers many essential features for machine learning model management compared to the open-source systems we researched. Furthermore, the ING Monithor offers the possibility to add new features. Therefore, we argue that in the short term, the ING Monithor solution offers the best opportunities. However, if we look into the long term, the open-source systems may give better opportunities in terms of ensuring the reliability of the program, and engaging external developers to try and improve the software in more environments[37]. This long term perspective requires further research.

7.3 Further Research

Our study illustrates that a monitoring system for machine learning models is critical in 3 aspects: 1) maintaining models, 2) checking the model creation process, and 3) user perceptions on machine learning models and automated model creation process. These are more pronounced in large environments working and engaging with hundreds of software development teams. We identify 3 core challenges that require further attention, both in research and in practice, in order to move the machine learning model management forward.

Trust improvements. In our study, we observed that users perceived that a monitoring system supports the trust in an automated model creation process by tracking logs. It means that the monitoring system is one of the factors that provides trust to the users. However, even though the model creation process is fully automated, the users still want to participate in the model creation process by checking logs. Also, the users note the logs to the monitoring system. There is a need for more insight into the characteristics of this dependency, since this trust issue can apply to the outcomes. How can the users trust the outcome of the models created by the automated system?. How can the users fully trust the automated model creation process without user dependency?. What factors can increase the trust of the automated model creation process? These questions may lead to better user perception of the automated model creation process.

Ease of use. We discovered that even though users perceived that a monitoring system increases the model quality, and offers efficient and relatively fast model management, the ING Monithor solution does not make managing models easy. In order to know and validate best practices and other factors that influence the ease of use, further research is required.

Model reviewing time. We found that the number of models does not significantly influence model reviewing time. This finding suggests that other factors, namely machine learning model algorithms, or the complexity of models, play a role in machine learning model management. An investigation of factors that affect the model reviewing time could help the field to understand better when (and why) model reviewing time increases.

7.4 Threats to Validity

Although this research yields interesting results, it should be considered against potential limitations.

Internal Validity. Since the ING Model Factory and the ING Monithor solution were deployed recently, the collected data is sparse. Moreover, the collected data does not provide enough relevant information concerning the impact and importance of the ING Monithor solution on machine learning model management, because only the number of models and the usage of models can be collected. In order to reduce these risks of data collection, we performed a survey, and we combined the collected data and survey results.

In our survey design, we phrased and ordered the questions to avoid leading questions and order effects. To avoid the latter, we did not randomize the questions, but we randomized the options at the ranking questions. The respondents may have been affected by other respondents. For example, respondents answer questions in a way that is viewed favorably by others. To mitigate this risk, we made the survey anonymous and informed the participants that the quantitative results would be analyzed statistically.

External Validity. Our case study was conducted with teams at ING, a large global banking company that develops software solutions in-house. Some findings can be restricted to the ING teams and team members who contributed to our surveys. Moreover, because the survey results rely on ING Monithor solution users who participate in our survey, the results can be different depending on respondents' background, the experience level of the machine learning models and the machine learning models being monitored, and the job role at ING. Plus, the number of respondees is relatively small, so we cannot generalize our conclusions to other organizations. In order to reach a general conclusion, this work needs to be performed in other organizations or environments.

Chapter 8

Conclusions

In this chapter, we summarize the findings from our study. After this overview, we will draw some conclusions.

8.1 Overview

The ING data scientists put much effort into developing an automated machine learning creation system (the ING Model Factory) and a monitoring system for machine learning models created by the automated system (the ING Monithor solution). After an automated machine learning framework was released like the ING Model Factory, a monitoring system became a more important factor in machine learning. This thesis describes the ING Model Factory in which the machine learning models are created by data scientists and the ING Monithor solution that supports the monitoring of these models. After that, the thesis illustrates the new ING Monithor solution features based on user requirements. Furthermore, the thesis explains three perceived perceptions by the users of the ING Monithor solution: 1) challenges of machine learning models monitoring, 2) the impact of the monitoring system, and 3) the importance of monitoring systems for machine learning models. Lastly, this thesis depicts a comparison of the challenges, as well as a comparison between the ING Monithor solution and three existing open-source solutions.

8.2 Conclusions

The goal of our study is to examine the importance and the impact of the ING Monithor solution on the machine learning model management. To that end, we collected data and performed a mixed case study. The key findings of this study are:

First, we described the challenges of the ING Monithor solution as perceived by its users. Furthermore, we compared the results of our study with the outcomes of a Microsoft study [3] on the same topic. We found that some of the issues were related to the experience of users with machine learning models monitoring.

Second, we illustrated the impact and importance of the monitoring system. We discovered that the ING Monithor solution provides a relatively fast average reviewing time per model. Plus, users of the ING Monithor solution perceived that it supports the models with regard to quality, the trust of the automated model creation with checking logs, and usability. Additionally, the monitoring system for a machine learning model is perceived by the users as a critical role in model management.

Finally, we identified the differences in the existing solution in terms of features for model management. Even though the ING Monithor solution does not support the features Scalability and Alerting, as some of the compared solutions do, still, we argue that the ING Monithor solution supports sufficient model management. However, as the ING Monithor solution is deployed recently, more research will require to see the impact of the ING Monithor solution.

Bibliography

[1] Hervé Abdi. Coefficient of variation. *Encyclopedia of research design*, 1:169–171, 2010.

[2] Azza Abouzeid, Kamil Bajda-Pawlikowski, Daniel Abadi, Avi Silberschatz, and Alexander Rasin. HadoopDB: an architectural hybrid of MapReduce and DBMS technologies for analytical workloads. *Proceedings of the VLDB Endowment*, 2(1): 922–933, 2009.

[3] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. Software engineering for machine learning: a case study. In *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice*, pages 291–300. IEEE Press, 2019.

[4] Michael Armbrust, Reynold S Xin, Cheng Lian, Yin Huai, Davies Liu, Joseph K Bradley, Xiangrui Meng, Tomer Kaftan, Michael J Franklin, Ali Ghodsi, et al. Spark SQL: Relational data processing in spark. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 1383–1394. ACM, 2015.

[5] Bootstrap. Bootstrap, An open source toolkit for developing with HTML, CSS, and JS. URL `https://getbootstrap.com/`.

[6] Dhruba Borthakur et al. HDFS architecture guide. *Hadoop Apache Project*, 53(1-13): 2, 2008.

[7] John W Creswell and J David Creswell. *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 2017.

[8] D3.js. D3 Data-Driven Documents. URL `https://d3js.org/`.

[9] Xin Luna Dong and Theodoros Rekatsinas. Data integration and machine learning: A natural synergy. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1645–1650. ACM, 2018.

[10] Andrej Dyck, Ralf Penners, and Horst Lichter. Towards definitions for release engineering and DevOps. In *2015 IEEE/ACM 3rd International Workshop on Release Engineering*, pages 3–3. IEEE, 2015.

[11] Elastic. A tool to collect, process, and forward events and log messages. URL `https://www.elastic.co/products/logstash`.

[12] Radwa Elshawi, Mohamed Maher, and Sherif Sakr. Automated Machine Learning: State-of-The-Art and Open Challenges. *arXiv preprint arXiv:1906.02287*, 2019.

[13] Egboro Felix. Marketing challenges of satisfying consumers changing expectations and preferences in a competitive market. *International Journal of Marketing Studies*, 7(5):41, 2015.

[14] Flask. A Python Micro web framework. URL `http://flask.pocoo.org/`.

[15] Uwe Flick. *An introduction to qualitative research*. Sage Publications Limited, 2018.

[16] Erik Frøkjær, Morten Hertzum, and Kasper Hornbæk. Measuring usability: are effectiveness, efficiency, and satisfaction really correlated? In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 345–352. ACM, 2000.

[17] Grafana. The open platform for analytics and monitoring. URL `https://grafana.com/`.

[18] Kathy E Green and Tony CM Lam. Is Neutral on a Likert Scale The Same As ”Dont Know” for Informed and Uninformed Respondents? Effects of Serial Position and Labeling on Selection of Response Options, University of Toronto and University of Denver.

[19] James Hamilton, Manuel Gonzalez Berges, Jean-Charles Tournier, and Brad Schofield. SCADA Statistics monitoring using the elastic stack (Elasticsearch, Logstash, Kibana). 2018.

[20] Kevin Anthony Hoff and Masooda Bashir. Trust in automation: Integrating empirical evidence on factors that influence trust. *Human factors*, 57(3):407–434, 2015.

[21] IBM. IBM PureData System for Analytics. URL `https://www.ibm.com/support/knowledgecenter/en/SSULQD_7.2.1/com.ibm.nz.welcome.doc/kc_welcome_V721.html`.

[22] IBM Cognos Analytics. An AI-fueled business intelligence platform. URL `https://www.ibm.com/products/cognos-analytics`.

[23] ING. 2018 Annual Report ING Group N.V., 2019, March. URL `https://www.ing.com/About-us/Annual-reporting-suite/Annual-Report/2018-Annual-Report.htm`.

[24] ISO. 9241-11. Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11: Guidance on usability. *The international organization for standardization*, 1998.

[25] Jupyter. The Jupyter Notebook. URL `https://jupyter.org`.

[26] Elvan Kula, Ayushi Rastogi, Hennie Huijgens, Arie van Deursen, and Georgios Gousios. Releasing fast and slow: an exploratory case study at ING. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 785–795. ACM, 2019.

[27] Arun Kumar, Robert McCann, Jeffrey Naughton, and Jignesh M Patel. Model selection management systems: The next frontier of advanced analytics. *ACM SIGMOD Record*, 44(4):17–22, 2016.

[28] Arun Kumar, Matthias Boehm, and Jun Yang. Data management in machine learning: Challenges, techniques, and systems. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1717–1722. ACM, 2017.

[29] John D Lee and Katrina A See. Trust in automation: Designing for appropriate reliance. *Human factors*, 46(1):50–80, 2004.

[30] Rensis Likert. A technique for the measurement of attitudes. *Archives of psychology*, 1932.

[31] Pierre Lison. An introduction to machine learning, an early draft of a proposed textbook. robotics laboratory, department of computer science, stanford university, 2015.

[32] Wes McKinney and PD Team. A Python data analysis toolkit. *Pandas Powerful Python Data Analysis Toolkit*, page 1625, 2015.

[33] Hui Miao, Ang Li, Larry S Davis, and Amol Deshpande. Modelhub: Deep learning lifecycle management. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 1393–1394. IEEE, 2017.

[34] Hui Miao, Ang Li, Larry S Davis, and Amol Deshpande. Towards unified data and lifecycle management for deep learning. In *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pages 571–582. IEEE, 2017.

[35] MLFlow. A platform for the machine learning lifecycle. URL `https://mlflow.org/docs/latest/index.html`.

[36] Besmira Nushi, Ece Kamar, Eric Horvitz, and Donald Kossmann. On human intellect and machine failures: Troubleshooting integrative machine learning systems. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[37] RK Pandey and Vinay Tiwari. Reliability issues in open source software. *International Journal of Computer Applications*, 34(1):34–38, 2011.

[38] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[39] Polyaxon. A platform for the machine learning lifecycle. URL `https://docs.polyaxon.com/concepts/introduction/`.

[40] Neoklis Polyzotis, Sudip Roy, Steven Euijong Whang, and Martin Zinkevich. Data management challenges in production machine learning. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1723–1726. ACM, 2017.

[41] Prometheus. A open-source systems monitoring and alerting toolkit. URL `https://prometheus.io/docs/introduction/overview/`.

[42] Python. Python, A programming language that lets you work quickly and integrate systems more effectively. URL `https://www.python.org/`.

[43] James Roche. Adopting DevOps practices in quality assurance. *Commun. ACM*, 56 (11):38–43, 2013.

[44] Sebastian Schelter, Felix Biessmann, Tim Januschowski, David Salinas, Stephan Seufert, Gyuri Szarvas, Manasi Vartak, Samuel Madden, Hui Miao, Amol Deshpande, et al. On Challenges in Machine Learning Model Management. *IEEE Data Eng. Bull.*, 41(4):5–15, 2018.

[45] Scikit-Learn. Simple and efficient tools for predictive data analysis. URL `https://scikit-learn.org/stable/`.

[46] David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Advances in neural information processing systems*, pages 2503–2511, 2015.

[47] Mali Senapathi, Jim Buchan, and Hady Osman. DevOps capabilities, practices, and challenges: Insights from a case study. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*, pages 57–67. ACM, 2018.

[48] Sanjeev Sharma and Bernie Coyne. DevOps for the Dummies, 2nd IBM Limited Edition, 2015.

[49] Spark. Mllib. URL `https://spark.apache.org/docs/2.2.0/mllib-evaluation-metrics.html`.

[50] Evan R Sparks, Ameet Talwalkar, Daniel Haas, Michael J Franklin, Michael I Jordan, and Tim Kraska. Automating model search for large scale machine learning. In *Proceedings of the Sixth ACM Symposium on Cloud Computing*, pages 368–380. ACM, 2015.

[51] John Spooner. Creating a SAS® model factory using in-database analytics. In *SAS Global Forum*, pages 1–8, 2011.

[52] Uber. M3 platform as a remote storage backend for Prometheus. URL `https://eng.uber.com/m3/`.

[53] Uber. Uber Reports First Quarter 2019 Results. URL `https://investor.uber.com/news-events/news/press-release-details/2019/Uber-Q1-2019-Earnings/`.

[54] Manasi Vartak, Harihar Subramanyam, Wei-En Lee, Srinidhi Viswanathan, Saadiyah Husnoo, Samuel Madden, and Matei Zaharia. Model DB: a system for machine learning model management. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, page 14. ACM, 2016.

[55] Paul Voigt and Axel Von dem Bussche. The EU general data protection regulation (GDPR). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 2017.

[56] Miha Vuk and Tomaz Curk. Roc curve, lift chart and calibration plot. *Advances in methodology and statistic*, 3(1):89, 2006.

[57] Hironori Washizaki, Hiromu Uchida, Foutse Khomh, and Yann-Gaël Guéhéneuc. Studying software engineering patterns for designing machine learning systems. In *2019 10th International Workshop on Empirical Software Engineering in Practice (IWESEP)*, pages 49–495. IEEE, 2019.

Appendix A

Survey Questions

In this appendix section, we give an overview of survey questions that we conducted. If there is no optional selection, the question is an open-ended question. Therefore, people can answer what they have in their mind. There are three rank questions in qualitative questions. The first question is ranking all components of the list, and the rest questions are ranking up to three elements of the list.

A.1 Survey Instruction

This survey is about a monitoring system at ING, in this survey referred to as the ING monitoring solution.

We are conducting research on the monitoring system for machine learning models. We would love to hear from you about the impacts and challenges of the ING monitoring solution, and how you perceive the importance of the monitoring system.

This will help us to understand model usage and quality, as well as user efforts regarding the ING monitoring solution. The survey should only take around 5 minutes, and your responses are reported anonymously. The survey results will be used in a master thesis of TU Delft.

A.2 Part 1: Demographics

1.1 What is your job role in ING?
a. Customer Journey Expert
b. Data Analyst
c. Data Engineer
d. Data Scientist
e. Other

If the answer is selected ”Other”, the following question is asked.
1.1.1 Can you please fill your actual role in ING?

1.2 How many years of experience do you have with machine learning models in general (in- and outside ING)?
a. No Experience
b. 1 year or less
c. 2-3 years
d. 4-5 years
e. 6-7 years
f. 8-9 years
g. 10 years or more

If the answer is selected ”No Experience”, the survey is finished.

1.3 How many years of experience do you have with monitoring machine learning models in general (in- and outside ING)?
a. 1 year or less
b. 2-3 years
c. 4-5 years
d. 6-7 years
e. 8-9 years
f. 10 years or more

1.4 How many years of experience do you have with the ING monitoring solution?
a. 1 year or less
b. 2-3 years
c. 4-5 years
d. 6 years or more

A.3 Part 2: Quantitative Questions

2.1 How many models are within the scope of your work?
a. 1 model
b. 2-3 models
c. 4-6 models
d. 7–9 models
e. 10 models or more

If the answer is selected ”10 models or more”, the following question is asked.
2.1.1 Can you please indicate the actual number of models?

2.2 How often do you use the ING monitoring solution in order to manage your team’s models?
a. Less than once in a week

b. Once in a week
c. 2-3 times in a week
d. 4-6 times in a week
e. 7 times in a week or more

If the answer is selected ”Less than once in a week” or ”7 times in a week or more”, the following question is asked.

2.2.1 Can you please indicate the actual number of models the ING monitoring solution is used on?

2.3 How long does it take to verify all your team’s models through the ING monitoring solution?

a. Approx. 1 minute
b. Approx. 5 minutes
c. Approx. 15 minutes
d. Approx. 30 minutes
e. Approx. 1 hour or more

If the answer is selected ”1 hour, or more”, the following question is asked.

2.3.1 Can you please elaborate on your choice?

A.4 Part 3: Qualitative Questions

3.1 Can you please mention three challenges of the ING monitoring solution?

3.1.1 Can you rank the following challenges in relation to the ING monitoring solution?

a. End-to-end pipeline support
b. Data Availability, Collection, Cleaning, and Management
c. Education and Training
d. Model Evolution, Evaluation, and Deployment
e. Collaboration and working culture
f. Specification
g. Scale

3.2 Can you please rank up to three features of the ING monitoring solution that helped effectively support the model management most?

a. Synchronizes the distribution of dataset, and compares the data distribution of predicted values and features
b. Search and print the log messages of the data science pipeline
c. Track the progress of the running data science pipeline

d. Track user-defined metrics
e. Overview of Multiple Models
f. Manage the Projects by Tags
g. Thresholds for metrics

3.3 The ING monitoring solution makes managing my team's models easy
a. Strongly Disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly Agree

If the answer is selected ”Agree” or ”Strongly agree”, the following question is asked.
3.3.1 Can you please mention three reasons why you agree with this statement?

If the answer is selected ”Disagree” or ”Strongly disagree”, the following question is asked.
3.3.2 Can you please mention three reasons why you disagree with this statement?

3.4 The ING monitoring solution makes managing my team's models efficient
a. Strongly Disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly Agree

If the answer is selected ”Agree” or ”Strongly agree”, the following question is asked.
3.4.1 Can you please mention three reasons why you agree with this statement?

If the answer is selected ”Disagree” or ”Strongly disagree”, the following question is asked.
3.4.2 Can you please mention three reasons why you disagree with this statement?

3.5 The ING monitoring solution increases the quality of my team's models
a. Strongly Disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly Agree

If the answer is selected ”Agree” or ”Strongly agree”, the following question is asked.
3.5.1 Can you please mention three reasons why you agree with this statement?

If the answer is selected ”Disagree” or ”Strongly disagree”, the following question is

asked.
3.5.2 Can you please mention three reasons why you disagree with this statement?

3.6 The ING monitoring solution supports trust in the automated model creation process
a. Strongly Disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly Agree

If the answer is selected ”Agree” or ”Strongly agree”, the following question is asked.
3.6.1 Can you please mention three reasons why you agree with this statement?

If the answer is selected ”Disagree” or ”Strongly disagree”, the following question is asked.
3.6.2 Can you please mention three reasons why you disagree with this statement?

3.7. The ING monitoring solution is an essential tool for developing and maintaining machine learning models
a. Strongly Disagree
b. Disagree
c. Neutral
d. Agree
e. Strongly Agree

If the answer is selected ”Agree” or ”Strongly agree”, the following question is asked.
3.7.1 Can you please mention three reasons why you agree with this statement?

If the answer is selected ”Disagree” or ”Strongly disagree”, the following question is asked.
3.7.2 Can you please mention three reasons why you disagree with this statement?