INF FACULTY OF COMPUTER SCIENCE

Master Thesis

Machine Learning approach for Enterprise Data with a focus on SAP Leonardo

Kavish Rastogi
Magdeburg, September 20, 2018

Supervisor: MSc. Gabriel Campero Durand
Professor: Prof. Dr. Alexander Zeier
Second Assessor: Prof. Dr. Klaus Turowski

Otto-von-Guericke-University Magdeburg
Magdeburg Research and Competence Cluster
Faculty of Computer Science
Department of Technical and Business Information Systems
Very Large Business Applications Lab

Abstract

In recent years, enterprises have experienced an expansion of their digital information in the form of structured and unstructured data. Certain adaptations are being done to accommodate this overflow of information in long established enterprise landscapes. Despite the availability of information, businesses still face challenges in acquiring knowledge from the data they posses. At the same time Machine Learning (ML) technology has quickly become a fast growing application area, since stakeholders can foresee the potential of these algorithms to gain competitive edge from their enterprise data. But integrating ML into enterprise information systems is non-trivial given the natural complexities in large enterprise systems and in ML technology.

In this study, I examine different approaches to face enterprise ML challenges, assessing the factors that might support users in their choice between approaches. The specific approaches that I evaluate include, deep learning solutions applied to enterprise data in a less integrated environment and adopting a more integrated approach for ML in enterprises, using components of the SAP Machine Learning.

In order to achieve this task, I perform a scoped literature review on enterprise ML and I create a prototypical Quality Management use case that involves image recognition. To guide my work I adopt the CRISP-DM Process Model, documenting all stages of the process. I offer two implementations of models for image classification, with the two approaches under study, evaluating the resulting performance of the models in the learning task. Based on the practical work of the implementation I am able to score the approaches with respect to relevant criteria inferred from my literature review. As an artifact from my work I am able to offer a principled trade off comparison between these approaches evaluated, summarized in the form of a decision matrix that can be immediately applied to help developers in deciding between approaches. Furthermore, my work shows a methodology that can be replicated in other studies for comparing integrated and less integrated solutions, and for closer evaluations of the criteria I have proposed.

Acknowledgments

By submitting this thesis, my long term association with Otto von Guericke University will come to an end.

First and foremost, I would like to express sincere gratitude to my professor, supervisor and co-founder of SAP HANA Prof. Dr. Alexander Zeier, for giving me the opportunity to write my Master's Thesis at his chair.

I start by thanking my university supervisor, Mr. Gabriel Campero, for being kind and cooperative in helping me nurturing the project's idea. He regularly assisted, corrected and improved my work. He guided me to eventually achieve my goals.

At Accenture GmbH, I would like to acknowledge the efforts of my supervisor in Kronberg, Mr. Heiko Steigerwald and my manager Mr. Marc Egdmann for accepting my ideas for this thesis and constantly supporting and guiding me. They also helped me get in touch with the experts on the relevant topics, that they met in their professional journey. Without their guided directions, this journey would have been much tougher for me. Furthermore, I would also like to thank Accenture's SAP HANA Innovation Center, Kronberg (AICS) team for providing infrastructure, expert advices, and moral support.

Finally, I express my profound gratitude to my parents and friends for their continuous support and encouragement. I am really grateful to all the people mentioned, and those who I might have forgotten but helped me nevertheless.

Contents

List of Figures

List of Tables

List of Abbreviations

AFL ..... Application Function Library

AIC ..... Akaike Information Criterion

AI ..... Artificial Intelligence

API ..... Application Programming Interface

APL ..... Automated Predictive Library

BI ..... Business Intelligence

BPMN ..... Back Propagation Neural Network

BYOM ..... Bring Your Own Model

CPU ..... Central Processing Unit

CRISP-DM ..... CRoss Industry Standard Process for Data Mining

DL ..... Deep Learning

EML ..... External Machine Learning Library

ERP ..... Enterprise Resource Planing

FCN ..... Fully Connected Network

GPU ..... Graphics Processing Unit

gRPC ..... Google's Remote Procedure Call

IoT ..... Internet of Things

JSON ..... JavaScript Object Notation

KDD ..... Knowledge Discovery in Databases

MCTS ..... Monte-Carlo Tree Search

ML ..... Machine Learning

PaaS ..... Platform as a Service

PAL ..... Predictive Analysis Library

PMML ..... Predictive Model Markup Language

QIP ..... Quality Inspection Process

QM ..... Quality Management

RL ..... Reinforcement Learning

RPC ..... Remote Procedure Call

S4/HANA ..... SAP Business Suite 4 SAP HANA

SaaS ..... Software as a Service

SCP ..... SAP Cloud Platform

SVM ..... Support Vector Machines

TBD ..... To Be Decided

UDA ..... User Defined Aggregates

UDF ..... User Defined Functions

1 Introduction

Data is an ever growing asset. The amount of digital information being produced has been growing exponentially for the past two decades [GR12]. Traditional data models have also been extended or upgraded to absorb novel data structure changes into information systems. Enterprises employ business process management software that allow organizations to use a system where their applications to manage business processes and back office functions are integrated [Swa00]. Such software tools are a combination of application modules and databases. These long established databases traditionally store data in relational models, as tables. In the past few years due to the expansion in the amount of data, enterprises have seen both structured as well as unstructured data as part of their information systems.

Machine Learning (ML) is a sub-field of Artificial Intelligence (AI) which adds the ability to automatically learn and improve from experience without setting up any rules or explicitly programming systems [M$^+$97]. Humans have sensed the possibility of machines being intelligent since the middle of the 20th century when Alan Turing studied the question *Can machines think?* in his work [Tur50]. Since then, the field has experienced its major evolution in the last 2 decades, partly thanks to the availability of more learning data, made possible by the digitization of the society. With all the advancements, the scientific community has recently reached a landmark in the form of Alpha Zero [SHS$^+$17], a machine learning framework mastering the games of Chess, Shogi (Japanese chess) and Go without any human data or guidance. One of the features that makes this such an accomplishment, is the fact that it is a general solution to more than one problem, and it has achieved super-human performance on all of them.

With the rising competition in business, Machine Learning is already a fast growing application area because stakeholders can see the potential of machine learning algorithms to gain competitive edge from their enterprise data. ML techniques have shown benefits for both marketers and consumers in all industrial domains. Small to large organizations are embracing deep learning, a subset of machine learning, to confront computationally challenging tasks varying from machine vision, to text analysis, genome analysis and many other complex tasks. Businesses seek to deduce insights and wisdom from their structured and unstructured enterprise data by leveraging AI techniques. Kashyap provides an overview of enterprise applications of ML spanning financial, manufacturing and healthcare domains, among others [Kas17].

For a majority of these applications either data has been brought to Machine Learning or vice versa, and applications are based on an implementation of an algorithm on use-case specific datasets. In the former case, the client of machine learning is commonly a data scientist or analyst who drives machine learning or uses it to mine knowledge, usually from a stand-alone copy of the data. But when it comes to large organizations they possess complex systems structures with varied and multiple data sources due to which integrating scalable and sustainable machine learning into enterprise landscape by either moving data to ML platforms, or bringing ML functionality to traditional data management tools is not a trivial task.

Relevant to this challenge, machine learning platforms have evolved to offer a set of features that enable them to sustainably support the application of ML at scale in large organizations, for increasing the business impact. Gartner defines machine learning platforms as "*A cohesive software application that offers a mixture of basic building blocks essential both for creating many kinds of data science solutions and incorporating such solutions into business processes, surrounding infrastructure and products*" [IKB+18]. There are numerous machine learning platforms available which organizations have either built based on their specific needs (e.g. TensorFlow, Apache SystemML, etc.) or they are buying them as a service from providers such as Google Cloud, Amazon AWS, Microsoft Azure, etc. Some of these machine learning platforms are part of Gartner's standard report of magic quadrant, identifying them as leaders in their fields [IKB+18].

This thesis revolves around SAP's Leonardo which may influence enterprise machine learning for more than 378000 small and medium sized enterprise using SAP products across the globe in the near future[1]. The focal point of this thesis is asking the alternative ways in which enterprises can leverage machine learning technologies for high end computational tasks on their structured and unstructured data, and mitigate the challenges involved in the process. I also aim to assess, practically the pros and cons of approaches. In order to do so, the following research questions are formed and answered throughout this document.

1.1 Research Aim

This work is built upon the knowledge that enterprises possess huge amounts of useful data both structured and unstructured, which can be used to gain competitive edge by using machine learning technologies; but integrating machine learning into enterprise information systems is nontrivial and might require data movements and the use of tools external to enterprise information systems. Hence, this thesis intends to find out possibilities for how these challenges or complexities can be circumvented. The following research questions are set as guidelines to fulfill this aim:

[1] As of April 2018, according to SAP's corporate fact sheet: `https://www.sap.com/corporate/en/documents/2017/04/4666ecdd-b67c-0010-82c7-eda71af511fa.html`

- When adopting less integrated ML solutions, including deep learning, within a system like the SAP Landscape, what can be the expected performance and benefits from the user perspective, from such approach?
- When adopting a more integrated approach for ML in enterprise data in a system like the SAP Landscape, what can be the expected performance and benefits from the user perspective? What are the trade-offs between integrated and less integrated approaches, such that scientists could make informed choices about what approach to adopt?

1.2 Research Methodology

In order to answer the defined research question, I comply to the Design Science Research Methodology and, more specifically, to the CRISP-DM model.

Peffers et al. describes methodology as a set of principles, practices, and processes that are applied to a specific branch of knowledge [PTRC07]. Rajasekar et al. have formulated a Research Methodology as *"the procedures by which researchers go about their work of describing, explaining and predicting phenomena"* [RPC06]. Design science research methodology executes a set of activities combining synthetic and analytical techniques to do research in information systems, solving problems at the intersection area of information technology and organizations. This methodology gathers new knowledge by going through the artifacts to understand and improve the behaviour of the characteristics of information systems [VK04]. For this methodology it is often required to create or evaluate successful artifacts which are intended to solve identified organizational problems. This approach includes:

- Problem identification and motivation
- Objectives definition for a solution
- Design and development
- Evaluation
- Demonstration
- Communication [PTRC07]

In information systems, a framework is required for design science research to identify and evaluate the results of experiments. Hence, a methodology in design science research includes three elements:

- Conceptual principles that define the meaning of design science research
- Practice rules and regulations

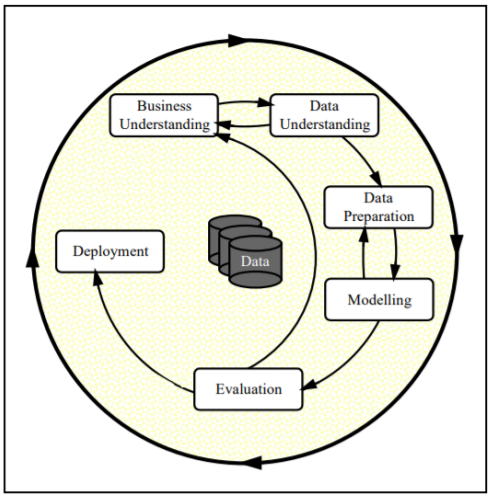

Figure 1.1: Process diagram to show the relationship between different steps of CRISP-DM from [WH00]

- Process performing and presenting the research activities

The conceptual principles include thorough processes to design artifacts that solve identified problems, makes research contributions, evaluates designs and communicates results to the appropriate recipients. The models, methods and instantiations can be included into the artifacts. The practice rules and regulations defines guidelines that describes the characteristics of research activities. It implies to create the artifacts to find the relevant solutions of the observed problems by evaluating utility, efficacy and quality [PTRC07].

Methodology development for design science research in information system can be done by introducing a design science process model. The process model provides a complete methodology along with prior research to design science research. The process design plans to meet three objectives:

- Provides a nominal process to conduct design science research.
- Build on prior literature about design science in information system and reference disciplines.
- Provides researchers with a template to present research results [PTRC07].

The CRoss Industry Standard Process for Data Mining (CRISP-DM) is one such process model which is being used by researchers in machine learning domain and widely accepted. The relevant literature considered for this research methodology are:

- Knowledge Discovery and Data Mining: Towards a Unifying Framework (Fayyad, 1996) [UG+96].
- CRISP-DM: Towards a standard process model for data mining (Wirth and Hipp, 2000) [WH00].

Fayyad et al. described Knowledge Discovery in Databases (KDD) as the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data in their work [UG+96]. Data mining and analysis in KDD has the goal of finding patterns in the data, through several diverse processes. The Cross Industry Standard Process for Data Mining model an overview of data mining project life cycles. The steps defined by CRISP-DM model are:

- Business understanding
- Data understanding
- Data preparation
- Modelling
- Evaluation
- Deployment [WH00]

From figure 1.1, the first step focuses on understanding the research or project objectives and requirements from a business perspective. Next, the information is converted into a data mining problem. Data collection activities are performed in the data understanding phase, along with getting insights into the data. Interesting subsets are discovered from hidden information to form hypothesis. In the data preparation phase, final data sets for modelling are constructed. This task includes data cleaning, data transformation and integration, attribute selection and the generation of new attributes. It can be performed multiple times to prepare final data sets. The modelling phase selects various modelling techniques to design and build the modelling framework. It is critical, from a data analysis perspective, to evaluate the model before deployment. One or more models are evaluated during the evaluation phase. Eventually, the final model is deployed during the deployment phase ensuring that the knowledge gained from the research work is organized and presented for further use [WH00].

In my research I adhere to the CRISP-DM process model stages while developing a representative use case from data quality management, following the steps of understanding, modelling and evaluation.

1.3 Structure of Thesis

This thesis document provides an overview of major literature contributions relevant to the topic of machine learning in an enterprise context, including introductory concepts on machine learning, deep learning, and the challenges of adding machine learning to enterprise data. Furthermore, I describe a prototypical implementation carried out to compare quantitatively and qualitatively two approaches. To this end I present a use case from quality management. Following the CRISP-DM Process Model I indentify the business

use case and describe the steps of data understanding and preparation. An experiment has been conducted, adhering to the design science research methodology, to answer the specified research questions. Additionally, I carry out a discussion to form comparison of adopted solutions, both the integrated and the less integrated approach. At the end I present a decision matrix as an artifact that encapsulates my evaluation, with criteria determined in this study. In the conclusion, I present a summary of my work along with my findings, some open points from experiments, and concluding remarks to end.

2 Literature Overview and Fundamentals

This chapter introduces the basic background necessary to understand this research. I structure this chapter as follows:

- I begin by providing an overview of literature studied to understand the concepts of ML, DL, Challenges of ML in Enterprises and SAP Leonardo - Machine Learning Portfolio (Sec. 2.1).

- I explain what is machine learning, its significance for enterprises, the challenges it faces and SAP Leonardo' elements (Sec. 2.2). From these contents the coverage on machine learning and deep learning is kept to the necessary in order to provide motivation and essential concepts. The coverage on the challenges of machine learning for an enterprise context is more comprehensive and carried out in a systematic, reproducible way. At the end of this section I provide an introduction to the framework used to apply deep learning with systems like SAP (Sec.2.2.5.3).

2.1 Literature Overview

Enterprise machine learning is a topic of interest since several years now, for both enterprises and machine learning researchers. There are certain challenges which this deed faces in terms of scalablity, integration, performance, governance, etc.
In this chapter I present fundamental concepts necessary to understand the topic. I start by discussing preliminary research works related to Machine Leaning concepts, then I talk about deep learning, its industrial use cases and enterprise-level implementations. Moreover, I put considerate focus on the expansion of enterprise big data and challenges in Enterprise ML. I conclude with an overview of SAP Leonardo's scientific and industrial white papers.
In this section I provide an overview of the literature used, in the next section (Sec. 2.2) I discuss the fundamental concepts themselves.

2.1.1 Machine Learning

The modern era of technology, gives us diverse use cases that depict the significance of machine learning algorithms in both our daily lives, as well as in the enterprise world. By virtue, the rapid development of this domain in the scientific research communities have contributed abundant literature works. This thesis report incorporates some basic notions

of machine learning algorithms from a set of research papers and textbooks. I utilize the following sources:

- Machine Learning (Mitchell, 1997) [M+97], a reference textbook which introduces basic concepts related to machine learning, information theory, statistics and artificial intelligence.
- Introduction to Machine Learning (Baştanlar and Özuysal, 2014) [BÖ14], which gives a good overview of fundamental concepts of machine learning.
- Alpha ZERO (Silver et al., 2017) [SHS+17], a paper exemplifying a general algorithm to play the game of Chess, Shogi and GO using Reinforcement Learning.
- Why Machine Learning and why now? (Weller et al., 2017) [WWDK17], a white paper from the Global Lead of SAP Digital Future, defining the relevance of ML to enterprises.
- Machine Learning for Decision Makers (Kashyap, 2017) [Kas17], a book that outlines industrial applications of machine learning.

2.1.2 Deep Learning

Deep learning is a sub field of machine learning based on the principle of neural networks. It has been discovered that traditional machine learning are bounded in their capacity to process natural data in their crude form, since these approaches usually expect some predefined features.

Deep learning overcomes this limitation by enabling learning processes capable of automatically detecting features from data (i.e., learning representations). Hence, these techniques have proven to be successful for domains where data presents some underlying hard-to-define structure, such as image recognition [KSH12].

To include industrial deep learning in this thesis report I use a number of research papers, web sources and industrial white papers, they are as follows:

- Deep Learning (LeCun et al., 2015) [LBH15], a comprehensive textbook that concepts regarding complex deep networks, their learning methods and discusses popular domains of application.
- What is Deep Learning on Machine Learning Mastery (Brownlee, 2016) [Bro16], inline with Deep Learning (LeCun et al., 2015) [LBH15].
- Deep Learning: Methods and Applications (Deng et al., 2014) [DY+14], which demonstrates an overview of methodology and application areas for deep learning.
- Deep Learning: A Guide for Enterprise Architectures (Dinsmore, 2017) [Din17], a white paper connecting deep learning methodologies to enterprise complexities.

- An Introduction to Deep Learning by SAP (Wu and Razavi, 2015) [WR15] a technical paper by SAP providing a view on the field.
- TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems (Abadi et al., 2016) [AAB+16], for large scale deep learning in enterprises. Since Google's open source software TensorFlow is opted because of its integration with SAP HANA. This paper introduces the framework and presents the structure for processing information with TensorFlow.
- TensorFlow: learning functions at scale (Abadi et al., 2016) [ABC+16], complements the previously mentioned work from the same team, with a good insight about the library focusing on the programming abstractions.
- Google's and SAP's web documents.

2.1.3 Expansion of Enterprise Data

The expansion of data catalyzes challenges and opportunities. Challenges concerning managing, storing, monitoring the data, and opportunity in terms of extraction of knowledge from big data. There are plentiful studies conducted in the past which rightly anticipated the inflation in the amount of data in enterprises, an aspect which will only grow in future. I utilize a set of white papers and textbooks to incorporate this topic in the report, as follows:

- The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East (Gantz and Reinsel, 2012) [GR12], a report on a study about the expansion of the digital universe, including estimations on the growth of data.
- Big Data : What It Is and Why You Should Care (Villars, 2011) [VOE11], a white paper connecting the dots between the amount of data creation and the technological structural changes big organizations must care about to gain value from the their data.
- Machine Learning for Decision Makers (Kashyap, 2017) [Kas17], a book that, as discussed above, highlights the projected amount of data in the world created by enterprises of all kinds, including some critical sectors like finance, banking, and insurance.
- High-Performance Storage Systems (Lefelar, 2017) [Lef17], a white paper that reiterates the same fact in a context that involves product/services endorsement from Jeskell Systems.

2.1.4 Challenges in Enterprise Machine Learning

While citing about machine learning and the its potential to extract knowledge from enterprise data for business advantages, one needs to contemplate how it can be integrated

into the existing enterprise landscape. There are few works available which explicitly focus on this objective. In order to provide a systematic, reproducible study, I carried out the following keyword searches in Google Scholar: "Data Management in Machine Learning", "Enterprise Machine Learning" and "Production Machine Learning". The searches produced 1, 33 and 118 results respectively[1]. Based on the results I was able to identify the following sources as my primary studies[2], for which I studied the papers that such studies cite and, in turn, the papers that cite such studies:

- Data Management in Machine Learning: Challenges, Techniques, and Systems (Kumar et al., 2017 [KBY17]) a tutorial given at a data management conference (SIGMOD), reviewing systems and techniques tackling data management challenges for machine learning workloads. This tutorial covers machine learning in data systems, database-inspired techniques in machine learning platforms, and systems to manage the machine learning lifecycle. In my presentation on the field I focus on the first section of the tutorial.

- Machine Learning : The High-Interest Credit Card of Technical Debt (Sculley et al., 2014) [SPE+14], a paper proposing a list of aspects which, if left unattended, can introduce complexity into production-level ML.

- Data Management Challenges in Production Machine Learning (Polyzotis et al., 2017) [PRWZ17] a paper which discusses key challenges in governing data for enterprise machine learning pipelines.

2.1.5 SAP Leonardo Machine Learning portfolio

SAP has an impactful customer base of 378000 small and medium sized enterprise across the globe[3]. To study their machine learning offerings several research work was consulted. The majority of technical information was retrieved from the corporate web library: *help.sap.com*. Other literature consulted can be listed as follows:

- SAP Leonardo Machine Learning Foundation Demystified (Dadouche, 2018) [Dad18], an article that gives an overview of the portfolio.

- Predictive Analytics Reimagined for the Digital Enterprise (SAP, 2017) [SAP17g].

Thus I have outlined the literature sources that I adopted for my study. I would like to note the use of a small amount of industrial white papers was considered appropriate for capturing the enterprise perspective. In the next section I discuss the study itself.

[1] As of the 24th of June, 2018

[2] After setting aside work that was about proposing new systems (e.g. [ABC+16]), blogposts (e.g. [Zin17]), bachelor theses or degree projects, work that was not publicly available, and less related work to my core topic of data management for enterprise machine learning (e.g., a survey on applications of machine learning in the statistical domain only [CP15]

[3] As of April 2018, according to SAP's corporate fact sheet: `https://www.sap.com/corporate/en/documents/2017/04/4666ecdd-b67c-0010-82c7-eda71af511fa.html`

2.2 Fundamentals

In this section of the chapter some fundamental topics are explained inline with the assessment of their relevance, provided by researchers and market stakeholders in the field. Each section focuses on a subject area which holds importance to answer the specified research questions. The first section concentrates on what is machine learning (Sec. 2.2.1) and how enterprises have the opportunity to overcome its implicit challenges (Sec. 2.2.2). Architectural challenges and complexities of enterprises with ML in Data Systems are highlighted in the subsequent section (Sec. 2.2.3). An overview of SAP Leonardo Machine Learning offerings is stitched based on reviewed literature on the topic (Sec. 2.2.4) and then the last section elaborates on deep learning integration within SAP systems explaining the basic technology requirements (Sec. 2.2.5) before proceeding with the implementation approaches I explored with implementation, in the subsequent chapter.

2.2.1 What is Machine Learning

Machine learning is an application of artificial intelligence that enables systems to learn and get better with experience without being explicitly programmed for it. ML focuses on the development of computer programs that can access data and use it to learn for themselves. For better understanding how Machine Learning works there is a need to understand how to form a learning problem.

I choose below definition from Tom M. Mitchell's book called Machine Learning [M$^+$97] because of its correctness and number of citations to his book [4]. He state *learning problem* as "*A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E*". For example, a computer program that learns to play checkers might improve its performance as measured by its ability to win at the class of tasks involving playing checkers games, through experience obtained by playing games against itself. Essentially, to have a well-defined learning problem, there is a precondition to identity these three features: the class of tasks, the measure of performance to be improved, and the source of experience.

2.2.1.1 What are Different Types of Machine Learning

As specified by Mitchell in [M$^+$97] ML comprises of several types of learning, based on the problem structure. The most frequently used in prevailing applications areas are Supervised and Unsupervised learning along with Semi- Supervised and Reinforcement Learning.

Başţanlar et al. in their work [BÖ14] divided ML techniques into two main categories depending on whether the output values are required to be present in the training data or not.

[4] scholar.google.de

Unsupervised learning techniques require only the input feature values in the training data and the learning algorithm discovers hidden structure in the training data based on them implicitly. Clustering techniques that try to partition the data into coherent groups fall into this category. In general, market segment analysis, grouping people according to their social behaviours, and categorization of articles according to their topics are popular tasks involving clustering and unsupervised learning. Frequent pattern mining is also a form of unsupervised learning [BÖ14, p. 107].

Supervised learning methods require the value of the output variable for each training sample to be known. As a result, each training sample is represented in the form of a pair of input and output values. The algorithm then trains a model that predicts the value of the output variables from the input variables using the defined features in the process. If the output variable is continuous valued then the predictive model is called a *regression function*. For instance, predicting the air temperature at a certain time of the year is a regression problem. If the output variable is a discrete set of values then the predictive model is called a *classifier*. A typical classification problem is automated medical diagnosis for which a patient's data need to be classified as having a certain disease or not [BÖ14, p. 109].

Semi-supervised learning lies in the middle of both above specified methods and can be more advantageous, since unlabelled data is more accessible than high-quality labelled data. This family of learning methods works with a small labelled training dataset (supervised) and a larger unlabelled dataset(unsupervised). While training a predictive model, these algorithms can exploit both the supervised output values and the data distribution in the unlabelled data. However, these algorithms make additional assumptions to take advantage of the unlabelled data, which may or may not be suitable for the problem at hand [BÖ14, p. 110].

Reinforcement Learning is a slightly different approach because the above methods learn functions, logical theories, and probability models from examples. Reinforcement learning is about how agents can learn what to do in the absence of labelled examples. It is a learning method that interacts with its environment by producing actions and discovering from the environment errors or rewards. Trial and error search, while dealing with the exploitation vs. exploration dilemma, and the challenge of modeling delayed rewards are some of the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behaviour within a specific context in order to maximize agent's performance. Simple reward feedback is required for the agent to learn which action is best. The fact that it doesn't require labelled examples makes it different and valuable [M$^+$97, p. 367].

2.2.1.2 Alpha ZERO

Alpha Zero [SHS+17] which is a specific implementation of a Reinforcement Learning approach (i.e., Deep Q-Learning), is considered as a modern day accomplishment for the research community in the field of AI. The Alpha Zero algorithm is based on the principle of combining a neural network and reinforcement learning, trained entirely through self play after being given the rules of the game. The difference between Alpha Go Zero, an earlier version of the algorithm, and Alpha Zero is that the former was designed specifically to play the game of Go. Alpha Zero is a more general algorithm than Alpha Go Zero to solve more than one task. According to Silver et al. [SHS+17] Alpha Go replaces the handcrafted knowledge and domain specific augmentations used in traditional game-playing programs with deep neural networks and a tabula rasa reinforcement learning algorithm. Instead of having a handcrafted evaluation function and move ranking heuristics, Alpha Zero utilizes a deep neural network $(p, v) = f_\theta(s)$ where θ represents parameters, s represents board positions and p is a vector of move probabilities. The neural network takes the board positions s as an input (i.e., as an observed state) and gives a vector of move probabilities p as output with components $p_a = Pr(a|s)$ for each action a, and a scalar value v estimating the expected outcome z from position $\mathtt{s,v} \approx \mathbb{E}[z|s]$. The outcome corresponds to the long-term reward expected from each move. Silver et al. mentions that Alpha Zero learns these move probabilities and value estimates entirely from self play; these are then used to guide its search both during training, and during playing.

Alpha Zero uses a general purpose Monte-Carlo tree search (MCTS) algorithm with each search consists of a series of simulated self-play games that traverse a tree from the root *sRoot* to leaves. Each simulation proceeds by selecting in each state s a move a with low visit count, high move probability and high value (averaged over the leaf states of simulations that selected a from s) according to the current neural network f_θ. The search returns a vector π representing a probability distribution over moves, either proportionally or greedily with respect to the visit counts at the root state.

The parameters θ of the deep neural network in Alpha Zero are trained by self-play reinforcement learning, starting from randomly initialized parameters θ. Games are played by selecting moves for both players by MCTS, $a_t \sim \pi_t$. At the end of the game, the terminal position s_T is scored according to the rules of the game to compute the game outcome *z:* -1 for a loss, 0 for a draw, and +1 for a win. The neural network parameters θ are updated so as to minimize the error between the predicted outcome v_t and the game outcome z, and to maximize the similarity of the policy vector p_t to the search probability π_t. Specifically, the parameters θ are adjusted by gradient descent on a loss function l that sums over mean-squared error and cross-entropy losses respectively,

$$(p, v) = f_\theta(s), l = (z - v)^2 - \pi^T \log(p) + c\|\theta\|^2 \tag{2.1}$$

where c is a parameter controlling the level of L_2 weight regularization. The updated parameters are used in subsequent games of self-play.

Alpha Zero challenged one of the best computer chess engines called *StockFish* whose performance is already higher than the best human chess player. In the games Alpha Zero vs StockFish they were both given 60 seconds of thinking time per move. Alpha Zero was able to outperform StockFish in about 4 hours of learning from scratch. From 100 games Alpha Zero won 28 times, drew 72 times and never lost to StockFish. StockFish is already powerful compared to even best human prodigies and Alpha Zero outperformed it with just 4 hours of self play and it was run with the one machine and 4 TPU's, a specialized hardware for tensor processing, which can help the training of deep neural networks. Another point to note was that StockFish doesn't use machine learning and is a handcrafted algorithm which is not learning anything; on the other hand Alpha Zero is a more general algorithm which can also play Shogi i.e. Japanese chess and Go at an extremely high level. Alpha Zero would be highly useful even if it were slightly weaker than StockFish because it is built on more general learning algorithm that can be reused for other tasks without investing significant human effort; in spite of the generality (or perhaps because of it), it is able to outperform StockFish. With every successive work from the authors the algorithm is becoming better and more general. The Alpha Zero algorithm was also applied to Shogi and Go. Unless otherwise specified, the same algorithm settings, network architecture, and hyper-parameters were used for all three games. A separate instance of Alpha Zero was trained for each game. And the results in these were surprising as well. As it out performed Elmo, the strongest Shogi program and a previous version of Alpha GO Zero convincingly.

The key points of this invention are as follows:

- It is using almost negligible computation power as compared to its previous versions.
- It is not using human knowledge or any data, as it is working solely on reinforcement learning.
- It is a general solution which can be used for more than one task with minimal human effort.
- Its performance is outstanding against the best in the tasks that it was trained on.
- It is able to use experience from one task to become better for another task.

Alpha Zero can be considered to be a landmark in AI research. In spite of the accomplishment, enterprises are yet to identify in which form can these promising technologies be integrated into everyday business. Some applications of RL exist in robotics, industrial automation, health, medicine, etc.; but more applications, leveraging advanced techniques for deep RL such as those used in Alpha Zero, could be considered, and they might hold a valuable potential.

2.2.2 An argument for the potentially privileged position of enterprises to leverage machine learning

Except for reinforcement learning, the other three machine learning approaches, supervised, unsupervised and semi- supervised, rely upon data. More data catalyzes the performance of these approaches.

With the digitization of our society and advancements in technologies, ML is able to find its utilities across landscapes and a large suite of applications. An enormous amount of data is generated on daily basis with high availability and accessibility. To process this huge amount of data, there have also been developments for large-scale processing tools, and of cloud-services offering cheap computation power. Both supervised and semi- supervised learning techniques should be able to benefit immediately from these trends, but these techniques need at least some relatively large amount of labelled data. In fact labels are usually not explicitly available. The annotation process (i.e., to create labels) can be time consuming, repetitive, expensive and very often it requires of human experts. There are few techniques developed as a workaround such as semi- supervised learning, which can work with little labelled data and make use of unlabelled data, active learning [Set10], among others. In spite of this the requirements for labelled data and the high cost for it remain central challenges for supervised and semi-supervised techniques.

In dealing with this challenge enterprises have an implicit advantage, since labelled data can be often found within their systems, since they are either handling their business processes through rule-based systems, through certain kind of automation tools or manually, through human intervention. By virtue, such methods have unexpectedly generated good amounts of data which can be leveraged to use machine learning based systems for upgrading and understanding better existing business processes. In order to optimize existing processes the stakeholders need to think about the whole value chain end to end. From designing products to having raw materials, inbound logistics to manufacturing, outbound logistics i.e. selling, marketing products and other operations. Stakeholders need to analyze the entire back office across finance, procurement, human resources, real estate, asset management. All these business functions together build business processes, all the tasks which are transactional and high volume in nature and act on digital information in business processes is potentially within reach of a significant augmentation of productivity enhancement with machine learning today.

Global Leads of SAP Digital Future have proposed such argument for the privileged position of enterprises for using machine learning, in their whitepaper: Can machine learning transform core elements of business ecosystem [WWDK17]. Authors claim that, a large number of business processes are administered by unbending, software-based rules. This approach is restricted in its capacity to handle complex processes. Besides, these processes regularly expect employees to invest energy in exhausting, exceedingly tedious work. If

enterprises change the rules and let self-learning algorithms handle their repetitive tasks, machine learning could uncover valuable new patterns and solutions that possibly they never knew existed. In the interim, employees could be reassigned to all the more captivating and vital tasks. Authors further added that, economy relies upon infrastructure, including energy, logistics, and IT, and additionally on administrations that help society, for example, education and health care. Enterprises appear to have reached an efficiency plateau in these territories. Machine learning can possibly find new signals in the data that could allow improvement of convoluted and quick evolving systems. This gives humans more opportunity to apply their innovativeness to new disclosures and advancement. The group also expressed thoughts on the advancements in machine learning technologies as where a future can be imagined in which robots, machines, and gadgets running on self-learning algorithms will work considerably more autonomously than they do now. They may arrive at their own particular decisions inside specific parameters, adjust their behaviour to various circumstances, and interact with people significantly more closely. Gadgets effectively ready to respond to voices which will become more intelligent, consistently learning assistants to assist with day to day business schedules, for example, scheduling meetings, translating documents, or analyzing text and data [WWDK17].

Machine Learning is undoubtedly a technology to look for in the near future for all small and large enterprises, but inculcating new setups into the old architectures is not a trivial task. In the next subsection of this chapter few challenges for the adoption of ML in enterprises are discussed. I consider both challenges on the technical aspects of data management, and on more organizational aspects.

2.2.3 Challenges in Enterprise Machine Learning

Machine Learning is a powerful tool to build complex data-driven systems expeditiously; but building such systems requires costs such as acquiring new hardware, building data pipelines, data management, complexities at system-level, etc. Making the enterprise intelligent is going to be an amalgamation of machine learning technologies and old traditional methodologies. In this sub section of the chapter I highlight some prominent issues which enterprise machine learning may face. Researchers have proposed two categories for presenting these challenges [PRWZ17]. The categories are :

- Challenges in Production Machine Learning, pertaining to ML activities.
- Challenges in Enterprise Machine Learning, pertaining to Enterprises.

But before presenting the results in these challenges I will discuss the work of Kumar et al. [KBY17], reviewing technological developments for Machine Learning to interact with Data Systems (i.e., mostly databases, being relevant for an enterprise context), instead of Machine Learning that is unaware of data systems and runs on batch loads. After presenting work on this field, I will introduce work covering the challenges mentioned, and

to conclude the section I will provide a brief summary of the discussed work, leading to a selection of criteria that could be relevant for consideration in the enterprise ML domain.

2.2.3.1 Machine Learning in Data Systems

The key goal for supporting ML in data systems is about moving the analysis to the database and not the data to the analysis platform [KBY17]. Authors state that such approach can not only simplify development, and reduce data movements, but can also contribute to optimizations of the process, provided that the ML system is able to optimize its data access, in a similar way that traditional database research has done. The approach of supporting in-database ML could be important for real-world uses, and hence the requirements for data movement could be considered one criteria to comparatively evaluate ML platforms. Similarly, since this line of work focuses on improving the execution speed and the integration with existing systems, I find that these two aspects are also criteria for evaluating ML platforms.

MAD was a system proposed by Cohen et al. [CDD+09], to realize machine learning algorithms within databases, by focusing on SQL-based matrix operations using sparse representations and UDFs. Similar work like RiotDB (Zhang et al., 2009) [ZHY09], Glade (Rusu et al., 2012) [RD12], Bismarck (Feng et al., 2012) [FKRR12] (focusing on SGD), SimSQL (Luo et al., 2018) [LGG+18] and MADLib (Hellerstein et al., 2012) [HRS+12] (using a Python UDF) have studied the adoption of user-defined aggregates (UDAs) to support the matrix operations that underly several ML algorithms such as K-means and gradient descent. Closely related work is embodied in SAP HANA SLACID (Kernert et al., 2014) [KKL14], a library for linear algebra kernels within a database.

In terms of language interface to the data scientists, most systems propose the support of ML through ML-oriented languages on top of SQL, whereas systems like MAD, SimSQL considered that ML could be best supported through extensions to SQL. The latter is an approach similar to SAP HANA PAL.

There is another area of work in supporting ML within a database, which is called *factorized processing*. This area of research considers the essential difference between ML and normalized data storage. Namely, that ML algorithms usually expect to have all features of an entity available for use in the learning process, whereas storage usually divides an entity into multiple tables. Hence, for ML algorithms to run it might be necessary to perform a large amount of join operations to reconstruct the entities. The core idea of factorized processing is to improve this procedure by pushing down the ML algorithms steps to the individual tables, without the explicit requirement to join all the data distributed across tables for each entity. Systems like Orion (Kumar et al., 2015) [KNP15], Morpheus (Chen et al., 2017) [CKNP17] and Santoku (Kumar et al., 2015) [KJY+15] study this approach. Recently the work of Abo et al. also consider this [AKNN+18].

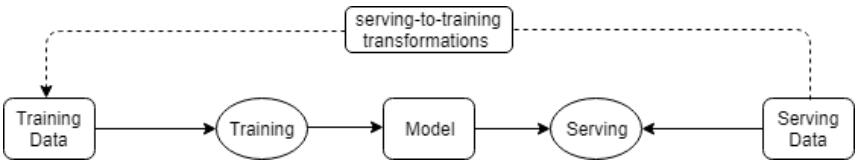

Figure 2.1: High-level diagrammatic representation of a Machine Learning Pipeline in a Production Environment from [PRWZ17]

Another area of research called statistical relational learning, considers ML models that are specially tuned to relational data models, where relational data serves as the ground truth for statistical inference using probabilistic models [KBY17].

2.2.3.2 Challenges for Machine Learning in a Production Environment

According to Polyzotis et al. [PRWZ17] the fundamental challenges to which machine learning systems are vulnerable while deployed in a production environment pertain to data management issues that arise in machine learning pipelines.

Figure 2.1 shows a high level machine learning pipeline where the input to the system is the training dataset which is then fed into an algorithm to train a machine learning model. The trained model, in turn, is then picked up by a serving infrastructure and used in combination with serving data to generate predictions over newly arrived data, also called inferences. In most cases a subset of serving data along with predictions are fed back in to the system as additional training data. For a simplified description, the diagram omits several crucial tasks, such as testing of generated models against arbitrary hold out data, replacing already served models or doing dark model launches, etc. Polyzotis et al. [PRWZ17] state that major challenges in managing data for production machine learning pipelines are *Raw-data Understanding*, *Data Validation*, *Data Cleaning*, *Data Enrichment*.
These four challenges pertain to the domain of *model governance*[5].

Raw-data Understanding
Data Engineers or Data Scientists, who set up the pipelines for machine learning place a significant amount of time and effort in analyzing their raw data. Some of such tasks includes generation and visualization of salient features about the data, identification of outliers present in the data, encoding the data into features appropriate to the trainer (e.g. one hot encoding, the choices in this step can have a great effect on the goodness of the training), and understanding data context (which can play a crucial role in the process to identify explicit and implicit data dependencies, eventually could affect the model, data

[5]Makhtar et al. describes the process of creating predictive models [MNR10], which includes steps of Data Preparation, Data Reduction, Data Modelling and Prediction, Evaluating and Validating of the model and Implementing and Maintaining the model; the effective management of all the steps in this process is collectively called Model Governance.

provenance management techniques are suited to keep track of such dependencies).

Apart from this, machine learning brings oddity to the problem, such as the requirements of identifying short term or long term impacts of removing a feature from the pipeline, or keeping track of provenance when data is coming and processed through highly heterogeneous sources.

To scale these tasks to a huge amount of training data is not trivial. Furthermore, to generalize this tasks across datasets and domains, with partial automation of them, is not trivial either [PRWZ17].

Data Validation

Data validity significantly affects the quality of the generated model. Data validity comprises of several aspects, namely: a) making sure that expected characteristics are present in training data (e.g., features and their correlations); and b) that serving data is also aligned with training data.

Some of the challenges in the first task can be addressed through traditional mechanisms from database systems. For instance, something analogous to schema can make sure the presence of expected features and characteristics of their values. Correlations among features are related to functional dependencies in relational databases, in spite of the fact that some issues are machine learning specific. For example, ML introduces a unique type of constraints such as bounds on the drift in the statistical distribution of feature values, moreover any schema over training data needs to be flexible to allow changes in the characteristics of the features as they reflect real world events.

Training-Serving skew, aberration between training and serving data is a dominant source of problem in production machine learning pipelines. The core issue is the data used to train the model is different than the data used to serve the model which results in incorrect predictions [PRWZ17].

Data Cleaning

After data validation and detection of validation error next step is to clean the data. This task can be disintegrated into three sub tasks: comprehending where the error occurred; comprehending the impact of the error; and, fixing the error. In order to explain better Polyzotis et al. have illustrated examples aligned to these subtasks. For first subtask, assume that the data became invalid because value distribution of a feature is changed significantly over time. An investigation of "feature X has different distribution" may not be helpful rather than a thorough investigation like "feature X has different distribution in training examples where Y takes values in [20, 40)". This confined point may help engineer or data scientist to understand whether this is an actual error or a natural evolution of data.

Forming this localization is similar to understanding which regions of the data are interesting or relevant for the detected anomaly, and so it may be possible to leverage works related to guided OLAP exploration. The second subtask pertaining to comprehending the impact of error on model quality. Due to several reasons, the team may be willing to continue with the pipeline with invalid data if the model quality is not deteriorating much. Although, calibrating the impact of the error is non trivial, considering that ML algorithm is in principle a black box. In this situation, some characteristics of the function for specific classes of machine learning algorithms can be leveraged in order to derive the impact analytically, or run a number of experiments to quantify the impact empirically which is tedious. The third task, cleaning the data is to fix the error which can be done by addressing the root cause, for example, fixing a bug in the script that generates the data. A work around to this is to patch the the data inside machine learning pipeline as a temporary fix [PRWZ17].

Data Enrichment

Data Enrichment refers to the expansion of training and serving data with new features in order to enhance the quality of the generated model. A common mode of it is to merge a new data source with new signals or using the same signals with different transformations. The core problem here is identifying which additional signals or transformations can fulfill the purpose in a meaningful way. Another evenly crucial problem is helping the team understand the boost in model quality by enriching the data with a certain set of features. This information will help the team decide whether to invest resources in implementing the enrichment in production [PRWZ17].

2.2.3.3 Challenges for Enterprise to absorb Machine Learning in their landscape

Not only machine learning systems but also the old traditional systems to which ML systems is integrating or communication needs to address few situational challenges as elaborated by Sculley et al. in [SPE+14]. Sculley et al. followed the framework of technical debt, introduced by Ward Cunningham in 1992 as a way to help quantify the cost of such decisions. His work aims at complexities in system- level interaction between machine learning code and larger systems. These issues can be segregated under four headings, they are *Complex Models Erode Boundaries*, *Data Dependencies Cost More than Code Dependencies*, *Configuration Complexities*, *Changes in External World* [SPE+14].

Complex Models Erode Boundaries

By virtue of modular design and concepts like encapsulation, long established software engineering practices have shown strong level of abstractions, which help to maintain code. In case of machine learning systems, it is observed that enforcing strict abstraction boundaries are difficult to enforce, because of the high dependency on external data. A possible instance of model eroding boundaries can be where a input distribution of value of a feature is changed, which may change the importance, weights for remaining features in a model. Adding a new feature can cause similar changes as removing one. The effect of

this change may cause change in the prediction pattern on various distribution of slices. One possible strategy to mitigate its effect is isolation of models and serve ensembles but this approach is useful in scenarios where cost of maintaining separate models is overshadowed by advantages of enforced modularity. However, this strategy may prove to be unscalable in many large-scale settings. And within a given model, the issues of innate entanglement may still be present. A similar situation lies in case of hidden feedback tools. Systems learning from world's behaviour are bound to be part of a feedback loop [SPE+14].

Data Dependencies Cost More than Code Dependencies

Analogous to code dependencies in traditional software settings, data dependencies carries same complexity in machine learning systems. Moreover, Sculley et al. argues that code dependencies can be relatively easy to recognize with the help of static analysis, linkage graphs, etc but existence of such analysis tools is less common in case of data dependencies. Most probable for creating data dependencies are unstable data signals. Unstable data signals are the signals which are generated by other systems and used as input features in some other system, which is a common convenient process, however, these signals change behaviour over time. This can happen implicitly, if the signal comes from another machine leaning model that updates over time, or explicitly, if the engineering ownership of input signal is separate from that of the model consuming it. In these cases, changes to the input signal may be routinely unrolled without considering how it may affect the ML system. Underutilized signals may also create unnecessary dependencies between ML models and enterprise data sources. For code, underutilized dependencies are packages or functions that are mostly nonessential . Likewise, underutilized data dependencies include input features or signals that contributes little towards accuracy. These dependencies are costly, since they make the system unnecessarily vulnerable to changes. Another dependencies arises in case of cascading models. It happens oftentimes that model l exist for problem L, and a model for marginally different problem L' is required. In this scenario, it can be enticing to learn a model $l'(l)$ which is derived from model l with a small correction. However, model l' has created a system level dependency on model l which will make it reasonably more expensive to detect improvements in that model in future. An even more complicated scenario may arise if correction models are cascaded in chains [SPE+14].

Configuration Complexities

Another potential area to look, for is configuration of machine learning systems. Any complex large system has a variable range of configurable options, which includes features selection, how data is selected, algorithm specific learning settings, required pre- or post-processing, validation and evaluation methods, etc.

Sculley et al. in his work [SPE+14] mentions that many engineers do a commendable job of thinking hard about abstractions and unit tests in production code, but may treat configuration (and extension of configuration) as an afterthought. Indeed, verification or

testing of configurations may not even be seen as important. Configuration by its very nature tends to be the place where real-world messiness intrudes on beautiful algorithms. They support their argument by the following example, "*Feature A was incorrectly logged from 9/14 to 9/17. Feature B is not available on data before 10/7. The code used to compute feature C has to change for data before and after 11/1 because of changes to the logging format. Feature D is not available in production, so a substitute features D' and D" must be used when querying the model in a live setting. If feature Z is used, then jobs for training must be given extra memory due to look-up tables or they will train inefficiently. Feature Q precludes the use of feature R because of latency constraints*". All this complexness makes configuration difficult to modify correctly, and hard to reason about. Though, mistakes in configuration can be costly, can lead to loss of time, waste of computing resources, or production issues. In an evolved system, which is being developed actively, the number of lines used for configuration can far exceed the number of lines of code that actually does machine learning. Each of those lines are prone to errors, and configurations are by their nature, less well tested [SPE+14].

Changes in External World

One of the most interesting thing about machine learning systems is that they often interact with external world, and experience has shown external world is rarely stable. The instability of external world can cause problems for dynamic systems with fixed threshold. It is a common practice to pick a decision threshold for a given model to perform some action for instance, to mark an email spam or nor spam. One typical approach is to choose a threshold from a set of threshold values in order to get good trade-offs on certain metrics. But often such thresholds are manually set. Consequently if a model gets update on new data, the old set threshold may become invalid. Manually updating many thresholds across many models is time-consuming and brittle. Another open point is monitoring and unit testing of such systems. Unit testing of components and end to end test of running systems are essential before the systems are deployed. But with changing world, such tests are not enough to prove that the system is working as intended. Real time live monitoring of system behaviour is critical [SPE+14].

I highlight few production machine learning learning challenges out of many, based on experience gathered by the stakeholders. Moreover, industries have realized about the effect of the dent these situational challenges can cause to the performance of their machine learning systems and consequently they are opting for machine learning platforms specific to their needs which can help them to automate machine learning innate processes, make their models scalable and can perform ML in a more integrated or close environment. In this I focus on SAP Leonardo which is an offering from SAP because of their high customer base among small and medium sized enterprises.

The four challenges mentioned here, from the work of Sculley et al. [SPE+14], are or-

ganizational challenges that affect data scientists and the team working with the model. Such challenges can be considered to fall in the domain of model governance, and they require effective tooling support from ML platforms, to assist teams in facing them. In addition, underlying all concerns are the aspects of the performance of the resulting model, and the ease in building a solution. Hence I conclude that the performance that can be achieved by ML models, and the ease in building a solution are also a criteria that might be relevant for comparing ML platforms.

In my review of work in the field of ML for enterprise data I covered 3 essential aspects. First, I looked into work on applying ML inside databases. In this area I identified three main directions, namely, matrix opertions support through UDFs, factorized processing and statistical relational learning. Secondly I considered challenges a list of challenges for production machine learning, and Third, challenges for machine learning in an enterprise context. Based on this literature review I was able to highlight the following criteria as relevant for comparing ML approaches in an enterprise context:

- Execution speed
- Performance of the resulting models
- Data movement (i.e., how efficiently can the system support users in reducing the overheads stemming from data movement).
- User support for model governance (i.e., for the tasks of data cleaning, preparation, etc.)
- Integration with existing data systems
- Ease in building a solution
- Support for state-of-the-art methods[6]

After listing this criteria, I will present necessary background on the Machine Learning offerings of SAP, as it pertains to the selection of tools and the domain of research in my work.

2.2.4 Machine Learning platform for Enterprises by SAP – SAP Leonardo

SAP is one of the providers of enterprise application software. They claim to have a customer base of 378,000 in more than 180 countries among small and medium sized enterprises. It has been estimated that 76% of the world's transaction revenue touches SAP systems [SAP17b].

SAP S4/HANA is SAP's new business suite. It has been architected for the in-memory SAP

[6] I introduce this criteria, as it holds relevance for the performance obtainable in a task.

HANA database. SAP S4/HANA is offered with on-cloud, on-premise and hybrid deployment options by the vendor. On the basis of underlying technology, S4/HANA facilitates the development of new business models. Due to changed business models, stakeholders ponders, it may change IT landscape for processes simplicity and efficiency. Simplification of business processes through integration of several components can be augmented if Artificial Intelligence technologies can be leveraged by using SAP Leonardo.

2.2.4.1 What is SAP Leonardo?

Patanjali Kashyap addressed SAP Leonardo in his work [Kas17] as a platform of different technologies from SAP and termed as *digital innovation system*. It includes technologies such as machine learning, blockchain, data intelligence, big data, IoT, and analytics to enable business competitiveness. Author state that, SAP Leonardo is developed to innovate from new business models and processes that S4/HANA demonstrates and change the way people work today. It connect things to people and process to data. It offers an exhaustive IoT and machine learning based solution ecosystem that encompasses the digital essential with adaptive applications, products, and services, including Big Data management and connectivity, to enable the following [Kas17]:

- New business processes (for example, industry 4.0).
- New business models (for example, cloud computing).
- New work environments.

SAP Leonardo is an umbrella term which includes elements that helps business to build applications from things/objects/devices to outcomes. For example, in the manufacturing industry, it could provide the following benefits [Kas17]:

- Live insights: Provides information, such as, condition monitoring, location tracking, environment, usage, and consumption patterns
- Predictive analytics: In real time, it analyses equipment health, remaining lifetime, demand and supply forecast, and time of arrival
- Optimize processes: Lower maintenance cost, higher efficiency, reduced waste, fewer claims, faster response, and shorter cycles
- New business models: Usage-based pricing, product-as-a-service

Some high level benefits which SAP Leonardo provides to business mentioned by Kashyap [Kas17] are:

- Increases revenue
- Re-imagines processes
- Increases quality time at work

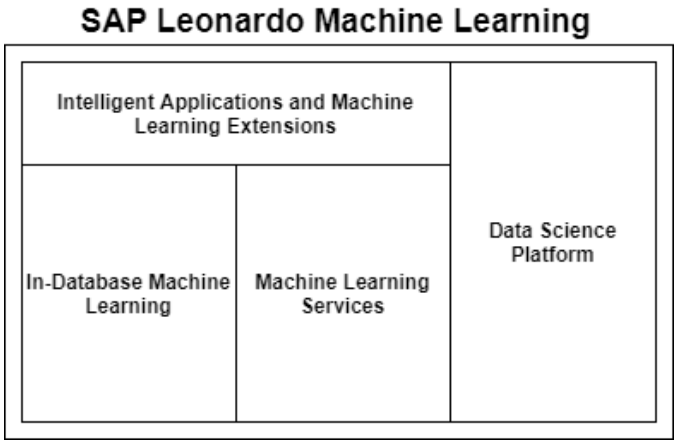

Figure 2.2: SAP Leonardo Machine Learning elements as shown in [Dad18]

- Increases customer satisfaction
- Enables innovations

The further subsections of this chapter will be specifically focusing on the SAP Leonardo Machine Learning technologies. I discuss about different offerings and their goal to ease down and automate enterprise machine learning with respect to SAP customers.

2.2.4.2 SAP Leonardo Machine Learning Portfolio

SAP Leonardo machine learning majorly comprises of four elements, shown in figure 2.2, discussed by Dadouche A. in his recent work [Dad18]. Elements are *In-Database Machine Learning*, *Data Science Platform*, *Machine Learning Services* and *Intelligent Applications and Machine Learning Extensions*.

In-Database Machine Learning
One of the core concepts of SAP Machine Learning is to move algorithms into the database, and to do so it makes use of application functions. Application functions are like database procedures written in C++ which are called from outside to perform data-intensive and complex operations. An Application Function Library contains these components in SAP HANA.

SAP HANA Predictive Analysis Library (PAL) and Automated Predictive Library (APL) are built-in C++ libraries to perform analytics. They defines functions that can be called from within SQL Script procedures and run algorithms on data stored in SAP HANA without requiring an expensive and time consuming data extraction process. The functional difference between two libraries is, *PAL* is designed for data scientists who have a data mining background and *APL* is an automated approach. *APL* only takes the type of data mining function needs to be applied to the data, then it composes its own models given the data. *PAL* is a customized approach and includes algorithms in ten

data-mining categories: Clustering, Classification, Regression, Association, Time Series, Pre-processing, Statistics, Social Network Analysis, Recommender System and Miscellaneous [SAP17d] [SAP18c] [Dad18].

In the work of Bittmann et al. PAL is used for frequent pattern mining [BNS+18] evaluating the role of optimizations to the algorithms.

SAP HANA External Machine Learning Library is another component of Application Function Library, introduced since SAP HANA Platform 2.0 SPS 02 [SAP17c]. It is different than *PAL* and *APL*, in terms of implementation and execution methodologies. The idea behind this library is to support the integration of SAP HANA with external machine learning frameworks, for instance, Google TensorFlow, and apply the state of the art solutions to problems, to which automated approaches of *PAL* and *APL* may not be suitable. The integration of TensorFlow with SAP HANA is based on the *EML* and Google's gRPC remote procedure call package. It also involves a separate server that hosts the actual machine learning functionality [SAP17c]. I discuss this library in details in section 2.2.5.3.

SAP HANA also provides integration with R[7]. The idea behind this integration is to enable the embedding of R code in the SAP HANA database context. That is, the SAP HANA database allows R code to be processed in-line as part of the overall query execution plan. The SAP HANA database uses the external R environment to execute this R code, similar to its database operations like joins or aggregations. This scenario is useful when a SAP HANA-based modelling and consumption application wants to use the R environment for specific statistical functions. In the integration, data movement part is analogous to SAP HANA TensorFlow approach (less integrated). The movement of intermediate database table to vector-oriented data structures of R is supported by data exchange mechanism. Data is duplicated as an additional data copy on the R side [SAP16].

Data Science Platform

Data Science Platform is a standalone or integrated software tool around which all data science tasks can take place. These data science tasks include preparing and exploring data from different sources, training an appropriate machine learning model on the data, apply trained model, capture the feedback. *SAP Predictive Analytics* and *SAP Analytics Cloud* are two versions of data science platforms from SAP Leonardo ML portfolio [Dad18].

SAP Predictive Analytics is a statistical analysis and data mining solution that enables building predictive models on the data, from which predictions can be made. It comprises of elements like *Data Manager* for data preparation, *Automated Modeler*, an automated approach for business analysts carrying out ML processes in an intuitive way, *Expert Analytics*, for more customized and problem specific models designed for data scientists and

[7] As of June 2018, R is an open source programming language and software environment for statistical computing, contains a rich collection of packages and functions. `https://www.r-project.org/`

Predictive Factory, as a component for automation of management of predictive model. The predictive factory approach enables governance, management, retraining, and scoring the probability for multiple models in a browser-based user interface. It allows to perform in-memory scoring with the SAP HANA platform and in-database scoring with third-party relational database management systems and data sources [SAP17g].

SAP Analytics Cloud, a cloud based Software as a Service (SaaS) built on SAP Cloud Platform (PaaS) for analytics solutions. It provides analytics capabilities including business intelligence (BI), planning, predictive analytics, and digital boardroom tools – in a single cloud-based solution [SAP17f].

Machine Learning Services

Another element of SAP Leonardo ML portfolio is its Machine Learning Services. These services are meant to provide extensibility, scalablity to the applications which consumes them to provide ready to consume machine learning functionality with minimal efforts and data risk by bringing analytics to the data. SAP Leonardo in this direction are equipped with several services which can serve as building blocks in process chains or applications [Dad18].

SAP Predictive Service is available on the SAP Cloud Platform (SCP) which enables cloud based applications with predictive functionality. With this service, an application can analyze the data which is stored in an SAP HANA instance on SAP Cloud Platform. It offers sets of RESTful web services that can be deployed on the platform as one application [Dad18].

Business Services allow to get insights from data, for business analysts. Each service is specific to a business question and returns a specific type of insight for example clustering, forecast, recommendation, key influencers, outliers, scoring equation and what-if [Dad18].

Predictive Analytics Integrator Services allow non-predictive cloud applications to easily integrate and consume predictive models. These services enable the use of predictive models within the context of real-life business processes. It enables users to configure, customize and manage the life-cycle of the model using SAP Predictive Analytics without development cycles. This application can be deployed on the instance of SAP Cloud Platform of the company or customer before using it [SAP18e].

SAP Leonardo Machine Learning Foundation [SAP17e] [Dad18] are another set of services which are available on SCP. These services provide offer customization while maintaining the focus on automation. The services are designed to tap the information hidden in unstructured or semi-structured enterprise data such as images, texts, videos etc. and making them useful for enterprise. By using these services ML technology can be incorpo-

rated into existing SAP or non-SAP solutions. The large amount of enterprise data within SAP systems can be used as training data for these services, in order to enable intelligence in business applications. They can learn from historical data of manual tasks in order to automate them. SAP Leonardo Machine Learning Foundation brings a predefined set of re-trainable machine learning models. It also provides an infrastructure for machine learning applications and services, for instance, possibility of bring and deploy your own model [SAP17e]. The models or services included in this collection are developed with the idea of avoiding customers efforts on complex tasks including both unstructured and structured data and providing accurate results. These services are majorly based on deep learning in order to understand semantic behind unstructured enterprise data. They are segregated into two sections: *Business Services* and *Functional Services*. *Business Services* includes *SAP Intelligent Financing* which analyses historical activities of users of a business network (suppliers and buyers) to calculate an index, called SAP Finance Health Score, which represents the sustainability of the business entity. One of benefits of SAP Finance Health Score is a supplier with a high score may receive a lower rate when financing with banks. And another one is SAP Service Ticket Intelligence as a service.

Functional Services are readily consumable pre-trained models that can be used as a web service by calling simple REST APIs. These services ranges from image and video processing, natural language processing, tabular and time series processing. The pre- trained models are generic in nature and may not fit customer specific needs. Re-Training for pre-trained functional services enable the customers to use state of art models in their business applications trained on their business context without any major efforts. By using *Customize Model API*, these services can be retrained and fine-tuned [Dad18].

Bring Your Own Model(BYOM) approach allows the customers to bring their own model, developed by their engineers specific to their use cases on the data residing on cloud HANA instance. By using this API, a TensorFlow model can be deployed to the Machine Learning Foundation and inference can run against it in SAP Cloud Platform by creating a Python application [Dad18].

SAP Leonardo Conversational AI Foundation is another offering as a part of SAP Leonardo Machine Learning. It allows development of conversational AI applications. With the help of Natural Language Processing, Conversational AI it can be used to simplify interactions in customer service, human resource, commerce, finance, employee self service etc. It is based on deep learning models for Natural Language Processing and integrates the latest developments in Machine Learning, Big Data, and Computing Power [SAP18a] [Dad18].

Intelligent Applications and Machine Learning Extensions

Dadouche in [Dad18] state that, SAP Leonardo Machine Learning provides fully build ready to use applications from different business lines in a simplified integrated manner,

they are as follows:

SAP Cash Application is an application from Finance domain, which deals with the analysis of utilization and acquisition of funds for enterprises. When customers pay invoices, account receivables accountants match payments with open invoices. There are a lot of challenges, if account receivables are not able to match payments with invoices. SAP Cash Application uses Machine Learning to automate this tedious task. It learns from accountants, automate repetitive task, and constantly adapts to changes. After bank statements are imported, ERP calls the cloud service for Cash application which then returns clearing proposals. This application is machine learning based cloud solution. The historical clearing data of all account receivables is captured and used for clearing new payments. With one time set up matching payments with invoices are automatically cleared and for payments that are not cleared automatically best fitting invoices are proposed [SAP17a] [Dad18].

SAP Customer Retention is an application from Sales domain, through which customer behaviour can be anticipated towards products by looking at the transactional data such as product cancellations or renewals. It automates data mining mining, predictive analysis, and capture leading churn indicators [SAP18a] [Dad18].

SAP Resume Matching is an application from Human Resource domain, which matches resumes to hiring needs with minimal manual effort. Recruiters spend most of their time in reviewing resumes this application can assist them to spend much more time in shortlisting candidates to automatically identified talents to open positions. It extracts the relevant information from candidate's resumes and matches them with job requirements and rank the candidates in order of relevance [Dad18].

SAP Service Ticketing is an application from Services domain, which helps to build a self driven customer service. Incoming customer tickets are automatically classified into their categories and directed to right agent. The agent is then provide with recommended solutions to improve operational efficiency. It uses deep learning networks trained on large amount of historical data. The model understands the semantics of unstructured ticket messages, classifies the ticket into their most likely categories and recommends solutions or knowledge base articles from similar previously answered tickets for the agent. With more processed service tickets and user's feedback, the model improves over time [SAP18b] [Dad18].

SAP CoPilot is a conversational AI based application. It runs on SCP and its key capabilities includes : Users can create notes while working with applications, SAP CoPilot links those notes to the application, user was working on, helping him find them when he return to the application. SAP CoPilot can take intelligent screen shots from SAP Fiori

applications and recognizes business objects within the current application context and allows user to add them to his collection of notes and screen shots. Chat with other users from your business application context without entering a collaboration room, and share notes, screen-shots and business objects. Creation of business objects from within SAP CoPilot, such as, a Contact Person. This application is considered as a digital assistant for the enterprises [Dad18].

SAP Brand Impact is an application from Marketing domain. Organizations make significant investment in sponsorship and in advertising their brand logo. It is a difficult task for them to understand what is their Return On Investment on this and how much their brand is being exposed in the media. SAP Brand Impact can detect how many times the logo of an organization is being shown in the media, for instance, in a football match. It provides accurate analytics about exposure time. Based on these analytics the advertiser or marketer can understand their brand value and brand exposure and calculate the Return On Investment of their money [Dad18].

SAP Leonardo Machine Learning portfolio provides options for their consumers, to automate business processes, around which scalable, automated and integrated machine learning solutions can be constructed.

2.2.4.3 Adding additional business value by unstructured and semi structured data

Gantz et al. expressed the expansion of Digital Universe in his work [GR12]. He defined elements of Digital Universe connected to daily life as images and videos on mobile phones uploaded to social networks, banking data swiped in an ATM, security footage at airports and major events such as the Olympic Games, subatomic collisions recorded by the Large Hadron Collider at CERN, transponders recording highway tolls, voice calls zipping through digital phone lines, and texting as a widespread means of communications [GR12]. Gartner research predicts a nearly 800 percent growth in the amount of enterprise data over the next 5 years and the majority of the data is expected to be unstructured. Unstructured data as part of enterprises is defined by Paul et al. in [Wil12] as any form of data that does not easily fit into a relational model or a set of database tables. It exists in variety of formats: books, audios, videos, or a collection of documents. Some of this data may very well contain a measure of structure, such as chapters within a novel or markup on a HTML web page, but not a full data model of relational database.

Although mining unstructured data is expensive due to several reasons, but enterprises which can better able to leverage meaningful business intelligence from their data gain a competitive edge over those who cannot. Major challenge in exploring unstructured data is *Requirements are not well defined*. It solely depends on business stakeholders and engineers extract as much benefits from the data. Machine Learning offers a diverse set of

applications which are working in enterprise context and helping them in simplifying their processes, for instance, Image recognition and classification API is being used for building *Intelligent Part Requisition Systems* from procurement area, where the user of the system can take an image of a part and information comes up from the catalogue proposing which part it is. The regular approach was, the user has to go the SAP screen, type in the part number and to know the part number the user has to go to multiple screens or to internet catalogue of the vendor, so it was a complex process and vulnerable to human errors. By using ML as an alternative option makes process simple, if the user knows the part number then he can type in directly or can take a picture of the part, the information comes up from the catalogue and he can request for the product.

In this thesis, I focus on a similar functionality in enterprise context from quality management domain, which is an integral part for most business lines. While focusing on quality management functionality I answer specified research questions.

2.2.5 Introduction of Deep Learning in Enterprise World

The focal point of this sub section is the introduction to the use case specific technologies that are implemented in later part of thesis. These technologies include deep learning, TensorFlow library which is a stable framework to design large scale deep models in enterprise context, the integration of SAP HANA database with deep model including real time scoring and other options of how deep models can be used with SAP Cloud Platform.

2.2.5.1 What is Deep Learning, Why Deep Learning and When to use Deep Learning

With the advent of GPUs and cheap computation power, the challenge of handling high end computation tasks is mitigated. Availability of tremendous amount of data have supported the expansion of this research area. Deep Learning is a sub field of Machine Learning works on the principle of Artificial Neural Networks. Schmidhuber described neural networks as it consists of many simple connected processing units, neurons, each producing the sequence of real valued activations. Input neuron layer gets activated through the environmental inputs, other intermediate neurons layers get activated through weighted connections from previous layer activated neurons, in his work [Sch15]. For a specific deep learning definition, Deng et al. [DY+14] summarized it as a class of machine learning techniques that exploit many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification [DY+14].

Deep Learning has outperformed all significant traditional machine learning approaches in diverse application domain, such as, image recognition, speech recognition, predicting activities of drug molecules, various tasks in natural language processing and several others. The reason of its success is, its ability to process natural data in its raw form and efficient handling of huge amount of data in its model structures. A white paper from

Cloudera [Din17] highlights key strengths of deep learning that distinguishes it from other machine learning techniques. The first among them is feature learning. With other traditional techniques, data scientists needs to typically transform features to get best results out of a particular algorithm which is a time consuming process and includes a lot of guesswork as well. On the other hand, deep learning learns high-level abstractions from input data at many levels and the data scientist does not guess how to combine, recode or summaries the inputs. Deep Learning is also capable of detecting interactions among variables which may be invisible on the surface. It can detect nonlinear interactions and approximate any arbitrary function automatically. While with other simpler ML techniques, it is possible to fit interaction effects but, those methods require manual specification and more guesswork from the data scientist [Din17].

Although, after extreme success and several benefits of deep learning, it is not feasible to use it everywhere in enterprise context because of its computational intensiveness, technical opaqueness, etc. In some situations classic machine learning techniques or rule based approaches might be preferable. Deep Learning could be a promising approach when:

- Large amount of training data is available.
- Learning problem involves unstructured form of data and the model need to learn meaningful representations, e.g. images, audios, text.
- Input is high dimensional discrete or real-value.
- Long training times are acceptable.

2.2.5.2 What is TensorFlow and TensorFlow Serving

In this section, I introduce TensorFlow, a functional, open source machine learning library for numerical computations and understand key concepts such as tensors, computation graphs and sessions.

TensorFlow Elements

TensorFlow was released by Google in 2015 [Zac16]. It is one of the most accepted and widely used library to develop machine learning frameworks, because of its high flexibility, end-to-end solution for development and deployment. The models developed in TensorFlow can easily be moved from prototyping to production. Figure 2.3 represents high level architecture of TensorFlow and how it goes from development to deployment without much re-engineering. The researchers at Google built TensorFlow library to scale, it is made to run on multiple CPUs or GPUs or mobile operating systems and has several wrappers. TensorFlow provides a device agnostic execution framework, that is, executed by the TensorFlow distributed execution engine and the end user deals with various language front-ends that TensorFlow supports. Python is the most stable and easy to use API but it supports various other front-ends, such as, C++, Java, etc. TensorFlow layers

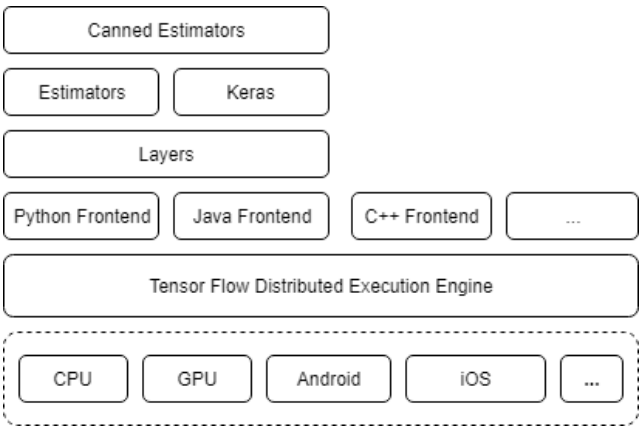

Figure 2.3: A high level representation of TensorFlow Architecture

provide utilities to build models while estimators allows experimentation with different architectures. TensorFlow estimators provides standard interface to perform task that need to do with a model. TensorFlow Keras is an API specification built on core TensorFlow features allowing easy prototyping of machine learning framework. On top of the architecture lies canned estimators, which supports efficient implementations of standard models [AAB+16] [ABC+16] [Zac16].

Programmatically, TensorFlow works around tensors, graphs and sessions. Tensors is a generalization of vectors and matrices to higher dimensions. Internally, they are represented as n-dimensional arrays of base data-types, hence a $n \times n$ tensor is an array of numbers in the shape of a square. Each element in a tensor has a constant and known data type. The main objective of a TensorFlow program is to manipulate and pass around *tf.tensor*, a tensor object represents a partially defined computation that will eventually produce a value. TensorFlow uses data-flow graph to represent computation in terms of dependencies between individual operations, which leads to a low level programming paradigm where a data flow graph is established and then session is created to run parts of the graph across a set of local and remote devices [AAB+16] [ABC+16] [Zac16].

Data-flow graph has several advantages which TensorFlow leverages during execution, such as, parallelism, distributed execution, compilation and portability. A *tf.graph* object contains two relevant information in it: Graph Structure and Graph Collections. Graph Structure, nodes and edges of the graph, representing how individual operations are composed together. Graph Collections, provides a mechanism for storing collections of metadata. This function enables an association of list of objects with a key which are used while serialization or optimization of the model variables [AAB+16] [ABC+16] [Zac16].

TensorFlow sessions are responsible for graph execution. The library uses *tf.session* class to do this, which represents a connection between the client program, typically a python program. The *tf.session* object provides access to devices in the local machine, and re-

mote devices using the distributed TensorFlow run-time. It also caches information about *tf.graph* so that same computation can run efficiently multiple times. It is typically used as a context manager that naturally closes the session while exiting the block since it owns physical resources (such as GPUs and network connections) [AAB+16] [ABC+16] [Zac16].

TensorFlow Serving

Realizing the benefits of sophisticated machine learning models is only possible when they are effectively served. Serving systems for production environments needs to have high efficiency, low latency, reliability, horizontal scalability, and robustness. TensorFlow Serving framework, is designed to be flexible, offers a complete serving solutions for machine learning models to be deployed in production and optimized for TensorFlow [Fie16] [BBC+17].

It can be used to serve multiple models, at large scale, that change over time based on real-world data, enabling: model life-cycle management, experiments with multiple algorithms, and efficient use of GPU resources. In a simplified training pipeline, training data is fed to the learner, which outputs a model. Once a new model version becomes available, upon validation, it is ready to be deployed to the serving system. TensorFlow Serving uses the (previously trained) model to perform inference - predictions based on new data presented by its clients. Since clients typically communicate with the serving system using a remote procedure call (RPC) interface, TensorFlow Serving comes with a reference front-end implementation based on gRPC, open source RPC framework from Google. TensorFlow Serving is written in C++ and it supports Linux. TensorFlow Serving introduces minimal overhead [Fie16].

2.2.5.3 SAP HANA integration with TensorFlow serving

In this subsection, I discuss Google TensorFlow integration with SAP HANA from External Machine Learning library of SAP HANA. The integration of these two softwares is based on the SAP HANA Application Function Library (AFL) and Google's gRPC remote procedure call package. It involves a separate server process that hosts the actual machine learning functionality. The integration allows the application developer to embed TensorFlow functions definitions and calls within SQL script and submit the entire code as part of a query to the database. The creation or training of the model takes place at the TensorFlow machine and the ready to use model is served by using TensorFlow serving. SAP HANA is a thin client which make calls to TensorFlow serving via HTTP access. The communication happens in the form of tables, the information passed to or get back into the system are transformed into HANA tables [SAP17c].

Figure 2.4 depicts a high level architecture of how SAP HANA communicates with Google TensorFlow. On the SAP HANA side, as shown, the communication is driven by Application Function Library, which provides machine learning functionality to SAP HANA with PAL and APL libraries. Although, to access other open source machine learning

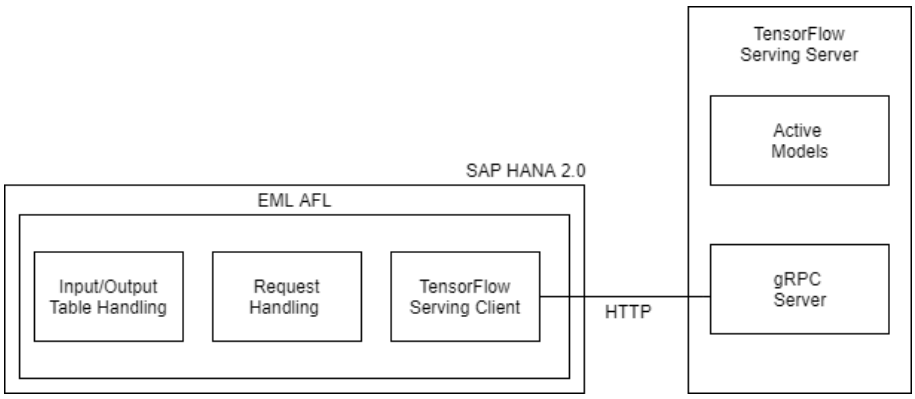

Figure 2.4: SAP HANA integration with Google TensorFlow from [SAP17c]

frameworks, it provides External Machine Learning Library through which SAP HANA communicates with TensorFlow. Since, AFL EML is providing communication only with TensorFlow till the current version release [SAP17c], it can be considered as a TensorFlow serving client implementation for SAP HANA. On the TensorFlow side, serving server makes TensorFlow exported models accessible for execution through gRPC remote procedure calls. The exported models are persisted in a specific format configured in the given serving model server. There are active models and non-active models, active models are the ones currently served and therefore available for the execution. The gRPC server interface facilitates communication with the TensorFlow Serving Model Server client [SAP17c].

With help of this integration enablement, enterprises can use deep learning methods on their enterprise data silos residing in SAP or Non-SAP systems and make these methods part of their business process chains. For instance, in a fraud detection use case where transactional data is analyzed to detect transactions which are potentially fraud, and take further actions (approve or reject). Moreover, transactional data can be huge and complex in terms of features set where conventional machine learning techniques may not be able to learn from feature sets or take advantage of huge amount of data and detect accurate decision boundaries, flexible deep learning techniques can be useful here. For use cases involving unstructured data, such as, images, videos, raw texts and audios, deep leaning is already established as state of the art.

2.3 Summary

In this chapter I presented essential background to understand my research topic. I started with literature overview pertaining to ML, DL and SAP Leonardo. I provided a literature review for challenges in enterprise machine learning and machine learning in data systems through which I eventually inferred my evaluation criteria used in Chapter 4 for integrated and less integrated approaches. In the second part of this chapter, I introduced fundamental concepts from overviewed literature related to my research topic.

In the next chapter I formulate the evaluation questions that stem from my research questions, and I present my results for the first stages of the CRISP-DM Process Model.

3 Prototypical Implementation

In this chapter, I present a prototypical implementation for showing how the previously elaborated research questions can be practically realized. Included in this discussion is a segregation of the lengthy research questions proposed into sub questions in order to answer them effectively. SAP Quality management is explained functionally, along with an introduction to the experimental setup for modelling implementations.

3.1 Evaluation Questions

To address my research questions I gave an overview, in the previous chapter, of challenges in enterprise machine learning and SAP Leonardo offerings. My research questions, as presented in Chapter 1, are focusing on the effect of circumventing movement of data silos for applying machine learning on them in enterprise context.

I will answer the following questions in subsequent chapters, providing more fine-grained considerations to the questions that I have presented thus far:

1. How can deep learning solutions be applied to enterprise data within a system like the SAP Landscape?

2. When adopting less integrated solutions, what is the impact of data movement?

3. What can be expected performance gains from such approach?

4. How can we adopt an integrated approach for ML in enterprise data, using SAP Leonardo?

5. What are the benefits achieved by reducing data movements?

6. What are the trade-offs between integrated and less integrated approaches, such that scientists could make informed choices about what approach to adopt?

For the evaluation of research questions I focus on the External Machine Learning Library and the Predictive Analysis Library from SAP Machine Learning. Furthermore I select a use case from quality management automation. In the next section I introduce this use case.

3.2 Use Case - Quality Management Automation

Quality Management (QM) is an essential piece of the logistics work inside an SAP framework[1]. It is important for warehouses, to review incoming material as it arrives at the base office; it is also important for manufacturing tasks, where the quality of in-process items are inspected during the manufacturing process; ready to go goods need also to be assessed before they reach the warehouse. The QM components of a framework like SAP cover three separate areas of *planning*, *notifications* and *inspections*[2]. The quality planning function allows the quality department to plan inspections for goods receipts from vendors and production, work in process and stock transfers. A quality notification can be used to request actions to be taken by the quality department. This may be to review an internal problem, an issue with items from a vendor or a customer complaint. The quality inspection is the physical inspection using specifications defined in quality planning, perhaps a quality technician physically identifies the defect during a routine inspection.

In this subsection, I formulate the possible automation for a task in the inspection process of quality management, with the help of Computer Vision and ML technologies. These technologies can be used to design an automated way of detecting defects in a product in an assembly line with the help of an image classification system, processing product images for checking physical defects in them. I suggest a prototype solution for this task which can be extended to several other capabilities, that is, not just detecting shape anomalies but other anomalies as well, based on available data.

In order to employ this automation task to test the research questions, I use SAP Leonardo Machine Learning components. In the next subsection I discuss in depth about how quality inspection works and how my selected use case contributes to the process. This section corresponds to the stages of Business Understanding from the CRISP-DM process model.

3.2.1 Quality Management - Quality Inspection Process and Automation

A QM module from the complete enterprise information system can deal with almost all conventional functions pertinent to quality in an organization[3]. The unification of QM application components in the SAP systems allow for quality management tasks to be combined with other process applications (for instance, material management, sales and distribution, cost accounting and production). The Quality Management application component assists tasks related to quality planning, quality inspection and quality control. In addition, it deals with the creation of certificates and manages problems with the help of notifications pertaining to quality.[4]

[1] Source: `https://help.sap.com/erp2005_ehp_05/helpdata/en/a6/df293581dc1f79e10000009b38f889/frameset.htm`, retrieved on 25.06.18
[2] Ibid.
[3] Ibid.
[4] Ibid.

The components of Quality Management in an enterprise tool are:

- Basic Data: For instance, material master, catalogues, inspection methods, inspection characteristics and sampling procedures.
- Quality Information System: Dashboard-like interfaces for interacting with the complete QM data.
- Quality Certificates: Core data from the system, ISO, ISO 14001, and other kinds of certificates are organized in this module.
- Task modules: These are modules that support specific tasks of the overall QM process:
  - Inspection Planning: Inspection planning, reference operations sets, material specifications.
  - Inspection Lot Processing: While doing sampling this process takes place. It includes lot creation and lot inspection.
  - Result Recording: Results from quality application are recorded here.
  - Defects Recording: In case defects are found in the manufacturing process, then those are recorded here.
  - Dynamic Modification of the Inspection Scope: Optional task, in case modifications are required in a traditional inspection process.
  - Quality Notification: While following a quality management, notifications can be raised if a process error comes or a defect is encountered.
  - Test equipment management: In the production process equipment need to be tested in order to avoid any error.
- Modules for specific domains:
  - QM in Procurement: While procuring goods from outside vendors, what are the quality parameters that should be taken into consideration?
  - QM in Sales and Distribution: Material is supplied to customers while supplying the material there are some parameters of quality management.

Figure 3.1 describes the Quality Inspection Process (QIP):

1. It starts with an inspection plan and routing; when the material is delivered by the vendor, it needs to be inspected before it goes to production line.
2. If the material is in huge in quantity, a sample procedure takes place. This means that, if in a production order there are 100 batches of a particular raw material and there is no sufficient time to inspect each and every batch, a sample of batches is chosen from these batches and only this sample gets inspected.

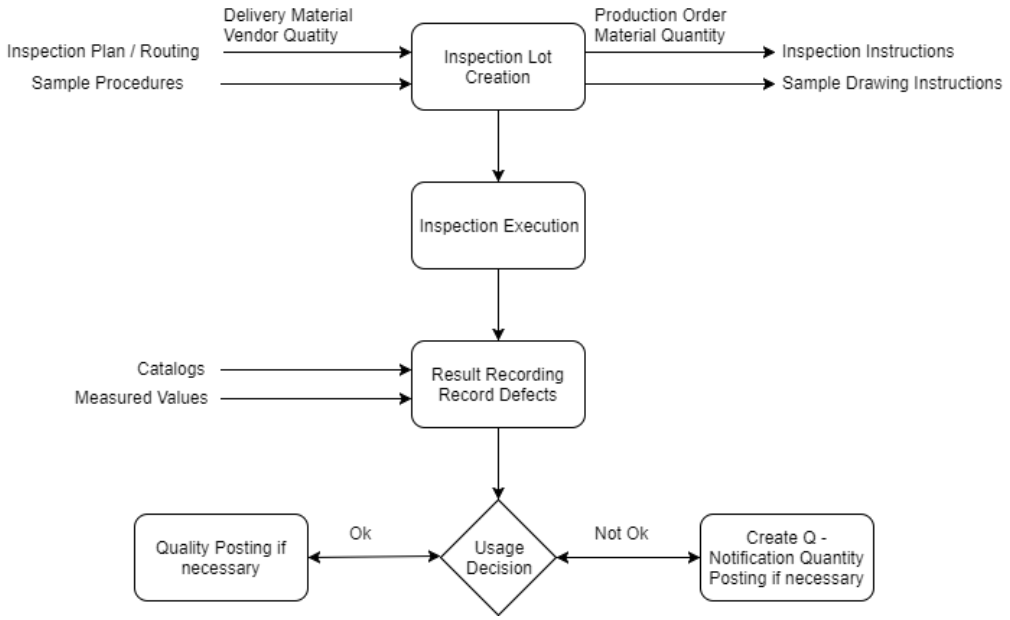

Figure 3.1: Quality Inspection Process Flow Chart from [Gre17]

3. After deciding on sample procedures, an inspection lot using SAP tools, by the help of inspection instructions and sample drawing instructions [5].

4. In the next stage of QIP, inspection execution takes place, here the actual inspection occurs, possibly in a manual way. The results and defects that are found have to be recorded in catalogues, also recording the measured values.

5. Once the results and defects are recorded, the process goes to a usage decision, determining whether this batch is Ok for production or is not; if the decision is Ok, a quality posting is created if necessary. If the batch is rejected then a quality notification is created and quality posting if necessary, so that the engineers may come to know the quality inspection of a particular material.

There is a prevailing assumption, that in case of huge quantities of materials, if a sample of material is fine than the whole material lot is good. Naturally, this depends on the specific department and use case.

If the inspection execution stage of this whole process flow is manual, then it could be prone to human errors. This can be highly costly, since errors at this stage may eventually affect the base line production. If the department or user were consider automating the specific inspection execution stage, then the whole process could become more accurate and repeatable, making sure that the products are produced at the highest levels. Moreover, the process could also be quicker, meaning that the product will be able to get to the market more efficiently. In addition, an automated system might be able to collect data

[5] Source: `https://help.sap.com/erp2005_ehp_05/helpdata/en/a6/df293581dc1f79e10000009b38f889/frameset.htm`, retrieved on 25.06.18

for every feature on all products which can be attractive to potential buyers, since the quality data is right there to be ascertained by consumers as they make their purchasing decisions. Hence, I consider that automating this process could be a beneficial choice.

While there is an upfront cost associated with an automated inspection system, I consider that this could be a one-time investment. The long term return on investment of the inspection system must be considered to adopt such automation. Over the lifetime of the system it could amount to significant cost-savings compared to the cost of labour associated with a manual quality inspection process.

Of course, in order to realize these potential benefits, the automated quality system must be set up and developed properly in the first place. It's necessary to spend time defining parameters for quality and testing that the models used by the system are production-ready. The automated system itself should also provide quality control of its decision-making process, allowing users to make necessary improvements and adjustments to avoid downtime.

Thus far in this section I have presented the context for enterprise quality management, overviewing the quality inspection process and the information system modules required to support such practice. I also narrowed down on a specific task amiable for improvements through automation. Such automation could be translated into a machine learning process that could be addressed as a use case for our study. In the next section I consider how this could be accomplished.

3.2.2 Business Understanding - Consider Machine Learning as an Alternative Solution

Du et al. reiterates the importance of quality in his work [DS06], authors remark that in today's competitive market, quality is a key factor for every industry, be it food, textile, automobile or others, because quality ensures success. In most of the industries, quality evaluation depends on manual inspection which is tedious, laborious, costly and sometimes influenced by physiological factors inducing inconsistent results. If quality evaluation is achieved automatically, production speed and efficiency can be improved in addition to the increased evaluation accuracy, with an accompanying reduction in production costs [SB03]. Computer vision systems have been used increasingly in the several industries for quality evaluation purposes from long ago [Sun00].

Timmermans states in [Tim95], that computer vision includes the capturing, processing and analyzing of images, facilitating the objective assessment of visual quality characteristics in products [Tim95]. Combined with an illumination system, a computer vision system is typically based on a computer in connection with electrical and mechanical devices to replace human manipulative effort in the performance of a given process. Mahendran et al. in their work [MJA12] specifically emphasize the importance of illumination systems

as a prerequisite of image acquisition for food quality evaluation. The quality of captured images can be greatly affected by the lighting condition. A high quality image can help to reduce the time and complexity of the subsequent image processing steps, which can in turn decrease the cost of an image processing system. Different applications may require different illumination strategies [MJA12]. [Nov85] report that most lighting arrangements could be grouped as one of the following: front lighting, back lighting, and structured lighting.

Sun [Sun16] compiles a large amount of work in computer vision for quality inspection in the food industry, with examples from meat, poultry, seafood, vegetables and other specific domains. The work covers, as well, some fundamentals on image acquisition, processing techniques and the classification task, as they pertain to quality inspection. Pau and Olafsson [Pau17] collect related work for the fish industry. Within their work authors define valuable features from a quality testing system, such as:

- Real-time dynamic response
- Multiprocessing of information
- Noncontact
- Accurate and repetitive
- High mean time between failures
- Low price
- Nontoxic
- Sterilizable
- Insensitive to electrical/electromagnetic interferences
- Physically robust

Authors establish that there are areas where automated quality inspection systems based on computer vision hold promise in their domain (e.g. in the detection of fillet defects), thanks to satisfying some of the valuable features that they establish, provided that the systems are well-tuned to the domain.

Computer vision systems comprise of two parts: a software component and a hardware component. For hardware implementations there are several viable options for image processing systems, such as application-specific integrated circuits, digital signal processors, and field programmable gate arrays. The real time capabilities of these systems depends on the software implementation because when image sizes grow larger and algorithms become more complex, the speed will be slower and may not satisfy the requirement for high speed in real-time systems. On the other hand, the speed of overall systems can be improved by

creating a custom hardware design. Therefore, hardware designers usually use some sort of PC programming environment to implement a design to verify functionality prior to a lengthy hardware design.

In this document I will focus on the software part of a computer vision system, with a more specific focus on the learning techniques adopted for a quality management classification task.

The learning technique is one of the essential features for quality evaluation using computer vision, as the aim of computer vision is to ultimately replace the human visual decision-making process with automatic procedures. Computer vision tries to mimic human behaviour in determining colour, content, shape, texture and relevant features during the inspection of a product. Backed up by powerful learning systems, computer vision provides a mechanism in which the human discernment process can be simulated artificially, helping humans in making complicated judgments accurately, quickly and consistently over a long period [MJA12]. Learning techniques can be employed to learn meaningful or nontrivial relationships (i.e., between input features and resulting decisions) automatically in a set of training data and produce a generalization of these relationships that can be used to interpret new, unseen data [MJA12]. Therefore, using sample data, a learning system can generate an updated basis for classification of subsequent data from a similar source [MJA12].

Figure 3.2, shows the general learning system configuration used in computer vision for quality evaluation. Using image processing techniques, the images of food products are quantitatively characterized by a set of features. These features serve as objective data to represent the products, which can be then used to form the training set. Once the training set has been obtained, the learning algorithm extracts the knowledge base necessary to make decisions for unknown cases. Based on this knowledge, intelligent decisions are made as an output, and they can be fed back to the knowledge base at the same time, thus generalizing how inspectors accomplish their tasks.

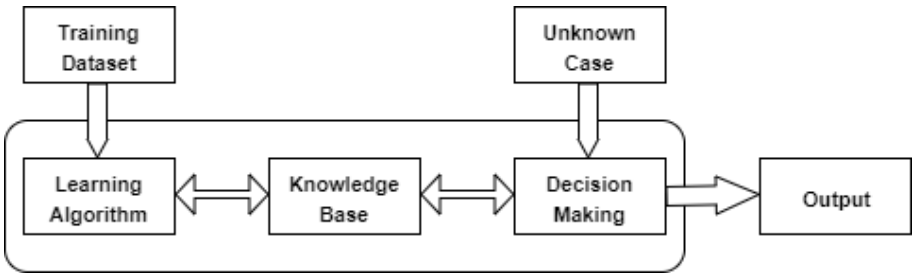

Figure 3.2: Learning System from [DS06]

After selecting, in order to answer the research questions, quality management as a domain

of application, and more specifically, the task of automated quality inspection, with a focus on the computer-vision-based learning for classification, I can now discuss the specific data information that I selected to use.

As stated previously I develop a learning system by using different SAP Leonardo Machine Learning components and answering previously detailed research questions along. Designing a learning system using SAP Leonardo Machine Learning components is a complex task and needs a variable amount of design thinking about the business scenario before executing things. As discussed before in Chapter 2 about SAP Leonardo components, there are alternative solutions available to solve a single problem, for example, in some scenarios foundational APIs could be useful saving development efforts and time, and, in others, building a system from scratch could be a long term solution for the problem. Here, I develop an image classification system which will learn from a use case-specific dataset. I adopt two implementations for the approaches, SAP HANA and TensorFlow integration, and SAP HANA PAL, since the two approaches align with research questions (with the former being a choice for the less-integrated approach and the latter being a choice for the more-integrated approach).

In the next subsection I introduce the specific data information that I selected to use. This subsection corresponds to the stage of Data Understanding in the CRISP-DM process model.

3.2.3 Data Understanding

In order to acquire data for a learning system, I created a data-set of 1765 images, since I was not able to find an open source dataset available that aligned precisely with the domain of automated computer-vision-based quality inspection.

The image dataset is created out of rubber toys which could be considered as a final product from a rubber toys vendor, who wants to automate their quality management - quality inspection process in sales and distribution. In our use case such vendor wants to perform quality inspections for delivery of a material or product before it leaves their premises. The vendor uses SAP S/4HANA as an integrated ERP system for their business and wants to leverage Machine Learning capabilities to gain competitive edge.

Figure 3.3 and figure 3.4 show an example from the dataset used for training, validation and testing the learning system. This dataset is created in naive settings without a specialized illumination system and without a specialized camera. I clicked 1765 images from a mobile phone camera at every 10 degree approximately, in clockwise direction with a white background under a yellow light. The dataset consists of 2 types of toys, that is, Blue Fish and a Ring, which are manually teared down to produce some highlighted structural dif-

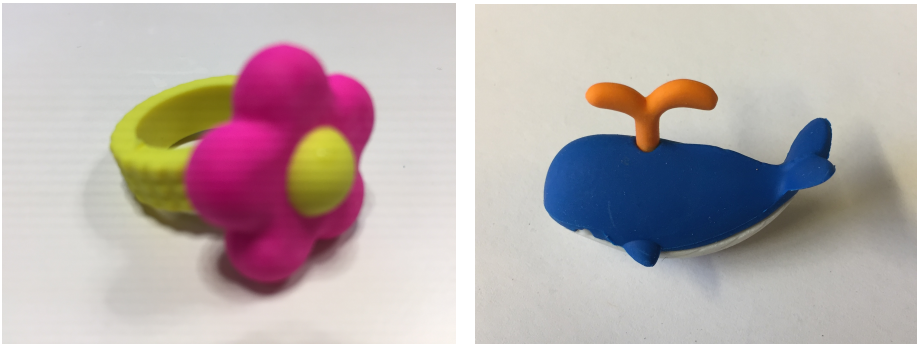

Figure 3.3: Images of Fine Products

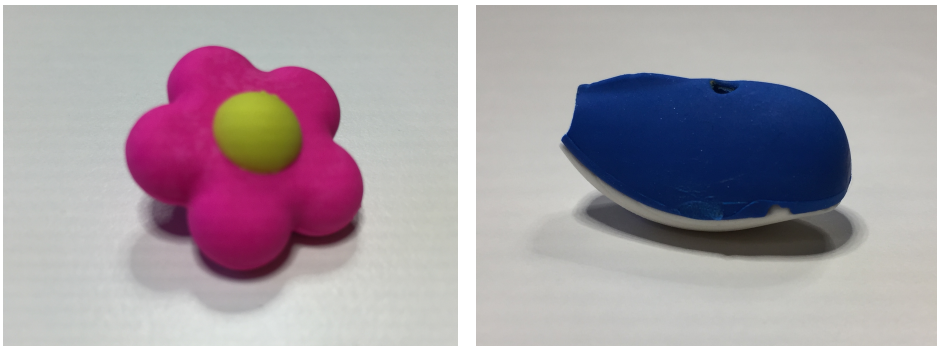

Figure 3.4: Images of Faulty Products

ferences. Each image has 24 bits of depth, indicating the number of bits used to represent each colour component of a single pixel. The images also have inconsistent lightening and some shadow effects. Table 3.1 shows some statistics of the data set.

Product	Defective	Good
Blue Fish	463	446
Ring	454	402
	917	848

Table 3.1: Data-set statistics

3.2.4 Data Preparation

Brosnan et al. state that image processing and image analysis are the core of computer vision systems [BS04]. Image processing involves a series of image operations that enhance the quality of an image in order to remove defects such as geometric distortion, improper focus, repetitive noise, non-uniform lighting and camera motion. Image analysis is the process of distinguishing the objects (regions of interest) from the background and producing quantitative information, which is used in the subsequent system for decision making [MJA12]. In computer vision tasks, the illumination system and the camera play an important role in the overall efficiency and accuracy of the system [MJA12].

Due to the lack of ideal conditions, the images in our data set were distorted and had shadows. The effect of these imperfections were addressed by using python 'openCV' library and, on occasions, through manual cropping of the images. In order to mitigate shadow and inconsistent lighting effects, the images were also converted to HSV and Gray scale from the original RGB scale.

To create a machine learning task based on these images, the images were grouped into two categories: Good and Defective, having 848 and 917 images respectively in each category (which is a good choice for classifiers, since both classes have a similar proportion of examples). Python is chosen for all data pre-processing tasks because of its extensive library collection, support for TensorFlow and availability of drivers for SAP HANA. The average resolution of the images was 3200 $\times$ 2400 which is compressed to 28 $\times$ 28 resolution by using Python Imaging Library which later resulted in a 784-D feature vector. The three data-sets (RGB, HSV, Gray) are converted to 3D arrays of the form (imageIndex, x, y), where imageIndex is the index of an image and x and y are the floating point values contained in the x and y coordinates. Since TensorFlow works on tensors and the SAP HANA PAL library works on tables, this choice of representation helps to keep data in a process-able format. For TensorFlow the array itself could be processed, for SAP HANA PAL, it was linearized into a large table, with 784 columns, with each row corresponding to the features of an image.

Each image was read into a two-dimensional array and normalized to a [-1,+1] value range. Each image was normalized to close to zero mean and small variance, shown in table 3.2, to make the training easier for the algorithms. The processed 3D arrays were pickled (i.e., serialized for storage) on disk, to avoid unnecessary processing before creating training, validation and test sets out of them. All this processing was done by using the SciPy and NumPy libraries from python.

	Dataset RGB		Dataset HSV		Dataset Gray	
	Defective	Good	Defective	Good	Defective	Good
Mean	0.174084	0.216296	-0.173253	-0.1414	0.174081	0.21619
SD	0.211176	0.220905	0.148323	0.145053	0.211218	0.220914

Table 3.2: Mean and Standard Deviation for normalized datasets

Out of the whole data set, 75% was kept for training and 25% was used as a test set. The process of selecting images in each set was randomized to avoid any bias into two classes and images from both the classes and products are well mixed together. To make sure high quality in the resulting sets when using SAP HANA PAL (i.e., considering that the random splitting was not immediately supported for the functionality that I used), I checked for overlapping in the training, validation sets for each data-set so that the model

will not process a single image data twice. Looking at the mean and standard deviation values, gray scale images were skipped for further processing as their mean and standard deviation values are approximately equal to RGB scale values.

Considering the statistics presented in table 3.2, and with the goal of speeding-up the process, I made the assumption that the information provided from the gray scale representation was similar to that included in the RGB one. Therefore, this representation was not adopted for further testing. Furthermore, gray scale is usually selected to improve the quality of images, in terms of shadows and inconsistent light; however in my implementation other pre-processing steps were adopted to address such issues.

Naturally, image processing is a complex domain, and there is a wide availability of options to improve the input images through the application of specialized filters and processing. Though such techniques could be leveraged to improve the classification results, they are orthogonal to my evaluation on the approaches to integrate machine learning with enterprise systems, and, as a result I adopted only the basic pre-processing steps described in this section.

I explain modelling steps for both approaches, that is, SAP HANA TensorFlow integration approach and SAP HANA PAL approach in the Chapter 4 : Evaluation, since they are highly tied to the evaluation.

3.3 Experimental Setup

In order to carry out the experiments, different environments were set for each approach. For the SAP HANA integration with TensorFlow, the Google Cloud Platform was used, serving as an infrastructure as a service, where external machine learning library-enabled SAP HANA 2.0 SPS 02 was instantiated. To implement TensorFlow, an Ubuntu 16.04 Xenial VM with 2 CPU's and 3.5GBs memory was instantiated where Python 3.5 was installed with TensorFlow (non-GPU) 1.3.0 was used with the corresponding Tenser Flow Serving version.

For the SAP Predictive Analysis Library approach, we worked on-premise. The Eclipse Oxygen.1a Release was used as HANA Client with HANA 1.0 SPS 12 as a database where PAL was installed , along with Python and the ODBC python driver (pyodbc) to communicate from Python with HANA. The experiments were conducted on a commodity multi-core machine running Suse Linux on x86_64 Enterprise Server 11 SPS 3 for SAP Application, with Intel Xeon E5-2686 v4 @ 2.3 GHz processor(32 cores) and 244 GiB of memory.
In addition, several Python libraries were used to aid the processing, such as SciPy and NumPy.

3.4 Summary

This chapter I established the necessary information on the prototypical implementation for the evaluation of my research questions. First, I defined the precise research/evaluation questions. Next, I carefully introduced the quality management use case to study both tightly integrated and less integrated approaches for adopting ML in an enterprise environment. I focused on a classification task, that I identified to be business relevant, from automated computer-vision based quality inspection. I also mapped each step in my study of the quality management use case, to project development stages within the CRISP-DM methodology, such that the results of my work could constitute a reproducible and meaningful artifact. I described the dataset used and the steps of data preparation. Although, I separate the modelling stages, to place them in 4: Evaluation, because of their high-coupling with evaluation, I briefed about several aspects of the experimental setup for both approaches, and I disclosed the versions of tools that I used.

4 Evaluation

This chapter elaborates the core stages of Modelling, Evaluation and Deployment from the CRISP-DM process model. The presentation of the results is guided by the research questions presented in Chapter 3, with results for the different approaches presented inline under different modelling subsections, and discussed in the closing sections of the chapter.

4.1 Modelling using TensorFlow

In this section, I present the TensorFlow model and its integration with the SAP HANA database using TensorFlow serving. This part of the chapter implicitly answers the first three evaluation questions, with the last three of them being discussed in later sections of this chapter (Sec. 4.3):

1. How can deep learning solutions be applied to enterprise data within a system like the SAP Landscape?
2. When adopting less integrated solutions, what is the impact of data movement?
3. What can be expected performance gains from such approach?

As mentioned before, the experiment was carried on the Google Cloud Platform. The experiment commenced with the setting up of all the required machine instances with the software and libraries mentioned before.

Prerequisite steps before modelling on the SAP HANA Database:

- Check EML installation in SAP HANA Database under AFL Package Library, logging in as SYSTEM user.
- Connect to tenant database, create a EML user in the tenant database with SYSTEM user.
- Provide MONITORING, CREATE REMOTE SOURCE authorization to the newly created user.
- Provide select, update, insert and delete authorization on EML MODEL CONFIGURATION to the new user, which is the central table for EML execution under _SYS_AFL schema.
- Authorize new user to create, delete and execute EML procedures.

- Log in to the database with new set up user.

Prerequisite steps before modelling on Ubuntu Machine:

- Configure Python.
- Install TensorFlow.
- Install TensorFlow serving.
- Configure the TensorFlow model server.

After going through the prerequisite steps for environmental setup on both HANA and Ubuntu machines, the training data was extracted from HANA as a TensorFlow Tensor (3D), then it was used for training a Fully Connected Neural Network written in TensorFlow (the details of the network are explained a bit further within this section), in the Ubuntu machine.

In order to develop a classification deep network I decided not to employ a pre-trained network (e.g. AlexNet, ResNet), since I considered that a simpler network could also be adopted. The development started from a simpler structure with no hidden layers and extended to more complex design introducing more hyper parameters. The final model consists of 6 hidden layers [150, 125, 100, 75, 50, 25] and was ran through 2000 epochs. Sigmoid Cross Entropy has been adopted as a Loss function with Sigmoid as an activation function at each layer. No convolutional layers were used. Due to over-fitting issues in the process, L2 Regularization has been applied in the final loss function with a regularization constant 0.002. I used a gradient descent optimizer. The model did not converge with the amount of training data but was able to obtain acceptable accuracy each time. The plot of loss value per iteration steps is shown in figure 4.1.

The final model was able to achieve training and test accuracy in the range of (82-85)% and (90-93)% respectively. The accuracy was comparable in case of the HSV image data set, reaching values in the ranges of (90-95)% and (85-90)% for training and test sets, respectively. These accuracies fluctuated due to randomization in initialization of weights and bias. Tables 4.1 and 4.2 display confusion matrix for TensorFlow model and table 4.3 shows precision, recall, f-measure and accuracy for the final model.

To use this model against unseen product images, the model is persisted in a directory and hosted using TensorFlow serving on the Ubuntu machine. On the HANA side, I create a remote source using a *gRPC* adapter.

A set of images was uploaded as BLOB images in a table, figure 4.2, each image was sent from HANA to TensorFlow for real time scoring, to get the predicted class by the served model in the format of model signature. For a fine-grained understanding I include

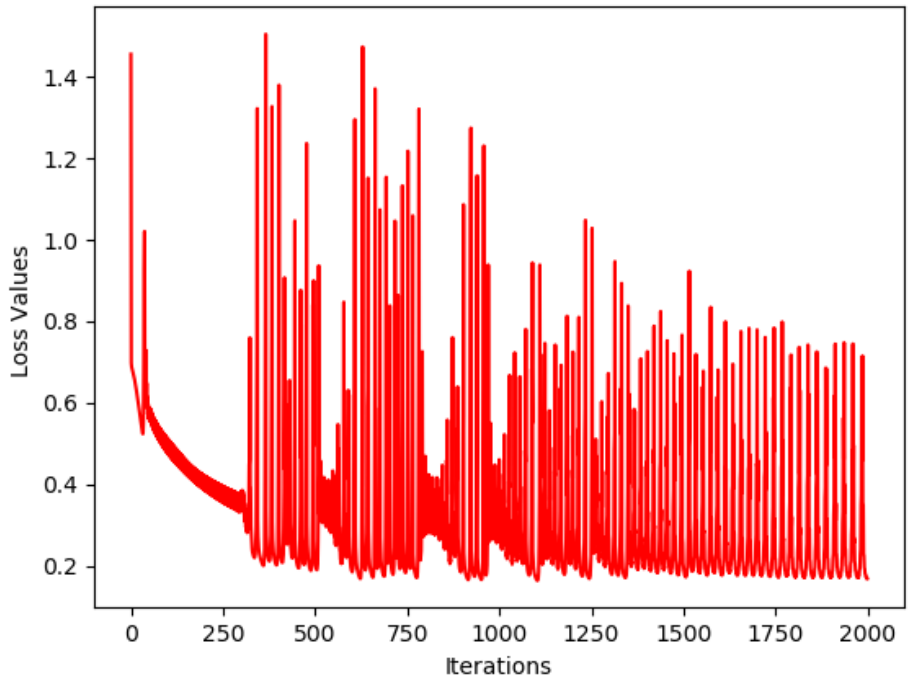

Figure 4.1: Loss vs Iteration Plot for Fully Connected Deep Neural Network

	CLASS 0 (Defective)	CLASS 1 (Good)
CLASS 0 (Defective)	172	33
CLASS 1 (Good)	5	200

Table 4.1: Confusion Matrix for FCN for RGB Dataset

results for a subset of 10 images (from the test set), for both HSV and RGB data sets. Table 4.4 shows the result of scoring done by the served model on this small selection, where, CLASS_GOOD and CLASS_DEFECTIVE are output classes and SCORE_1 and SCORE_2 are output probability scores.

The model was able to classify 9 out of 10 images correctly in both cases, that is, at each time a single image was sent to the model from HANA and it returned the class with its score. The scores were high in most out of the 9 correct classifications except for PRODUCT_ID 8 and 9 shown in figure 4.3.

	CLASS 0 (Defective)	CLASS 1 (Good)
CLASS 0 (Defective)	200	5
CLASS 1 (Good)	34	171

Table 4.2: Confusion Matrix for FCN for HSV Dataset

```
SELECT * FROM PRODUCT_QUALITY_INSPECTION
```

	PRODUCT_ID	PRODUCT_TYPE	CLASS	IMAGE
1	1	Ring	Defective	ÏÐÌÐÐxÐÆ...
2	2	Ring	Defective	ÏÐÌÐÐxÐÆ...
3	3	Blue Fish	Defective	ÏÐÌÐÐxÐÆ...
4	4	Blue Fish	Defective	ÏÐÌÐÐxÐÆ...
5	5	Blue Fish	Defective	ÏÐÌÐÐxÐÆ...
6	6	Ring	Good	ÏÐÌÐÐxÐÆ...
7	7	Ring	Good	ÏÐÌÐÐxÐÆ...
8	8	Ring	Good	ÏÐÌÐÐxÐÆ...
9	9	Blue Fish	Good	ÏÐÌÐÐxÐÆ...
10	10	Blue Fish	Good	ÏÐÌÐÐxÐÆ...

Figure 4.2: Sample test set in SAP HANA database for Real-time Scoring with FCN Model

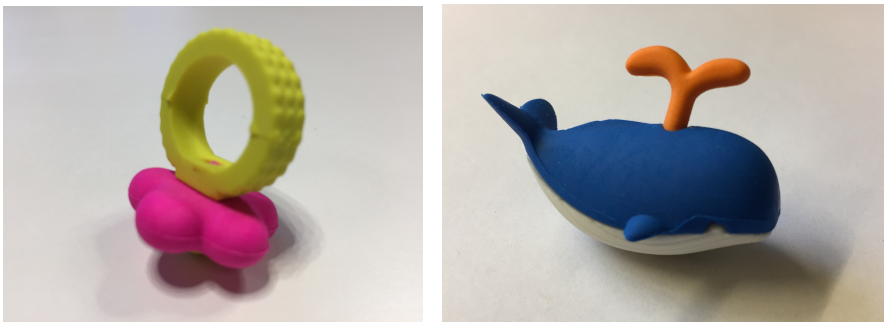

Figure 4.3: Images of Doubtful Cases for FCN Model

4.2 Modelling Using PAL

In this section, I present the SAP HANA PAL approach, using the PAL library from the SAP HANA AFL package. This part of the chapter implicitly answers these evaluation questions:

4. How can we adopt an integrated approach for ML in enterprise data, using SAP Leonardo?

5. What are the benefits achieved by reducing data movements?

SAP HANA PAL algorithms are invoked within SQL Script procedures to perform analytical tasks. They work on the HANA tables and each PAL algorithm is based on predefined table structures. As mentioned before, for experimentation Eclipse Oxygen.1a Release was used as a HANA Client and HANA 1.0 SPS 12 as the specific database.

	Precision		Recall		F1 Measure		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
Dataset RGB	0.971751	0.858369	0.839024	0.975609	0.900522	0.913241	90.73%
Dataset HSV	0.854700	0.971590	0.975609	0.834146	0.911160	0.897637	90.48%

Table 4.3: Metric table for FCN Model

PRODUCT_ID	CLASS_GOOD	CLASS_DEFECTIVE	SCORE_1	SCORE_2
1	0	1	0.97496903	0.02496817
2	0	1	0.97781139	0.02212004
3	0	1	0.98080253	0.01911775
4	0	1	0.98087453	0.01904558
5	0	1	0.97675937	0.02317348
6	1	0	0.88421577	0.11515981
7	1	0	0.77123570	0.22636595
8	0	1	0.61277002	0.38431492
9	1	0	0.56826382	0.42816525
10	1	0	0.80062800	0.19737215

Table 4.4: Real-Time Scoring Output of Served FCN Model in SAP HANA Database Table

```
-- CREATE USER FOR AFL DEVELOPMENT
CREATE USER [redacted] PASSWORD [redacted];
-- AUTHORIZE CREATION & REMOVAL OF PAL PROCEDURES
GRANT AFLPM_CREATOR_ERASER_EXECUTE TO [redacted];
-- AUTHORIZE EXECUTION OF PAL PROCEDURES
GRANT AFL__SYS_AFL_AFLPAL_EXECUTE TO [redacted];
-- AUTHORIZE READ ACCESS TO INPUT DATA
GRANT SELECT ON SCHEMA [redacted] TO [redacted];
```

Figure 4.4: SQL Script for PAL Prerequisites

Prerequisite steps before modelling on the SAP HANA Database:

- Check PAL installation in SAP HANA Database under the AFL Package Library, logging in as SYSTEM user.
- Connect to the tenant database, create a PAL user in the tenant database with SYSTEM user as shown in figure 4.4.
- Provide CATALOG READ access to the newly created user.
- Authorize the new user to create, remove and execute PAL procedures.
- Provide access to the working schema for the new user.
- Log in to the database with the new set up user.

Since PAL works on HANA tables, the normalized RGB and HSV image data was inserted into a HANA table in the form of arrays. A single image was stored in an array column of type double with length 784, within a HANA table. As a condition, the PAL classification algorithms accept columns with integer, double, varchar or nvarchar data types as inputs. In order to meet this requirement, the table with arrays was flattened and a new table was formed with 786 columns, including an ID column (Primary key), 784 columns for

	CLASS 0 (Defective)	CLASS 1 (Good)
CLASS 0 (Defective)	140	99
CLASS 1 (Good)	62	109

Table 4.5: Confusion Matrix for Logistic Regression Model for RGB Dataset

	CLASS 0 (Defective)	CLASS 1 (Good)
CLASS 0 (Defective)	155	84
CLASS 1 (Good)	55	116

Table 4.6: Confusion Matrix for Logistic Regression Model for HSV Dataset

the image data from arrays, and a "Class" column (Target Variable), which registered the class of a given image in the table.

In order to develop a PAL classification model, data was been prepared in a structured table format which qualifies for a simple binary class classification problem. I experimented with Back Propagation Neural Network (BPMN), Support Vector Machines (SVM) and Logistic Regression. Due to technical limitations[1], Logistic Regression was chosen as a suitable classification algorithm. I also selected it due to its ease for applicability, and small memory/execution requirements. Furthermore, logistic regression model in SAP HANA PAL gives output model in Predictive Model Markup Language (PMML) format which is simpler to handle in case of a large number of features, compared to BPMN's model output in the form of JSON.

For the deployment of the model, I referred to SAP web documents [SAP18d], some table types and model signatures were defined, and the data was ingested into the model by using a predefined procedure called 'LOGISTICREGRESSION'. The resultant model table which stores the trained model, was defined with a minor change, instead of using VARCHAR(5000) for the model content column, CLOB was used as its data type because VARCHAR(5000) is not sufficient to store the large textual content of a PMML model with a large number of features, such as, 784. The Newton iteration method was used with 1000 iterations to train the model in the configuration table. The 'LOGISTICREGRESSION' procedure provided three resultant tables. One giving the AIC statistics of the model. AIC stands for Akaike information criterion, a comparison statistic measure between models that deals with goodness of fit compared to the complexity of the model. This measure, however does not provide any information about how good the model fits to the data. The lesser the value the better model will be than another model. The final trained model had the value of *1.589* for both RGB and HSV dataset, making them comparable in their complexity and goodness of fit to the training data. The second table

[1]Though BPMN could've been a good candidate, in my experiments it was not possible to use BPMN given that the model was serialized as a large JSON object in memory, and given the high number of features (784), it was slow (compared to simpler, alternative models) to load into memory for exploring different network architectures. Possibly with more time a good configuration could have been achieved.

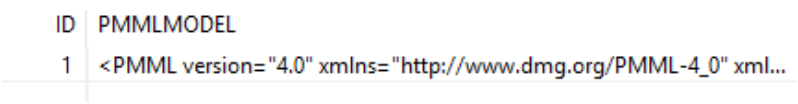

Figure 4.5: PMML Logistic Regression Model saved in a HANA Table

contains the actual PMML model, shown in figure 4.5, which I used during the prediction phase for inference on unseen data.

To find evaluation measures, the in-built 'CONFUSIONMATRIX' procedure was used to create a confusion matrix on the test set (the result is shown in tables 4.6 and 4.5) and calculate precision, recall, f-measure and accuracy of the developed model on the 25% of testing data, as shown in table 4.7. The accuracies achieved are 60.73% and 66.09% for the RGB dataset and the HSV dataset, respectively.

For a better, more fine-grained understanding of the results, I did also a prediction on the same sample of 10 images used for the TensorFlow model. The trained model signature acknowledges images as 784 feature values. For 10 images, a table view was created containing 784 column values. Both HSV data model and RGB data model were able to predict 8 out of 10 images correctly, as shown in table 4.8 where 'ID' refers to the PRODUCT_ID in the test set, 'Fitted' refers to the logistic model's predicted value and 'Type' refers to the output class of the image, 0 being 'Defective' and 1 being 'Good'. PRODUCT_ID 3 and 5 got wrong predictions from the models, shown in figure 4.6.

	Precision		Recall		F1 Measure		Accuracy
	Class 0	Class 1	Class 0	Class 1	Class 0	Class 1	
Dataset RGB	0.693069	0.524038	0.585774	0.637426	0.634916	0.575196	60.73%
Dataset HSV	0.738095	0.58	0.648535	0.678362	0.690422	0.625335	66.09%

Table 4.7: Metric table for Logistic Regression Model

ID	Fitted	Type
1	0.00	0
2	0	0
3	1	1
4	0.00	0
5	0.99	1
6	0.99	1
7	1	1
8	0.99	1
9	1	1
10	1	1

ID	Fitted	Type
1	0.00	0
2	0.00	0
3	0.99	1
4	0.00	0
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	1	1

Table 4.8: Prediction Result for HSV (left) and RGB (right) dataset using trained Logistic Regression Model in a HANA Table

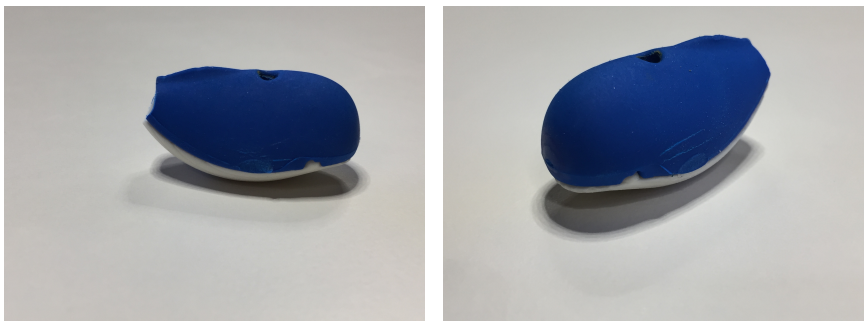

Figure 4.6: Images of Doubtful Cases for Logistic Regression Model

4.3 Results and Discussion

In this section, as a compilation of my findings, I present a comparison of both tightly integrated and less integrated approaches, that is, the SAP HANA and TensorFlow integration approach, and the SAP HANA PAL approach, according to criteria defined in this section, as a conclusion from my study.

In this section I specifically answer to the following evaluation question:

6. What are the trade-offs between integrated and less integrated approaches, such that scientists could make informed choices about what approach to adopt?

With this section I provide my answers to all the original research questions, such as, *the impact of data movement*, *expected performance gains from integrated approach*, *benefits achieved by reducing data movements*, and *a trade-off comparison between both approaches.*

Deciding on an implementation approach and the infrastructure required is a foremost consideration, after requirement gathering and data understanding in any machine learning project in the business landscape. Here, I highlight some points from both the approaches, in order to establish a trade off comparison between the two.

This comparison is created based on the experience gathered during the implementation phase of both approaches and may hold importance for guiding a team in deciding about the infrastructure for their project. The criteria that I establish involves the following: *Performance of the Model*, *Execution Speed of the Process*, *Data Movement*, *Model Governance*, *State-of-the-art Methods*, *Ease in Building the Solution* and *Integration with Existing System.* I establish a decision matrix towards the end of this section including the specified criteria and scores that, through my study, each evaluated approach got with respect to criteria. The comparison is as follows:

Performance of the Model

The performance of the model is highly dependent on data quality. A direct comparison of both the models' performance shows that the FCN model is able to reach F-measures of 0.9

and 0.91 for the RGB and HSV datasets, table 4.3, respectively (assuming the defective to be the positive class), while the logistic regression model is able to reach F-measures of 0.63 and 0.69 for the RGB and HSV datasets, table 4.7, respectively (assuming the defective to be the positive class). Hence our evaluation shows a case where the FCN model is able to perform better than logistic regression, as can be expected[2].

The FCN model learns a non-linear representation of the data because of the multiplicative relationship between the weights among the layers and the sigmoid activation functions, and Logistic Regression learns a linear representation of the data. So, the FCN model is highly capable of representing complex data in comparison to Logistic Regression.

Execution Speed of the Process

The TensorFlow deep neural network models includes high-end computations because they process on the principle of tensors and matrix multiplications. As the network will go deeper with increases in size of the data or learnable features, the training process is going to be slower. This can be compensated by the use of GPUs or distributed computations, but that comes at additional costs. For the Quality Management use case, 2 CPU's with 3.5 GBs Ubuntu machine was used on the Google Cloud Platform. It took 15 minutes on an average to train several models with different numbers of layers and nodes with a small data-set.

On the other hand, PAL's Logistic Regression model was better in terms of training time but it took very long time during the data preparation process. As specified in above sections, 784 features of the images were inserted into HANA in an array and the array elements were segregated to make it compatible with the PAL algorithm's inputs. It took around 8 hours for 1375920 commits in a single table using dynamic SQLs. Perhaps batch-wise processing could've improved this runtime. Furthermore, there is a scope for using better programming methodologies involving 2-D arrays in HANA, which may improve this time.

I do not provide a quantitative comparison for the cost of training the models under both approaches, due to the fact that I was using different computing platforms, to truthfully represent the use cases.

Data Movement

In the context of machine learning, data movement holds importance when dealing with huge amounts of data and real time data insights are needed from it. Data movements lead to data duplication, resulting in no single source of truth and data security exposures in analytical systems. In contrast, eliminating data movements may lead to reducing the

[2]Since many years ago, specialized neural networks have outperformed simpler models at image classification tasks (see [KSH12]). Though the network that I implement is not particularly complex, I am able to report a similar observation.

total cost of ownership and a faster delivery of enterprise-wide analytics solutions.

As big data expands over the three V's : velocity, volume and variety, eventually at some time it becomes impractical to move large data amounts to separate servers for the data analysis.

Further advantages of not having to move the data are: the elimination of load and transform times, and a reduced risk of something going wrong in the translation between the source data and the predictive analytics tool.

Data Movement is one of the focal point of this work and an important aspect in solving machine learning tasks which directly affects data dependencies and several other systems integration in enterprise landscape. In the TensorFlow approach data needs to be moved or transfered on a separate server where training will happen and the trained model communicates with HANA for acquiring new data and gives back model results to HANA. It involves additional costs for new server and network and firewall setups. In the presented use case, the dataset size is 1.5 GBs which is compressed (by reducing the image quality) before ingestion in the model. In real world problems datasets are much larger in size and transferring data to other servers for processing is time-consuming, adding more costs to the projects and leading to several data management issues.

As stated above, in my experiments, loading the data into the model was not a specially time consuming process, taking a matter of 2 or 3 minutes for transfering into HANA as a BLOB object and then into TensorFlow. On the other hand, the PAL approach works within HANA using in-database machine learning which is beneficial if data is already residing in HANA. So, there is no need to move the data silos to an external system. Models trained on larger data might be necessary to better study the impact of data movement.

Data movement is a factor that affects both the training and the real-time inference done during production, with the different data access patterns for each.

For training, the data access patterns depend on the training algorithm of the machine learning model. Some models, for example, can be trained with k-folds cross-validation, which requires several passes over the data while other models can be trained with a single-pass training algorithm. If the training data is sufficiently large, it might not be feasible for the machine running the machine learning algorithm to contain it all, and instead it could be queried from a database several times. For multi-pass processes this could lead to very large data movement.

Real-time inference during production might be sufficiently served by supporting one inference-at-a-time/record-at-a-time. Therefore it involves no data reuse during the in-

ference, and small movements per individual inference, though large ones as the inference requests grow.

Once again, due to the incomparability of the platforms selected (i.e., our SAP HANA PAL approach ran on-premise, whereas the SAP HANA- TensorFlow integration ran on the Google Cloud Platform, without us being able to control the communication costs between the resources), in addition to the requirements for repetitions and reproducibility, I do not provide exact quantitative measures for the costs of data movements.

Finally, other than data movement I have found that changing data formats to match the machine learning models can e a time consuming process. For example, moving from a BLOB to TensorFlow was a speedy task, but changing the images into a processable format for PAL (i.e., changing each into an array of features to be stored in HANA) was a time consuming process, taking hours, as stated in the previous section.

Model Governance

Makhtar et al. describes the process of creating predictive models [MNR10], which includes steps of Data Preparation, Data Reduction, Data Modelling and Prediction, Evaluating and Validating of the model and Implementing and Maintaining the model. The effective management of all the predictive modelling steps is collectively called Model Governance.

TensorFlow and TensorFlow serving provide a stable set of classes and methods for model training, exporting the model and hosting the exported model for use. Tensorflow Serving supports various versions of the same model and can publish them. On the other hand, with HANA Native Libraries the whole process of model development is standardized with a set of parameters attached to each algorithm, which can be tweaked for fine tuning of the models, the models furthermore be exported through extensive formats to represent them, for example, PMML, JSON etc, hence these models can be shared across applications. Additionally, the predictive factory, a model management tool from SAP can automate the management of predictive models developed by the standard SAP Data Science tools. Therefore I consider that, in terms of Model Governance, both approaches are comparable, except that the less integrated approach might leave the model outside of the scope of enterprise platform. This could be a drawback.

Auditability of the models can also be considered to be an aspect that pertains to model governance. A precise study of this aspect, considering different machine learning models and the ability of the selected tools to support evaluation and explanations for the predictions, is outside the scope of this Thesis.

State-of-the-art Methods

The support for state-of-the-art methods in a given task, either through supporting users

in building them or in offering implementations of them, can be a crucial factor in terms of deciding between ML platform offerings.

Advanced Deep Learning, in the past few decades, has been proven to be the state of art technology for image classification, object recognition, speech recognition, and other domains where data contains a hidden structure from which it is difficult to extract the most informative features [DY+14].

TensorFlow is one of the most stable enterprise deep learning frameworks, and it can be adopted, alongside current enterprise offerings, to implement and test state of the art methods in an enterprise context, when these methods are relatively new and not available from enterprise offerings.

Given that machine learning techniques are being continuously developed, it could be the case that enterprise offerings lag behind what current data scientists are evaluating. Therefore it is important for enterprise offerings to support sufficiently expressive tools, that enable scientists to test new concepts. TensorFlow is a good tool for such comparisons.

It should be noted that evaluating the goodness of an ML framework in supporting state-of-the-art methods for a task, could be a contentious process. Hence, clear guidelines and standards should be developed for this process.

Ease in Building the Solution
Ease in building a solution directly affects the development time and effort put-in to build the solution. More standardized solutions take less development time and are easier, in some scenarios this approach may be useful in obtaining a quick solution for the problem. SAP HANA PAL algorithms are one of them which can provide a stable and quick solution for a machine learning problem.

On the other side, TensorFlow Deep Models are flexible and the final structure of the model is based on experimental results mostly. The number of parameters are flexible and solely depends on the developer which provide extra freedom to the developer, but this requires good data science skills to develop an accurate model that fits the data perfectly.

In my scoring for this criteria for both approaches I reflect my qualitative assessment, comparing man-month hours in the development, in contrast to work in other tasks. A careful study employing software engineering approaches, to measure developer effort, could be applicable offering more quantitative measurements to support the scoring criteria.

Integration with Existing System
The integration with the existing system is a disputable topic and solely depends on the

		Scores	
Criteria	**Weightage**	**SAP PAL Approach**	**SAP EML Tensorflow Approach**
Performance of the Model	*TBD*	*3*	*5*
Execution Speed of the Process	*TBD*	*4*	*4*
Data Movement	*TBD*	*5*	*3*
Model Governance	*TBD*	*5*	*2*
State of the Art Methods	*TBD*	*3*	*5*
Ease in Building the Solution	*TBD*	*3*	*3*
Integration with Existing System	*TBD*	*5*	*3*

Table 4.9: Decision Matrix representing Criteria, Weights and Scores for the Approaches studied

existing infrastructure. Both External Machine Learning (TensorFlow) and the Predictive Analysis Library approaches can be integrated with the SAP HANA database, but the question arises which one is better in terms of execution methodologies. I mentioned the point of data movement, in real ERP system data lays in silos and performing a data migration on a different server constitutes an additional resource-consuming (i.e., processing time, network bandwidth) task, which is required in the TensorFlow approach. SAP HANA PAL works on the data within SAP HANA or S/4HANA and runs natively inside HANA, so integration-wise this amounts to a better approach.

Decision matrixes are a commonly used tool to suppport multi-criteria decision making [Tri00]. Table 4.9 represents a decision matrix which displays criteria, weights (to be included by decision-makers, according to how important each criteria is to their scenario) and scores given to both approaches. The weightage represents the importance of each criteria to the decision-makers, where the total sum can be made to be 1. Weights are to be assigned by users, such that they can calculate a total score on each alternative, to guide their decision. Scores correspond to the results of my study, ranking how each approach fares with respect to the defined criteria. They are assigned on a scale of 1 to 5, where '1' is minimum and '5' is maximum.

From the scores it can be inferred that tightly integrated approaches can be preferred when the decision making favors the reduction of data movement, model governance and integration with an existing system.

Less integrated approaches can hold a competitive edge for performance of the model and the use of state of the art methods. There is, in fact, an underlying relationship between these aspects, since the performance can be expected to be correlated to the use of state of the art methods, often requiring support for users to build those (given the continuous advances in ML techniques for diverse tasks). In addition, ease of building the solution and the execution speed of the process do not seem to be core criteria distinguishing the approaches, according to the evaluation carried out in this work. Hence, the scoring based on my study and the criteria defined, encapsulate the findings of this study with respect to the trade-offs between systems embodying the approaches under study, for machine

learning in an enterprise landscape.

These conclusions hold, according to my evaluation, relying on quantitative and qualitative analytical assessments, for the SAP PAL and the SAP EML TensorFlow approach; further work is required to include other technologies, as representatives of the approaches.

4.4 Summary

In this chapter, I provide answers to the research questions framed for my study, based on the use case determined in the previous chapter. With this chapter I accomplish the CRISP-DM stages of Modelling, Evaluation and Deployment. First I decomposed the research questions into more fine-grained evaluation questions, such that the presentation of the evaluation could be guided through answering these questions. Next, I presented the results for the implementation of the less integrated approach, using SAP HANA and TensorFlow, with a neural network model that I developed. I explained the modelling steps, involving how the images from the use case were stored in HANA. I described the steps of the implementation, and the parameters required, such that my results are reproducible. I provided an evaluation on the performance of the model under the training dataset and showed examples of cases where it was not able to predict correctly. Next, I presented the results for the tightly integrated approach, I explained in detail the choice for the model (i.e., a PAL logistic regression model) and presented its performance over a test dataset. Subsequently I compared the results for both models and discussed how they fared with respect to goodness criteria. I explained why it was not reasonable to study data movement in a quantitative manner (i.e., it was not comparable to measure runtime when comparing different platforms), as a result, I provided a qualitative comparison of this aspect between the approaches. To conclude I establish a decision matrix including the decided criteria and scores for each approach, with respect to the criteria, based on quantitative and qualitative assessments disclosed in this chapter.

In the next chapter I conclude this Thesis, by collecting the findings of my work, establishing some threats to validity of my study and proposing future work in evaluating machine learning for an enterprise context.

5 Conclusion

In this chapter, I conclude this Thesis by summarizing the essential aspects of my work pertaining to machine learning challenges in enterprise context. I describe the use case that I selected from quality management, the approaches that I studied, my implementations and the key takeaways, part of which are embodied in a decision matrix that I propose to score the approaches based on criteria identified from my study.

I also disclose some threats to the validity of my results, and I scope possible future works in this research subject.

5.1 Conclusion

The growing importance of machine learning technologies in the enterprise context can not be questioned. The businesses are ready to consume these technologies either in improving their long standing business processes or in forming new functionalities to gain competitive edge. But, incorporating these new technologies into old landscapes is not a trivial task. Due to these challenges, the emergence of Machine Learning Platforms took place, with some of them addressing the specific enterprise context. In this work, I sought to compare tightly integrated and less integrated approaches for machine learning in an enterprise landscape. To this end I focused on the SAP Leonardo Machine Learning Portfolio, which includes several libraries, products and a cloud platform, considering its high potential to influence, in the near future, the 378000 small and medium sized enterprises using SAP across the globe.

The purpose of this thesis was to assess some existing approaches (tight-integrated and less-integrated) to mitigate enterprise machine learning challenges that occur in an enterprise landscape. The focal point of enterprise platforms is to circumvent data movements during machine learning tasks when used over enterprise data. Less-integrated solutions, though competitive, are susceptible to bad performance in the enterprise context.

In my study I define two clear research questions about comparing these two approaches. I start by providing necessary background to understand the domain. Next, in lack of an existing benchmark for my study, I establish a machine learning use case. To this end I adhered to a widely used design science research methodology called CRISP-DM, from the machine learning domain. I followed CRISP-DM's processing stages in order to develop a

credible artifact for a use case that I propose. More specifically as a use case I present a functional business case based on the SAP Quality management module, and I propose a possible automation with computer vision, for the often tedious and error prone Quality Inspection Process.

Following CRISP-DM I document my stages of business understanding, data understanding and data preparation. Next, I perform modelling with the integrated SAP HANA PAL library and the less integrated SAP HANA EML library adopting TensorFlow. In order to evaluate the selected approaches I created a dataset of 1765 images of rubber toys, both defective and in good state, under diverse angles and lighting conditions. They are intended to represent final products of a toy vendor using SAP systems.

For the implementation stage, I segregated the data set into two categories : GOOD and DEFECTIVE and formed a machine learning classification problem. The raw dataset underwent data preprocessing and several transformations for both approaches. I built a Logistic Regression model from SAP HANA PAL to represent an integrated solution and I developed a fully connected neural network model using TensorFlow to represent the less integrated solution in an SAP landscape. In order to answer the research questions I decomposed them into finer-grained evalaution questions. I presented my achieved results inline with the two modelling subsections, answering the evaluation questions. I found comparable results in the learning task for both approaches based on the dataset that I had.

During evaluation of results, it was found that both the approaches are good specific to machine learning problems they are applied to.

The major finding, in terms of performance at the learning task, was that the logistic regression from PAL library was able to solve a machine learning problem involving unstructured data after some data preparation process. This can provide an increase in return on investment to the end users. Although, its accuracy was less in comparison with a fully connected neural network, it still gave a satisfying number which could be applied for some business scenarios.

Towards the end, I presented a trade-off comparison between opted approaches from a broader perspective. I established a decision matrix based on the criteria selected earlier, where a user can assign weights to each criteria based on their requirements and use provided scores to get a quantified measure for both alternatives. The scores assigned on my quantitative and qualitative assessments, which naturally are subject to threats to validity (discussed in Sec. 5.2), but nonetheless convey the results of my practical study. From my study, the less integrated approaches were found to hold a competitive edge for performance of the model and the use of state of the art methods. Whereas, integrated approaches are ahead in terms of reduction in data movement, model governance and in-

tegration with existing infrastructure. The resulting decision matrix can help the end user to get quantifying measures for both integrated and less integrated approaches. Thus this artifact represents my answer to the research and evaluation questions established at the beginning of my work.

5.2 Threats to Validity

In this section, I acknowledge some threats to the validity of my findings. They are as follows:

- Internal threats:
  - The data generation process in my study is naive and was carried out with a mobile device, in contrast, for actual implementation for quality management use case through images, a specialized camera and illumination system should be opted. This has a direct effect on the data quality in the data sets, and it might have lead to improper model accuracies.
  - The data processing step, where the quality of the images was reduced such that a manageable set of features could be selected, could also impact the quality of the evaluation.
  - The selected data set size is small. It is a proven point that machine learning algorithms perform better with more examples. Deep learning is even more sensitive towards the amount of training examples provided to them for supervised learning.
  - I developed a model from scratch for the selected use case to develop a model which suits the data set. But this is not a necessary step. Some well-known convolutional neural networks have proven good performance in Image classification tasks and they can be used directly if they are aligned with the problem domain. Another approach could be Transfer Learning [PY10] where knowledge gained from one machine learning task of one domain of interest can be reused or applied in solving another but related task. This approach is useful while dealing with small data sets. Employing already built model or implementing transfer learning may lead to gains in development and training times. This choice could have lead to better performance in the classification task for the less-integrated approach.
- External threats:
  - Observations referring to state of art in SAP PAL, as they pertain to the resulting decision matrix score provided in my work, are based on the current set of algorithms and might change in future. Specifically, the ability to support users in implementing novel methods, or in choosing other state of the art techniques, might change.

- The criteria proposed in the resulting decision matrix was produced by my evaluation, and my literature review. Though appropriate for my observations, it might not be comprehensive. A survey with quality management experts and practitioners of automated quality management techniques could help improve my criteria, and serve to provide future guidelines for further evaluations of technologies and approaches.
- Though I compare the performance of models in both approaches, it is possible that the adoption of alternative models could lead to different observations with regards to the performance in the classification task.
- Due to my test designs, and to the time limitations of this project, for some cases I was only able to provide scores for some criteria based on qualitative assessments. Further studies could benefit from designing tests in such a way that repeatable and fair quantitative measurements could be used to support the decision matrix scoring. It should be noted that there are specialized aspects, from the criteria that I establish in the decision matrix, which require a more specialized evaluation, such as the auditability of models, within the model governance criteria. The evaluation for support of state-of-the-art methods is a criteria that could be particularly contentious to evaluate, hence, clear guidelines and standards for their evaluation are required. In addition, software engineering approaches could be employed, to complement my scores given by qualitative assessment to the criteria of ease for building the solution and integration to existing system.
- My research in comparing approaches for machine learning on enterprise data was guided by my choice of a use case from quality management, and specifically quality inspection with computer vision. An evaluation on other enterprise use cases could benefit this research area, and help to provide more general results to complement my observations.

5.3 Future Work and Concluding Remarks

As concluding remarks, I believe that I successfully accomplished my aim to study less integrated and tightly integrated approaches with respect to enterprises in several dimensions. Naturally, there are few shortcomings of my work as stated in section 5.2, which are all worthy of being addressed in future work.

The major difference in integrated and less integrated approaches for machine learning in SAP Landscape is data movement. Data Movement is one of the crucial factors while employing machine learning for automating traditional business processes which leads to high processing time because of training data transfer to separate machines. In less integrated approaches, such as, TensorFlow or R, data transfer takes place, raw form in case of TensorFlow and data frames in case of R. But at the same time these external libraries

provides an extensive set of packages and functions which simplifies data analysis tasks. Apart from data movement I have observed that changes in data format (which can be required by the ML models) can be time consuming, and hence should be considered as another criteria to be added into the decision matrix that I have proposed.

On the other hand, SAP HANA native libraries, such as, Predictive Analysis Library and Automated Predictive Library offers a set of frequently used machine learning algorithms. But, there has not been a lot of development towards extending this set from the stakeholders recently. It can be extremely beneficial if high end computational deep learning algorithms, such as, convolutional deep networks, recurrent neural networks, etc. can leverage the processing power of SAP HANA in the near future, which will lead to high return on investment for business users. Similarly, the offering of libraries could be complemented with alternatives that allow users to build their own models. I believe that future work should study this.

Bibliography

[AAB+16] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[ABC+16] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[AKNN+18] Mahmoud Abo Khamis, Hung Q Ngo, XuanLong Nguyen, Dan Olteanu, and Maximilian Schleich. In-database learning with sparse tensors. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 325–340. ACM, 2018.

[BBC+17] Denis Baylor, Eric Breck, Heng-Tze Cheng, Noah Fiedel, Chuan Yu Foo, Zakaria Haque, Salem Haykal, Mustafa Ispir, Vihan Jain, Levent Koc, et al. Tfx: A tensorflow-based production-scale machine learning platform. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1387–1395. ACM, 2017.

[BNS+18] Ran M Bittmann, Philippe Nemery, Xingtian Shi, Michael Kemelmakher, and Mengjiao Wang. Frequent item-set mining without ubiquitous items. *arXiv preprint arXiv:1803.11105*, 2018.

[BÖ14] Yalin Başţanlar and Mustafa Özuysal. Introduction to machine learning. In *miRNomics: MicroRNA Biology and Computational Analysis*, pages 105–128. Springer, 2014.

[Bro16] Jason Brownlee. What is deep learning? `https://machinelearningmastery.com/what-is-deep-learning/`, 2016. Accessed: 2018-06-16.

[BS04] Tadhg Brosnan and Da-Wen Sun. Improving quality inspection of food products by computer vision—-a review. *Journal of food engineering*, 61(1):3–16, 2004.

[CDD+09] Jeffrey Cohen, Brian Dolan, Mark Dunlap, Joseph M Hellerstein, and Caleb Welton. Mad skills: new analysis practices for big data. *Proceedings of the VLDB Endowment*, 2(2):1481–1492, 2009.

[CKNP17] Lingjiao Chen, Arun Kumar, Jeffrey Naughton, and Jignesh M Patel. Towards linear algebra over normalized data. *Proceedings of the VLDB Endowment*, 10(11):1214–1225, 2017.

[CP15] Kenneth Chu and Claude Poirier. Machine learning documentation initiative. In *CONFERENCE OF EUROPEAN STATISTICIANS*, 2015.

[Dad18] Abdel Dadouche. Sap leonardo machine learning foundation demystified. `https://wiki.scn.sap.com/wiki/download/attachments/473963058/2018-01-13%20-%20sitWDF%20-%20Demistify%20SAP%20Leonardo%20Machine%20Learning.pdf?version=1&modificationDate=1516116490000&api=v2`, 2018. Accessed: 2018-01-28.

[Din17] Thomas Dinsmore. Deep learning: A guide for enterprise architects. white paper, cloudera. `https://www.cloudera.com/content/dam/.../whitepapers/Cloudera_Deep_Learning.pdf`, 2017. Accessed: 2018-06-16.

[DS06] Cheng-Jin Du and Da-Wen Sun. Learning techniques used in computer vision for food quality evaluation: a review. *Journal of food engineering*, 72(1):39–55, 2006.

[DY+14] Li Deng, Dong Yu, et al. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

[Fie16] Noah Fiedel. Running your models in production with tensorflow serving. `https://research.googleblog.com/2016/02/running-your-models-in-production-with.html`, 2016. Accessed: 2018-05-13.

[FKRR12] Xixuan Feng, Arun Kumar, Benjamin Recht, and Christopher Ré. Towards a unified architecture for in-rdbms analytics. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 325–336. ACM, 2012.

[GR12] John Gantz and David Reinsel. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView: IDC Analyze the future*, 2007(2012):1–16, 2012.

[Gre17] ERP Great. Quality inspection process flow. `http://www.erpgreat.com/quality/quality-inspection-process-flow.htm`, 2017. Accessed: 2018-02-11.

[HRS+12] Joseph M Hellerstein, Christoper Ré, Florian Schoppmann, Daisy Zhe Wang, Eugene Fratkin, Aleksander Gorajek, Kee Siong Ng, Caleb Welton, Xixuan Feng, Kun Li, et al. The madlib analytics library: or mad skills, the sql. *Proceedings of the VLDB Endowment*, 5(12):1700–1711, 2012.

[IKB+18] Carlie J. Idoine, Peter Krensky, Erick Brethenoux, Jim Hare, Svetlana Sicular, and Shubhangi Vashisth. Magic quadrant for data science and machine-learning platforms. *Gartner Research*, 2018.

[Kas17] Patanjali Kashyap. Machine learning for decision makers. 2017.

[KBY17] 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.

[KJY+15] Arun Kumar, Mona Jalal, Boqun Yan, Jeffrey Naughton, and Jignesh M Patel. Demonstration of santoku: optimizing machine learning over normalized data. *Proceedings of the VLDB Endowment*, 8(12):1864–1867, 2015.

[KKL14] David Kernert, Frank Köhler, and Wolfgang Lehner. Slacid-sparse linear algebra in a column-oriented in-memory database system. In *Proceedings of the 26th International Conference on Scientific and Statistical Database Management*, page 11. ACM, 2014.

[KNP15] Arun Kumar, Jeffrey Naughton, and Jignesh M Patel. Learning generalized linear models over normalized data. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1969–1984. ACM, 2015.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[Lef17] Greg Lefelar. High-performance storage systems : Answering the data explosion with massive scale and compelling economics, a white paper by jeskell systems. `https://cdn2.hubspot.net/hubfs/403983/Jeskell%202017WP1%20HPSS%20High%20Performance%20Storage%20System.pdf?t=1495139256891`, 2017. Accessed: 2018-06-16.

[LGG+18] Shangyu Luo, Zekai Gao, Michael Gubanov, Luis Leopoldo Perez, and Chris Jermaine. Scalable linear algebra on a relational database system. *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[M+97] Tom M Mitchell et al. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877, 1997.

[MJA12] R Mahendran, GC Jayashree, and K Alagusundaram. Application of computer vision technique on sorting and grading of fruits and vegetables. *J Food Process Technol S1-001. doi*, 10:2157–7110, 2012.

[MNR10] Mokhairi Makhtar, Daniel C Neagu, and Mick Ridley. Predictive model representation and comparison: Towards data and predictive models governance. In *Computational Intelligence (UKCI), 2010 UK Workshop on*, pages 1–6. IEEE, 2010.

[Nov85] Amir R Novini. *Fundamentals of machine vision component selection.* Society of Manufacturing Engineers, 1985.

[Pau17] Louis-François Pau. *Fish quality control by computer vision.* Routledge, 2017.

[PRWZ17] 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.

[PTRC07] Ken Peffers, Tuure Tuunanen, Marcus A Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77, 2007.

[PY10] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[RD12] Florin Rusu and Alin Dobra. Glade: a scalable framework for efficient analytics. *ACM SIGOPS Operating Systems Review*, 46(1):12–18, 2012.

[RPC06] S Rajasekar, P Philominathan, and V Chinnathambi. Research methodology. *arXiv preprint physics/0601009*, 2006.

[SAP16] SE SAP. Sap hana r integration guide. `https://help.sap.com/doc/6f2ff4c50f7e4e4d90b93aa33652d063/2.0.00/en-US/SAP_HANA_R_Integration_Guide_en.pdf`, 2016. Accessed: 2018-01-29.

[SAP17a] SAP. Sap cash application. `https://www.sap.com/germany/products/cash-application.html`, 2017. Accessed: 2018-01-30.

[SAP17b] SE SAP. Sap global corporate affairs, corporate factsheet 2017. `https://www.sap.com/corporate/de/documents/2017/04/4666ecdd-b67c-0010-82c7-eda71af511fa.html`, 2017. Accessed: 2018-06-24.

[SAP17c] SE SAP. Sap hana external machine learning library (eml) reference. *SAP HANA Platform 2.0 SPS*, 2, 2017.

[SAP17d] SE SAP. Sap hana predictive analysis library (pal) reference. *SAP HANA Platform 2.0 SPS*, 2, 2017.

[SAP17e] SE SAP. Sap leonardo machine learning foundation. `https://help.sap.com/viewer/p/SAP_LEONARDO_MACHINE_LEARNING_FOUNDATION`, 2017. Accessed: 2018-01-31.

[SAP17f] Studio SAP. Deliver all analytics for all users through a single product in the cloud. https://www.sap.com/documents/2016/02/5c1f8d00-617c-0010-82c7-eda71af511fa.html, 2017. Accessed: 2018-01-31.

[SAP17g] Studio SAP. Predictive analytics reimagined for the digital enterprise. https://www.sap.com/documents/2015/05/280754e0-247c-0010-82c7-eda71af511fa.html, 2017. Accessed: 2018-01-31.

[SAP18a] SAP. Sap leonardo machine learning. https://www.sap.com/products/leonardo/products.html, 2018. Accessed: 2018-01-30.

[SAP18b] SAP. Sap service ticket intelligence. https://help.sap.com/viewer/934ccff77ddb4fa2bf268a0085984db0/1712/en-US, 2018. Accessed: 2018-01-30.

[SAP18c] SE SAP. Sap hana automated predictive library (apl) reference. https://help.sap.com/viewer/4055990955524bb2bc61ee75de3b08ff/3.3/en-US, 2018. Accessed: 2018-06-24.

[SAP18d] SE SAP. Sap hana predictive analysis library (pal) reference. *SAP HANA Platform SPS*, 12, 2018.

[SAP18e] SE SAP. Sap predictive service user guide. https://help.sap.com/doc/dd5f65e965fc4f75a96aea00bf898344/Cloud/en-US/ps_cloud_ug_en.pdf, 2018. Accessed: 2018-01-31.

[SB03] Da-Wen Sun and Tadhg Brosnan. Pizza quality evaluation using computer vision—-part 1: Pizza base and sauce spread. *Journal of Food Engineering*, 57(1):81–89, 2003.

[Sch15] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[Set10] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[SHS+17] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*, 2017.

[SPE+14] D Sculley, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, and Michael Young. Machine learning: The high-interest credit card of technical debt. 2014.

[Sun00] Da-Wen Sun. Inspecting pizza topping percentage and distribution by a computer vision method. *Journal of food engineering*, 44(4):245–249, 2000.

[Sun16] Da-Wen Sun. *Computer vision technology for food quality evaluation*. Academic Press, 2016.

[Swa00] P. M. Swamidass, editor. *ERPEnterprise Resource Planning (ERP)ERP*, pages 197–197. Springer US, Boston, MA, 2000.

[Tim95] AJM Timmermans. Computer vision system for on-line sorting of pot plants based on learning techniques. In *II International Symposium On Sensors in Horticulture 421*, pages 91–98, 1995.

[Tri00] Evangelos Triantaphyllou. Multi-criteria decision making methods. In *Multi-criteria decision making methods: A comparative study*, pages 5–21. Springer, 2000.

[Tur50] Alan M Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

[UG+96] Fayyad Usama, P Gregory, et al. Knowledge discovery and data mining: Towards a unifying framework. In *KDD*, volume 96, pages 82–8, 1996.

[VK04] Vijay Vaishnavi and William Kuechler. Design research in information systems. 2004.

[VOE11] Richard L Villars, Carl W Olofson, and Matthew Eastwood. Big data: What it is and why you should care. *White Paper, IDC*, 14, 2011.

[WH00] Rüdiger Wirth and Jochen Hipp. Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, pages 29–39. Citeseer, 2000.

[Wil12] Paul Williams. Unstructured data and the enterprise. *Estados Unidos de América: California: Smartlogic US*, 2012.

[WR15] Ying Wu and R Razavi. An introduction to deep learning examining the advantages of hierarchical learning. *Thought Leadership Paper Predictive Analytics SAP*, 2015.

[WWDK17] Daniel Wellers, Jeff Woods, Jendroska Dirk, and Christopher Koch. Why machine learning and why now? `https://www.sap.com/products/leonardo/machine-learning.html`, 2017. Accessed: 2018-01-20.

[Zac16] Giancarlo Zaccone. *Getting Started with TensorFlow*. Packt Publishing Ltd, 2016.

[ZHY09] Yi Zhang, Herodotos Herodotou, and Jun Yang. Riot: I/o-efficient numerical computing without sql. *arXiv preprint arXiv:0909.1766*, 2009.

[Zin17] Martin Zinkevich. Rules of machine learning: Best practices for ml engineering, 2017.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Magdeburg, den ....